Learning to Evolve with Guiding Solutions Generated by Generative Adversarial Network

Hongwei Ge, Zhi Zheng, Yaqing Hou, *Member, IEEE*, Xia Wang, and Hisao Ishibuchi, *Fellow, IEEE*

***Abstract*—Many search strategies have been designed to generate a promising offspring population for efficiently solving large-scale multi-objective optimization problems (LSMOPs). The effectiveness of existing search strategies relies on the quality of good parent solutions. However, especially in early generations, the current population does not always include high-quality solutions. This paper proposes a generative adversarial network (GAN)-guided search (G2S) strategy for learning to evolve with guiding solutions. Its main idea is to employ GAN for mapping a set of guiding points in the objective space with good convergence and diversity back to the decision space to guide evolution. Specifically, the current population is used as real data, and the guiding points consisting of nondominated solutions and reference vectors are used as virtual data. The trained GAN generates guiding solutions in the decision space to guide the population to evolve efficiently. A large-scale multi-objective evolutionary framework using G2S is also proposed which can be embedded into multi-objective evolutionary algorithms (MOEAs) to improve their ability to handle LSMOPs. Experimental studies on several benchmark problems with the highest 5000-dimensional decision space show that the proposed G2S is competitive compared with the state-of-the-art algorithms and has impressive efficiency as the component to improve the performance of MOEAs for solving LSMOPs.**

***Index Terms*—Large-scale Multi-objective Optimization, Generative Adversarial Network, GAN-guided Search Strategy.**

I. Introduction

THERE are many multiobjective optimization problems (MOPs) in the real world [1], [2], [3]. Multiobjective evolutionary algorithms (MOEAs) for solving MOPs aim to search for a set of trade-off solutions with good convergence and diversity from multiple conflicting objective functions. Many MOPs have hundreds of decision variables, such as cloud computing [4], feature selection [5] and neural architecture search [6]. These optimization problems are usually called large-scale multiobjective optimization problems (LSMOPs) when the dimensionality of the decision space is greater than 500. Such a huge search space increases the difficulty of tackling LSMOPs. The curse of dimensionality [7] caused by high-dimensional decision spaces makes MOEAs ineffective in handling LSMOPs. Hence, many large-scale multiobjective optimization evolutionary algorithms (LMOEAs) have been proposed to solve LSMOPs efficiently. These LMOEAs can be broadly classified into three categories [8]: decision variable grouping-based approaches, objective space reduction-based approaches, and novel search strategy-based approaches.

Since the high-dimensional search space of an LSMOP causes difficulties, the most intuitive approach is to categorize the decision variables and divide the original LSMOPs into some smaller subproblems. The grouping strategy based on variable analysis is more accurate compared to random grouping [9], [10] and differential grouping [11]. These grouping strategies have been widely studied. MOEA/DVA [12] analyzes the control property of each decision variable by perturbation and divides the decision variables into position, distance and mixed variables. LMEA [13] proposes a method for clustering decision variables into convergence-related and diversity-related variables by measuring their contribution to convergence and diversity. LMEA categorizes decision variables more accurately because MOEA/DVA classifies some variables as mixed. LSMOEA/D [14] argues that MOEA/DVA and LMEA assume consistent control properties of decision variables in the global objective space. LSMOEA/D proposes a local decision variable analysis method that uses reference vectors in MOEA/D [15] to categorize decision variables. LERD [16] proposes an approach to solve the problem of high computing overhead in the decision variable analysis phase. This approach reformulates the decision variable analysis process as an optimization problem with binary decision variables.

Different from decision variable grouping strategies, the decision space reduction strategy mainly reduces the decision space dimension of the original LSMOPs. Specifically, the search is carried out in the reduced decision space by problem transformation or dimensionality reduction. WOF [17] assigns a joint weight to each set of decision variables, transforming the problem of optimizing n decision variables into optimizing q weights ($q << n$). LSMOF [18] selects k high-quality solutions to construct $2k$ reference directions with the upper and lower bounds of decision spaces, respectively. These reference directions are associated with $2k$ weight variables and the decision space is reconstructed by the weight variables ($2k << n$). MOEAs can be embedded into LSMOF which can improve the convergence speed of MOEAs for solving LSMOPs. In addition, various machine learning techniques are also applied to solve LSMOPs, including but not limited

This work was supported by the National Natural Science Foundation of China under Grant 61976034, the Dalian Science and Technology Innovation Fund under Grant 2022JJ12GX013, the Liaoning Natural Science Foundation under Grant 2022-YGJC20, and the Fundamental Research Funds for the Central Universities under Grant DUT23YG103. *(Corresponding author: Yaqing Hou.)*

H. Ge, Z. Zheng, X. Wang, and Y. Hou are with the School of Computer Science and Technology, Dalian University of Technology, Dalian 116024, China (e-mail: hwge@dlut.edu.cn; houyq@dlut.edu.cn).

H. Ishibuchi is with the Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen 518055, China (e-mail: hisao@sustc.edu.cn).

This paper has supplementary downloadable material available at xxxx, provided by the authors. This consists of a PDF file containing some additional experimental analysis and results, such as tables and figures. This material is 5.3 MB in size.

to random embedding [19], pattern mining [20] and principal component analysis [21].

Furthermore, decision variable grouping-based strategies waste a great number of function evaluations (FEs) in the grouping phase, which results in low efficiency. The strategies based on decision space reduction search in the reduced subspace are prone to fall into local optima because the global optimal solution may not be in the reduced decision space. Recent trends focus on designing novel search strategies and evolutionary operators in the original decision space in order to realize efficient search without losing important global information. These search strategies are mainly divided into two categories: probability model based [22], [23] and new reproduction operators. One type of probability model-based strategies is to build an inverse model from the objective space to the decision space. IM-MOEA [24] uses Gaussian process to map the nondominated solutions from the objective space to the decision space. IM-MOEA/D [25] utilizes k-means clustering to divide the population and builds inverse models for each subpopulation that can handle LSMOPs. EAGO [26] considers the search in the high-dimensional decision space as time-consuming and therefore proposes three-objective vector generation methods to generate a population in the objective space. Then, a matrix is used to map the population generated in the objective space back to the decision space.

Some studies recognize the necessity to design novel reproduction operators for searching efficiently in high-dimensional decision spaces to enhance the performance of solving LSMOPs [27], [28]. These strategies directly use the dominance relationships of solutions in the current population to generate search directions. However, the quality of nondominated solutions is poor in early generations. As a result, good offspring cannot be generated. To further improve the search efficiency on LSMOPs and to ensure good convergence and diversity of solutions generated through search directions, this paper proposes a search strategy based on GAN [29] to guide evolution through learned guiding information. This paper has the following three main contributions:

(1) We train a GAN to generate guiding solutions with good convergence and diversity. The trained GAN maps the learned information back to the decision space to guide the evolution of the population.

(2) We propose a GAN-guided search strategy, termed G2S. The current population is associated with guiding solutions generated by a well-trained GAN. G2S ensures that the population learns to evolve in a good direction using the guiding solutions superior to the current population.

(3) We design a general LMOEA framework using G2S, termed G2SLMOEA. G2S can be embedded into most well-known MOEAs, which significantly improves the performance of the original MOEAs.

The rest of this paper is organized as follows. Section II gives the reviews of representative algorithms and the motivations of this paper. Section III gives the details of the proposed method focusing on G2S. Section IV presents experimental studies to validate the effectiveness of the proposed G2S. Section V concludes the main contributions of this paper and gives crucial directions for future research.

II. Related work and motivation

A. Generative Adversarial Networks

GAN is a generative model that contains a pair of neural networks, a generator and a discriminator, denoted G and D, respectively. GAN can generate data similar to the distribution of the original data. Suppose G is to learn the distribution of the data $\boldsymbol{x}$ and a prior noise $p_{\boldsymbol{z}}(\boldsymbol{z})$ as the input to G. A multilayer perceptron (MLP) [30] with parameter θ_g is modeled as a mapping $G(\boldsymbol{z};\theta_g)$ from noise to data space. An MLP $D(\boldsymbol{x};\theta_d)$ is also used as D to output a probability value between 0 and 1, which represents the probability of discriminating whether the input data is real. D is trained to maximize the probability of identifying real data, while G is trained to minimize the probability of D identifying the generated data as virtual data. The process of training is a game between G and D. The function of the process can be expressed as:

$$\begin{aligned}\min_G \max_D V(G,D) = \ & \mathbb{E}_{\boldsymbol{x}\sim P_{data(\boldsymbol{x})}}[\log D(\boldsymbol{x})] \\ & + \mathbb{E}_{\boldsymbol{z}\sim P_{\boldsymbol{z}(\boldsymbol{z})}}[\log(1-D(G(\boldsymbol{z})))]\end{aligned} \tag{1}$$

where $\boldsymbol{x}\sim P_{data(\boldsymbol{x})}$ and $\boldsymbol{z}\sim P_{\boldsymbol{z}(\boldsymbol{z})}$ represent sampling from the real distribution and noise, respectively. $D(\boldsymbol{x})$ and $G(\boldsymbol{z})$ represent the output of the discriminator and the generator, respectively.

During the training of GAN, m noises are sampled from noise prior $p_g(\boldsymbol{z})$ and m data are sampled from the real data to update D using stochastic gradient ascent:

$$\bigtriangledown_{\theta_d}\frac{1}{m}\sum_{i=1}^{m}[\log D(\boldsymbol{x}^{(i)})+\log(1-D(G(\boldsymbol{z}^{(i)})))] \tag{2}$$

Then, m noises are sampled from noise prior $p_g(\boldsymbol{z})$, and G is updated using stochastic gradient descent:

$$\bigtriangledown_{\theta_g}\frac{1}{m}\sum_{i=1}^{m}\log(1-D(G(\boldsymbol{z}^{(i)}))) \tag{3}$$

There have been some studies that use GAN to assist in solving LSMOPs. GMOEA [31] is the first algorithm to apply GAN to the field of multi-objective optimization. The population is categorized into real and virtual samples based on convergence and diversity as training data for GAN. Subsequently, offspring solutions are obtained from the trained GAN. The generator samples data from a multivariate normal distribution of real samples and generates data that can deceive the discriminator, i.e., high-quality offspring solutions. Although GMOEA has demonstrated effectiveness in solving several hundred-dimensional LSMOPs, its performance and efficiency in dealing with higher-dimensional LSMOPs need further investigation. GAN-LMEF [32] is a manifold interpolation framework based on GAN, which learns the characteristics of the high-dimensional solution set through GAN. By interpolating on an $m-1$ dimensional manifold, and generating offspring solutions through the trained generator. However, the goodness of the manifold learned by the GAN relies on the quality of nondominated solutions used for GAN training. Moreover, there is room for improvement in the manifold interpolation method for LSMOPs with complex search spaces, such as problems with disconnected PF.

B. Representative Studies

Representative studies are introduced from two aspects: the application of learning techniques in MOPs and strategies that provide promising search directions.

With the rapid development of machine learning (ML) techniques, some studies have begun to apply ML to solve LSMOPs. Tian et al. [33] propose an algorithm for solving sparse LSMOPs by learning the Pareto-optimal subspace via two unsupervised networks. Liu et al. [34] create a discriminative reconstruction network consisting of an encoder, a decoder, and a classifier for each LSMOP. This network can directly transfer solutions from the source LSMOP to the target LSMOP. These two methods utilize neural networks to learn the optimal subspace to improve solving efficiency. Designing new evolutionary computation paradigms inspired by the behavior and capabilities of human beings is also an important research direction. Taking cues from the learning and optimization capabilities in human problem-solving processes, Zhan et al. [35] introduce a novel learning-based evolutionary optimization framework. This framework seeks to enhance the efficiency of solving optimization problems by integrating neural networks and evolutionary computation. Inspired by human beings' ability to acquire knowledge from the successful experiences of their predecessors, a knowledge learning evolutionary computation (KLEC) is proposed [36]. The KLEC maintains a feedforward neural network-based knowledge library model (KLM) to learn from experiences to obtain knowledge. Based on KLM, KLEC utilizes the knowledge about the relationship between the individual position and its successful evolutionary direction to guide individuals for better evolution. Considering the importance of knowledge to evolution, a type of meta-knowledge has been proposed to guide the evolution of task-specific knowledge [37]. The meta-knowledge represents the knowledge of how to solve problems via evolution and the methods of evolving high-quality solutions. Although machine learning can acquire information that guides evolution, several questions merit further in-depth study: how to train the models, how to define the evolutionary information that the models are expected to learn, and how to utilize the learned information effectively in guiding evolution.

One of the major challenges in solving LSMOPs is the poor performance of traditional evolutionary operators. For example, simulated binary crossover (SBX) is inefficient in generating promising offspring solutions when dealing with LSMOPs [31]. Both parent mating and offspring reproduction of traditional operators have a large randomness [38]. The failure to give a precise evolutionary direction is the main factor that makes traditional evolutionary operators ineffective in solving LSMOPs. Due to the inefficiency of traditional evolutionary operators for LSMOPs, some novel search strategies have been designed. Inspired by LSMOF, a direction sampling strategy is proposed in LMOEADS [27] called DS. DS has the promise of sampling high-quality solutions close to the PS, and these solutions assist in generating offspring solutions. Two kinds of adaptive direction vectors are proposed in DGEA [38] to guide the generation of good offspring solutions, balancing convergence and diversity. Nondominated solutions are first used to guide the convergence of solutions towards the PS, and then the solutions spread along the PS. ALMOEA [30] introduces an MLP to solve LSMOPs. The MLP is trained to learn the fastest possible direction of convergence. The current population is input into the trained MLP, and evolution is guided by the direction of the learned fastest convergence, accelerating the speed of solving LSMOPs.

These studies can be categorized as strategies for giving promising search directions in large-scale search spaces. However, they still suffer from the problem that the accuracy of the search directions depends on the quality of the current population. For example, in ALMOEA, the fastest direction of convergence predicted by the trained MLP relies on the convergence and diversity of nondominated solutions. In DGEA, direction vectors are generated from dominant solutions to nondominant solutions. Since nondominated solutions are not necessarily well-converged and well-distributed solutions over the Pareto front (especially in early generations), there is no guarantee that the offspring solutions generated through the provided search directions have good convergence and diversity. This leads to difficulties in generating a high-quality offspring population in both ALMOEA and DGEA.

C. Motivation

To address the above-mentioned difficulties in existing methods to generate good offspring, our goal is to design a more accurate search strategy that guarantees good convergence and diversity of generated solutions. The difficulties in the existing methods are illustrated in this subsection by taking IMMOEAD, LMOEADS, DGEA, ALMOEA, LERD, FDV [39] and LMOCSO [28] as examples. These algorithms are compared in Fig. 1 on 2-objective LSMOP1 [40] with $n = 1000$ where average IGD values over 20 runs of each algorithm are shown. The abscissa represents the percentage of consumed FEs. These state-of-the-art algorithms suffer from several problems such as large IGD values, slow convergence speed and premature convergence to local solutions in Fig. 1. The IGD values of IMMOEAD (black hexagram) and LMOCSO (yellow squares) are not ideal and may converge to local solutions, which shows that tackling LSMOPs requires the design of efficient search strategies. Whereas LERD (light blue circles) is slightly better than LMOCSO in Fig. 1, its final IGD value is still large. All the other algorithms achieve better results in Fig. 1. This may be because they are all designed with efficient search strategies. The weaker results of FDV (purple triangles) than DGEA (red diamonds), LMOEADS (small blue circles) and ALMOEA (green stars) suggest that traditional genetic operators of MOEAs used in FDV are less efficient than novel offspring generation strategies in DGEA and ALMOEA. DGEA exhibits a slower convergence speed than ALMOEA and LMOEADS, resulting in the third-ranked IGD values in Fig. 1. Although LMOEADS uses traditional operators, DS can sample high-quality guiding solutions. The AES strategy in ALMOEA speeds up convergence and achieves the best IGD values. However, LMOEADS and ALMOEA tend to converge prematurely.

Fig. 1 demonstrates that all of the examined state-of-the-art algorithms still have difficulties in efficiently handling

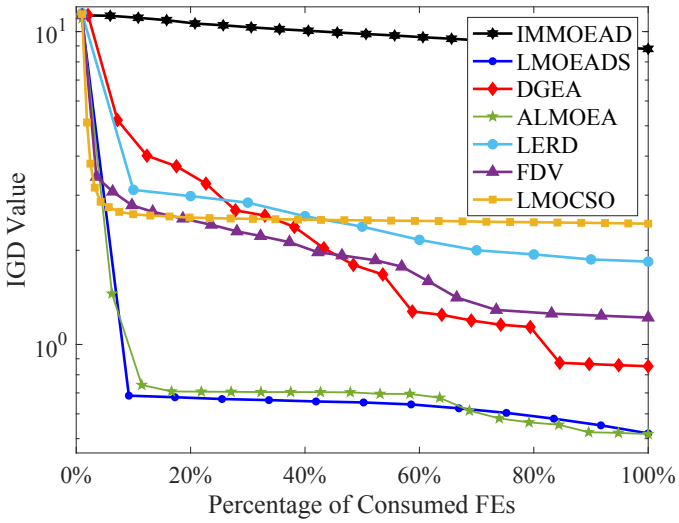


Fig. 1. IGD convergence curves of seven algorithms on the LSMOP1 problem with $n = 1000$ under FEs $= 10000$.

LSMOPs whereas some algorithms are more efficient than others. In this paper, we aim to give a more advanced search strategy to guide evolution, which ensures good convergence and diversity of offspring solutions generated through search directions. The purpose of solving MOPs is to obtain a subset of the PS that uniformly covers the Pareto front. Inspired by this, if we know the approximate location of the Pareto front in the search space, its efficient use to generate guiding solutions can greatly improve the efficiency of the search. Because the Pareto front has superior convergence and diversity compared to the current population, it naturally provides effective guidance. In principle, the high quality of generated offspring is achieved by mapping the Pareto front back to the decision space to guide evolution. Our idea is to use a set of guiding points with good convergence and diversity in the objective space for the population to learn to evolve. In order to realize this idea, it is necessary to know the distribution of guiding points in the decision space as accurately as possible

It is a crucial problem to map the guiding information (i.e., approximation of the Pareto front) in the objective space to the high-dimensional decision space. Existing mapping methods [24], [25], [26] have difficulty in generating reliable distribution in a high-dimensional decision space of a large-scale multiobjective problem. In [24], Gaussian process requires establishing a univariate inverse model for each objective function and randomly grouping the decision variables. The scalability of IM-MOEA [24] for solving LSMOPs remains to be examined. Whereas the matrix approach is also used to map the solutions generated in the objective space to the decision space in [26], the accuracy of the mapping needs to be improved. GAN is widely used to generate high-dimensional data by training G and D to learn the distribution of data. In this paper, GAN is trained to learn the distribution of guiding points in the objective space, and then generate guiding solutions by mapping guiding points back to the decision space. This paper constructs a simplex to simulate the Pareto front by nondominated solutions and reference vectors and trains GAN to map useful guiding information back to the decision space. The proposed G2S is based on GAN to generate guiding solutions so that G2S can find a well-converged and well-distributed solution set efficiently for LSMOPs.

III. Proposed Framework G2SLMOEA

In this section, Section III.A and III.B give a detailed description of the proposed G2S which enhances the performance of MOEAs in solving LSMOPs by training GAN to generate high-quality solutions. Section III.C gives a general framework for solving LSMOPs using G2S.

A. Training GAN for Generating Guiding Solutions

Since the Pareto front is the final target of MOPs, GAN is used to learn the distribution of the approximated Pareto front in the decision space. The current population is chosen as the real data to train D. Specifically, training D needs a set of input-output pairs $\{\mathbf{x}_i, \mathbf{F}(\mathbf{x}_i)\}_{i=1}^{N}$, where $\mathbf{F}(\mathbf{x}_i)$ is a solution in the objective space and $\mathbf{x}_i$ is the related solution in the decision space. The ith variable x_i of solution $\mathbf{x}$ is normalized as follows:

$$x_i' = \frac{x_i - lower_i}{upper_i - lower_i}, i = 1, 2, \ldots, n \tag{4}$$

where $\mathbf{lower} = (lower_1, ..., lower_n)$ and $\mathbf{upper} = (upper_1, ..., upper_n)$ are the corresponding lower and upper bounds of the n-dimensional decision space. All solutions in the current population are used as the input-output pair for the training of D. For virtual data, the simplex constructed by nondominated solutions and reference vectors is used to approximate the Pareto front. The solutions of the first front are found through nondominated sorting. The mean value of the sum of objective values over all nondominated solutions in the current population is then calculated, denoted as o_n. A simplex is defined using o_n as $f_1 + \ldots + f_m = o_n$. The intersections of the reference vectors with the simplex are used as guiding points in the objective space, denoted as $\mathbf{O_{guide}}$. The use of GAN is aimed at learning the distribution of $\mathbf{O_{guide}}$ in the decision space. So $\mathbf{O_{guide}}$ is used as the input for G. The output of G is the mapped solutions of $\mathbf{O_{guide}}$ from the objective space to the decision space, denoted as guiding solutions $\mathbf{X_{guide}}$. The guiding solutions generated by G are also input to D for discrimination. The parameters of D and G are updated through (2) and (3), respectively.

The pseudo code for generating guiding solutions through GAN is presented in Algorithm 1. The input of Algorithm 1 is the initialized GAN, the current population and the set of reference vectors. The output of Algorithm 1 is guiding solutions. First, normalize the decision variables of the current population. Then, a simplex is constructed and its intersections with reference vectors are sampled to obtain guiding points (lines 3-5). GAN is trained by guiding points and decision variables (lines 7-10). Finally, guiding solutions are generated using the trained generator (line 12). Useful guiding information is mapped to the decision space by a well-trained GAN to guide the evolution of the population toward well-converged and well-distributed solutions.

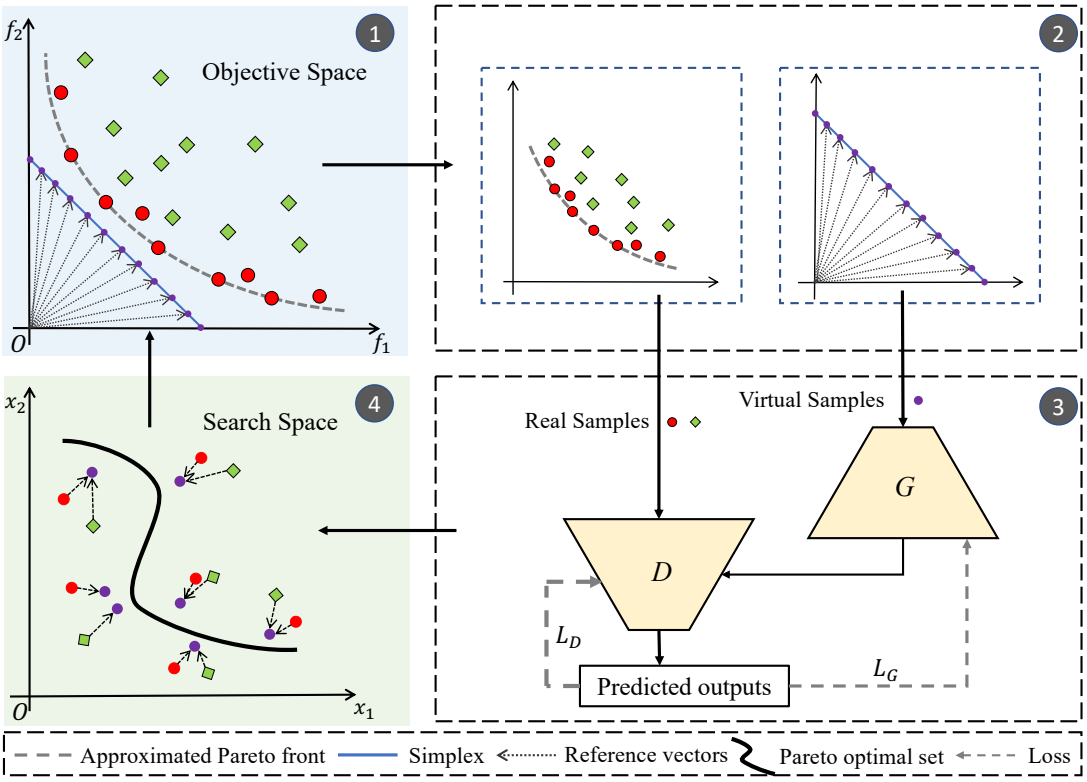


Fig. 2. Illustration of the G2S stategy. Step 1: Nondominated sorting and simplex construction. The nondominated sorting divides the population into frontiers (red circles) and nonfrontiers (green diamonds). Construct the simplex by calculating the mean of the sum of objective values. Step 2: Pick virtual data for training GAN on the simplex. Intersection points between the simplex and reference vectors are selected and distributed evenly in the objective space. Step 3: Training of GAN. Frontiers and nonfrontiers are input as real data into D, and the virtual data is input into G. The output of G is also input into D. Calculate the loss and update the parameters of G and D. Step 4: Generation of offspring solutions. The trained G generates guiding solutions. Each solution is associated with a guiding solution to generate the offspring population.

Algorithm 1 Training of GAN.

Input: GAN, $\mathbf{P}$ (current population), $\mathbf{Z}$ (set of reference vectors)

Output: $\mathbf{X_{guide}}$ (guiding solutions)

Normalize the variable values of all solutions in $\mathbf{P}$ by (4);
// Sampling guiding points $\mathbf{O_{guide}}$;
$o_n \leftarrow$ Calculate the mean of the sum of objective function values of $\mathbf{P}$;
Build simplex : $f_1 + \ldots + f_m = o_n$;
$\mathbf{O_{guide}} \leftarrow$ Intersect(simplex : $f_1 + \ldots + f_m = o_n$, $\mathbf{Z}$);
// Update discriminator D and generator G
for $epoch < epochs$ **do**
    Update θ_d by (2) $\leftarrow D(\mathbf{P}.\mathbf{x}; \theta_d) + G(\mathbf{O_{guide}}; \theta_g)$;
    Update θ_g by (3) $\leftarrow G(\mathbf{O_{guide}}; \theta_g)$;
end for
// Generate guiding solutions $\mathbf{X_{guide}}$
$\mathbf{X_{guide}} \leftarrow G(\mathbf{O_{guide}}; \theta_g)$;

B. GAN-Guided Search Strategy

The quality of the offspring population depends on the accuracy of the search direction in a high-dimensional decision space during evolution. However, recently-proposed advanced search strategies often rely on the quality of the current population, especially the quality of nondominated solutions. The quality of the population is generally poor in the early stages of evolution, so existing search strategies may lead to slow convergence or fall into local optima. A more precise search direction needs to be given to guide the evolution if better optimization results are to be achieved more efficiently. Therefore, in this paper, a GAN-guided search strategy is proposed to generate a high-quality offspring population enhancing the performance of MOEAs to solve LSMOPs.

A set of guiding solutions $\mathbf{X_{guide}}$ are obtained in Algorithm 1. These solutions are obtained by mapping from guiding points $\mathbf{O_{guide}}$ in the objective space, which exhibit better convergence and diversity than the current population, to the decision space. Each $\mathbf{x}$ in the current population is matched with an $\mathbf{x_{guide}}$ in $\mathbf{X_{guide}}$ through the value of the acute angle between them. $\theta(\mathbf{x}, \mathbf{x_{guide}})$ is as computed follows:

$$\theta(\mathbf{x}, \mathbf{x_{guide}}) = \arccos \left| \frac{\mathbf{F}'(\mathbf{x}) \cdot \mathbf{F}'(\mathbf{x_{guide}})}{\|\mathbf{F}'(\mathbf{x})\| \cdot \|\mathbf{F}'(\mathbf{x_{guide}})\|} \right| \quad (5)$$

where $\|\cdot\|$ denotes the Euclidean distance, $\mathbf{F}'(\mathbf{x}) = (f_1'(\mathbf{x}), \ldots, f_m'(\mathbf{x}))$ is the normalized vector of the m objective function values for the individual $\mathbf{x}$, $f_i'(\mathbf{x})$ is computed as:

$$f_i'(\mathbf{x}) = \frac{f_i(\mathbf{x}) - f_i^{min}}{f_i^{max} - f_i^{min}}, i = 1, 2, \ldots, m \quad (6)$$

where f_i^{max} and f_i^{min} are the maximum and minimum values of the i-th objective function in the population, respectively.

Then, the individual $\mathbf{x}$ matches a guiding solution with the minimum acute angle as follows:

$$\mathbf{x}_{\mathbf{guide}}^{\mathbf{best}} = \mathbf{x}_{\mathbf{guide}} : \underset{\mathbf{x}_{\mathbf{guide}} \in \mathbf{X}_{\mathbf{guide}}}{\arg\min} \ \theta(\mathbf{x}, \mathbf{x}_{\mathbf{guide}}) \tag{7}$$

Since there is a risk of mode collapse when training GAN [41], the model may generate poor guiding solutions. Mode collapse occurs when the generator fails to produce a diverse and high-quality set of samples. Instead, it generates a limited or repetitive selection of samples. To prevent this, it's crucial to determine the search direction by comparing the distances of each $\mathbf{x}$ and $\mathbf{x}_{\mathbf{guide}}^{\mathbf{best}}$ to the ideal point. Thus, the search direction for each individual $\mathbf{x}$ is defined as follows:

$$\mathbf{dir} = \begin{cases} \mathbf{x} - \mathbf{x}_{\mathbf{guide}}^{\mathbf{best}}, & \text{if } ED(\mathbf{x}_{\mathbf{guide}}^{\mathbf{best}}) \leq ED(\mathbf{x}), \\ \mathbf{x}_{\mathbf{guide}}^{\mathbf{best}} - \mathbf{x}, & \text{otherwise.} \end{cases} \tag{8}$$

where $ED(\mathbf{x})$ is as follows:

$$ED(\mathbf{x}) = \sqrt{\|\mathbf{F}'(\mathbf{x})\| \cdot \|\mathbf{F}'(\mathbf{x})\|} \tag{9}$$

It is also worth noting that when the value of $\mathbf{dir}$ is too large (i.e., the distance between $\mathbf{x}_{\mathbf{guide}}^{\mathbf{best}}$ and $\mathbf{x}$ in decision space is too far, especially in early generations), some of offspring solutions generated by $\mathbf{dir}$ will be outside the upper and lower bounds of the search spaces. So $\mathbf{dir}$ is normalized by its Euclidean distance.

An offspring solution $\mathbf{x}_{\mathbf{new}}$ is obtained by the proposed G2S strategy as follows:

$$\mathbf{x}_{\mathbf{new}} = \mathbf{x} + r_1 \times \frac{\mathbf{dir}}{\|\mathbf{dir}\|} + r_2 \times (\mathbf{x}_{\mathbf{d_1}} - \mathbf{x}_{\mathbf{d_2}}) \tag{10}$$

where $\|\mathbf{dir}\|$ is the Euclidean distance of $\mathbf{dir}$, $\mathbf{x}_{\mathbf{d_1}}$ and $\mathbf{x}_{\mathbf{d_2}}$ are two solutions randomly selected from the current population and $\mathbf{x} \neq \mathbf{x}_{\mathbf{d_1}} \neq \mathbf{x}_{\mathbf{d_2}}$. r_1 and r_2 are random numbers between 0 and 1. Two solutions $\mathbf{x}_{\mathbf{d_1}}$ and $\mathbf{x}_{\mathbf{d_2}}$ randomly selected from the current population make the offspring population have better diversity and reduce the probability of falling into local optima. The regeneration mode of the offspring population is similar to that of differential evolution [42], but G2S gives a more precise evolution direction, ensuring that the generated offspring population has good convergence and diversity. The $\mathbf{dir}$ obtained through (8) ensures that the current population evolves toward the ideal point.

G2S can obtain an approximate optimal solution set at the early stages of evolution by providing more precise search directions. Therefore, to minimize the risk of mode collapse and reduce runtime during GAN training, GAN is used to generate the guiding solutions only in the early stages of the evolution, and the nondominated solutions are used as the guiding solutions to guide the evolution in the remaining stages. The threshold for the percentage of consumed FEs used to generate guiding solutions via GAN is θ.

When the percentage of consumed FEs exceeds the threshold θ, the primary focus of G2S is the diversity of the population. G2S aims to generate a high-quality offspring for first convergence in early generations (i.e., in the first stage) whereas it aims to increase the diversity of solutions in later generations (i.e., in the second stage). In the first stage, GAN generates guiding solutions, while in the second stage, nondominated solutions of the current population are used as guiding solutions. In the second stage, the guiding solutions and the current population are associated using a similar method as (5):

$$\mathbf{x}_{\mathbf{guide}}^{\mathbf{best}} = \mathbf{x}_{\mathbf{guide}} : \underset{\mathbf{x}_{\mathbf{guide}} \in \mathbf{X}_{\mathbf{guide}}}{\arg\max} \ \theta(\mathbf{x}, \mathbf{x}_{\mathbf{guide}}) \tag{11}$$

The pseudo code for the G2S strategy is presented in Algorithm 2. The input of Algorithm 2 is the current population and the set of reference vectors. The output of Algorithm 2 is the offspring population. The method to generate guiding solutions is determined based on the threshold θ, either through GAN (line 5) or nondominated sorting (line 7). When the percentage of consumed FEs is below the threshold θ, each individual is associated with the $\mathbf{x}_{\mathbf{guide}}$ having the minimum acute angle value (line 12). When the percentage of consumed FEs is above the threshold θ, it is associated with the $\mathbf{x}_{\mathbf{guide}}$ having the maximum acute angle value (line 14). The first stage focuses on exploration and the second stage focuses on exploitation, balancing exploration and exploitation through the two stages to ensure the performance of the optimization while speeding up the handling speed.

Algorithm 2 G2S for generating offspring population.

Input: $\mathbf{P}$ (current population), $\mathbf{Z}$ (set of reference vectors)
Output: $\mathbf{O}$ (offspring population)

1: $\mathbf{X_n} \leftarrow$ nondominated sorting($\mathbf{P}$);
2: GAN $\leftarrow$ initialize(θ_d,θ_g);
3: //Two methods to obtain guiding solutions
4: **if** $\frac{FE_{GAN}}{FE_{max}} \times 100\% < \theta$ **then**
5: $\quad\mathbf{X_{guide}} \leftarrow$ Algorithm 1(GAN, $\mathbf{P}$, $\mathbf{Z}$);
6: **else**
7: $\quad\mathbf{X_{guide}} \leftarrow \mathbf{X_n}$;
8: **end if**
9: //Generate offspring population
10: **for** each $\mathbf{x} \in \mathbf{P}$ **do**
11: $\quad$**if** $\frac{FE_{GAN}}{FE_{max}} \times 100\% < \theta$ **then**
12: $\quad\quad\mathbf{x}_{\mathbf{guide}}^{\mathbf{best}} \leftarrow$ associate $\mathbf{x}$ and $\mathbf{X_{guide}}$ by (5);
13: $\quad$**else**
14: $\quad\quad\mathbf{x}_{\mathbf{guide}}^{\mathbf{best}} \leftarrow$ associate $\mathbf{x}$ and $\mathbf{X_{guide}}$ by (11);
15: $\quad$**end if**
16: $\quad[\mathbf{x_{d_1}}, \mathbf{x_{d_2}}] \leftarrow$ choose two random solutions from $\mathbf{P}$;
17: $\quad\mathbf{x_{new}} \leftarrow$ generate offspring by (10);
18: $\quad\mathbf{x_{new}} \leftarrow$ mutate($\mathbf{x_{new}}$);
19: $\quad\mathbf{O} \leftarrow$ add $\mathbf{x_{new}}$ into the offspring population;
20: **end for**

C. General Framework of G2SLMOEA

The proposed G2S strategy can be seamlessly embedded as a plug-and-play component into existing MOEAs, ensuring the quality of the generated offspring population and enhancing the ability to solve LSMOPs. The pseudo code for the framework of G2SLMOEA is presented in Algorithm 3. G2SLMOEA consists of three main components: training GAN to learn the distribution of guiding points in the decision space, generating an offspring population using the G2S

strategy, and environmental selection for the next population. The core of G2SLMOEA is the G2S strategy which can provide high-quality guiding solutions and give precise search directions for the current population to evolve. The G2S strategy can accelerate the convergence speed while ensuring the diversity of the offspring population. G2SLMOEA can use existing environmental selection strategies, encompassing three approaches: Pareto-based, indicator-based and decomposition-based strategies.

Fig. 2 illustrates the diagram of the execution process of the G2S strategy. The flowchart of G2SLMOEA is presented in Fig. 3. The core of G2SLMOEA is highlighted in the blue box which can generate an offspring population with good convergence and diversity.

Algorithm 3 General framework of G2SLMOEA.

Input: $\mathbf{P}$ (current population), $\mathbf{Z}$ (set of reference vectors), MOEA (existing optimization algorithm)
Output: $\mathbf{P}$ (population for the next generation)

GAN $\leftarrow$ initialize(θ_g,θ_d);
//Optimize using G2S in early generations
while $\frac{FE_{GAN}}{FE_{max}} \times 100\% < \theta$ **do**
    // Algorithm 2
    Generate offspring population $\mathbf{O} \leftarrow$ G2S($\mathbf{P}$,$\mathbf{Z}$);
    //Employ environmental selection in MOEA
    $\mathbf{P} \leftarrow$ Environmental selection($\mathbf{P}$,$\mathbf{O}$);
end while
//Optimize using MOEA or G2S
while $\theta \leq \frac{FE_{GAN}}{FE_{max}} \times 100\% \leq 100\%$ **do**
    $\mathbf{O} \leftarrow$ MOEA($\mathbf{P}$);
    $\mathbf{P} \leftarrow$ Environmental selection($\mathbf{P}$,$\mathbf{O}$);
end while

The computational complexity of G2SLMOEA consists mainly of training GAN, G2S strategy and environment selection. GAN contains a pair of G and D, except for the input layer and output layer, the number of hidden layers is 3 in this paper. It takes $O(\sum_{l=1}^{d} n_l \cdot n_{l-1} \cdot N^2 \cdot n^2)$ to execute one epoch of GAN training, where d is the depth of the neural networks (i.e., the number of layers), n_l is the number of neurons in the l-th layer, N is the population size, and n is the dimensionality of the decision space [32]. The time complexity of Algorithm 1 is $O(mN^2 + \sum_{l=1}^{d} n_l \cdot n_{l-1} \cdot N^2 \cdot n^2)$, where m is the number of objectives. It also takes $O(\sum_{l=1}^{d} n_l \cdot n_{l-1} \cdot N^2 \cdot n^2)$ to input the guiding points into the trained G in order to generate the guiding solutions. Different environment selection strategies have different computational complexities. For example, NSGA-II requires $O(mN^2)$ to select the offspring population. In this case, G2SLMOEA needs a total computational complexity of $O(mN^2 + \sum_{l=1}^{d} n_l \cdot n_{l-1} \cdot N^2 \cdot n^2)$. Since G2SLMOEA only trains GAN to generate guiding solutions in early generations, the computational complexity of G2SLMOEA is comparable to NSGA-II. A detailed comparison of the runtime of the proposed G2SLMOEA and state-of-the-art algorithms is carried out in the next section.

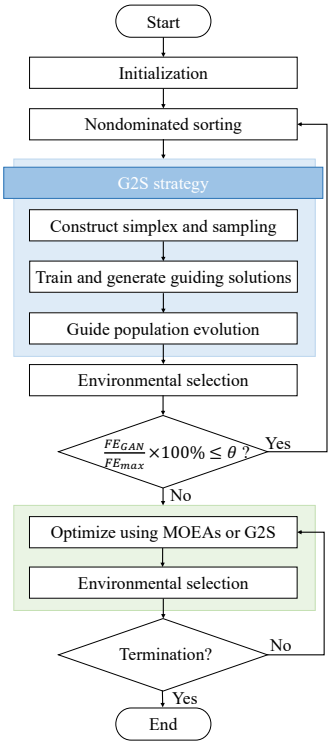


Fig. 3. Illustration of the framework for G2SLMOEA.

IV. EXPERIMENTAL STUDIES

In this section, experimental studies are conducted to validate the superiority of the proposed G2S in tackling LSMOPs. First, to verify that G2S can improve the ability of MOEAs to handle LSMOPs, G2S is embedded into NSGA-II, IBEA [43] and CLIA [44]. The effectiveness of G2S is validated by comparing the performance of the original algorithms with G2S-NSGA-II, G2S-IBEA and G2S-CLIA in solving LSMOPs. Then, to further validate the competitiveness of G2SLMOEA in dealing with LSMOPs, it is compared with six state-of-the-art algorithms to analyze whether G2SLMOEA has better performance. More explicitly, G2S-NSGA-II, G2S-IBEA and G2S-CLIA use the G2S strategy only in early generations (i.e., the percentage of consumed FEs $< \theta$), whereas G2SLMOEA always uses the G2S strategy. Finally, an ablation study is utilized to demonstrate the effectiveness of the main G2S components. In addition, runtime analysis and parameter sensitivity analysis are also conducted.

A. Experimental Settings

To validate the performance of G2S, experimental studies are conducted on the LSMOP [40] and LMF [45] test suites. In LSMOP1-9, the number of the decision variables n is set to 500, 1000, 2000 and 5000. The number of objectives is 2 and 3. For LMF1-8, the number of decision variables is set to 1000, 3000 and 5000. The number of objectives is 2. To strengthen the practical relevance and applicability of G2S, experimental studies are also conducted on the real-world scenario of the TREE [46]. For TREE1-2, the number of decision variables is set to 3000. For TREE3-5, the number of decision variables

is set to 6000. The number of objectives is 2. In this paper, two widely used metrics, inverted generational distance (IGD) [47] and hypervolume (HV) [48], are chosen to evaluate the performance of different algorithms. The formula of IGD is as follows:

$$\mathrm{IGD}(\mathbf{P}) = \frac{1}{|\mathbf{P}^*|}\sum_{\mathbf{z}^*\in\mathbf{P}^*} d(\mathbf{z}^*,\mathbf{P}) \tag{12}$$

where $\mathbf{P}$ denotes the obtained final solution set and $\mathbf{P}^*$ denotes the points uniformly sampled from the true Pareto front. $d(\mathbf{z}^*,\mathbf{P}) = \min\{d(\mathbf{z}^*,\mathbf{z})|\mathbf{z}\in\mathbf{P}\}$ represents the minimum Euclidean distance in the objective space from the sampled point $\mathbf{z}^*$ to the obtained final solution set $\mathbf{P}$. $|\mathbf{P}^*|$ denotes the number of samples on the Pareto front and is set to 10000 in experiments. HV calculates the volume of the region in the objective space surrounded by the solutions in $\mathbf{P}$ and the reference point. IGD and HV can evaluate the diversity and convergence of the obtained final solution set $\mathbf{P}$. The smaller value of IGD and the larger value of HV indicate better performance of the algorithm in handling LSMOPs. In this paper, each algorithm runs independently 20 times, and the Wilcoxon rank sum is used to compare the statistical results at a significance level of 0.05. In all tables in this paper, the symbols "+", "−" and "≈" indicate that a compared algorithm performs significantly better, significantly worse, or is indifferent to the proposed G2SLMOEA, respectively.

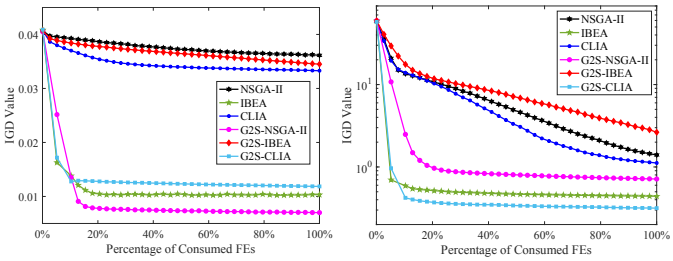
B. Parameter Settings

The parameters of the compared algorithms in the experimental studies are set the same as in the original papers and can be found in PlatEMO [49] and Supplementary Document. For 2-objective problems, the population size N is set to 100. For 3-objective problems, the population size N is 153. The number of FEs is set to 100000 for LSMOP, LMF and TREE. For the proposed G2SLMOEA, G and D are symmetric trapezoidal structures. The number of neurons of the five-layer fully connected neural network in G is set to m, 16, 64, 128 and n, respectively. The number of neurons of the five-layer fully connected neural network in D is set to n, 128, 64, 16 and 1, respectively. GAN is trained by full batch learning. The learning rate of GAN is set to 0.0002, the epoch is set to 1 (Parameter sensitivity analysis in Supplementary Document). The threshold θ for the percentage of consumed FEs used to generate guiding solutions with GAN is set to 10%. For 2-objective problems, the size of the point set sampled on the simplex N_S is set to N. For 3-objective problems, the N_S is set to $2N$. The selection of parameters is based on parameter sensitivity analyses, which are detailed later in this section.

C. Effectiveness of the G2S strategy

In order to validate the effectiveness of G2S and prove that G2S can improve the ability of MOEAs to deal with LSMOPs, the performance of the three original algorithms (NSGA-II, IBEA and CLIA) and the embedded ones (namely G2S-NSGA-II, G2S-IBEA and G2S-CLIA) on LSMOP1-9 are compared. The environmental selection strategies of the three original algorithms are Pareto-based, indicator-based, and decomposition-based strategies. Table I and Table S9 provide results for IGD and HV, respectively. (Table S9 is provided in Supplementary Document). The comparison results in Table I show that the proposed G2S can improve the performance of MOEAs to solve LSMOPs. G2S-NSGA-II, G2S-IBEA and G2S-CLIA obtain 36, 35 and 31 better IGD values than the original versions on the total of 36 problems of LSMOP1-9, respectively. More specifically, on LSMOP3 and LSMOP7 with $n = 2000$ and $n = 5000$, the original algorithms using the traditional crossover operator have difficulty to search for promising solutions in the high-dimensional decision spaces, and the obtained IGD values are poor. However, their variants with G2S give better results in optimizing difficult LSMOPs. It is also found that G2S-IBEA and G2S-CLIA suffer from performance decreases or no significant improvement in processing LSMOP6. The decision variables of LSMOP6 are partially separable and the Pareto front is convex. G2S-IBEA and G2S-CLIA tend to fall into local optima when dealing with this problem, and G2S-NSGA-II which uses a Pareto-dominated strategy to select the next population performs better. Nevertheless, from the overall results, the proposed G2S is more advantageous than the traditional crossover and mutation operators in tackling LSMOPs, which improves the efficiency of MOEAs in large-scale search spaces. The guiding solutions generated by G2S ensure that MOEAs, which may have difficulties in handling LSMOPs, obtain high-quality solutions.

The comparison between G2SLMOEA and large-scale multiobjective optimization frameworks can be found in Table S4 (in Supplementary Document). It demonstrates that G2SLMOEA can improve the performance of MOEAs better than other state-of-the-art frameworks.



(a) LSMOP2 (b) LSMOP9

Fig. 4. The IGD convergence curves of the three original algorithms and their versions embedded with G2S on 2-objective 1000-dimensional LSMOPs.

In Fig. 4, the convergence curves of the IGD values of the above algorithms are illustrated, which enables a more straightforward observation of the guiding effect of G2S in the evolutionary process. The G2S-based MOEAs outperform the original versions in terms of convergence speed and optimization performance when solving 2-objective LSMOP2 and LSMOP9. It can be observed that the population quality of the three enhanced versions significantly improves in early generations. For LSMOP2, all three original algorithms find small IGD values of about 0.035, but adopting G2S instead of SBX allows convergence to approximate optima more efficiently and quickly. Compared to LSMOP2, the true Pareto front of LSMOP9 is disconnected and more difficult

TABLE I
THE IGD RESULTS OF THREE ORIGINAL ALGORITHMS AND G2S EMBEDDED VERSIONS ON 2-OBJECTIVE LSMOP1-9

Problem	n	NSGA-II	G2S-NSGA-II	IBEA	G2S-IBEA	CLIA	G2S-CLIA
LSMOP1	500	1.8352e+0 (2.41e-1) -	**8.4627e-2 (1.24e-2)**	1.3322e+0 (2.59e-1) -	**3.2849e-1 (3.28e-2)**	6.0640e-1 (1.45e-2) -	**9.2600e-2 (1.54e-2)**
	1000	3.6250e+0 (2.44e-1) -	**2.0228e-1 (2.78e-2)**	3.8224e+0 (2.40e-1) -	**3.4383e-1 (2.09e-2)**	7.7834e-1 (8.68e-2) -	**1.9883e-1 (3.28e-2)**
	2000	5.1100e+0 (1.79e-1) -	**3.8843e-1 (8.57e-2)**	5.7945e+0 (1.79e-1) -	**4.5567e-1 (5.73e-2)**	2.2431e+0 (1.38e-1) -	**3.4533e-1 (6.96e-2)**
	5000	6.8932e+0 (1.96e-1) -	**8.4097e-1 (1.78e-1)**	7.4749e+0 (2.65e-1) -	**7.7886e-1 (1.22e-1)**	4.2737e+0 (1.48e-1) -	**7.1817e-1 (1.69e-1)**
LSMOP2	500	5.9664e-2 (1.18e-3) -	**2.0416e-2 (1.97e-3)**	5.4432e-2 (1.07e-3) -	**1.2752e-2 (6.17e-4)**	5.5397e-2 (1.12e-3) -	**1.2733e-2 (1.14e-3)**
	1000	3.6055e-2 (4.13e-4) -	**1.8826e-2 (4.98e-4)**	3.2988e-2 (6.62e-4) -	**1.1001e-2 (3.59e-4)**	3.4005e-2 (3.71e-4) -	**1.1531e-2 (3.13e-4)**
	2000	2.0670e-2 (1.63e-4) -	**1.2681e-2 (3.69e-4)**	1.9352e-2 (1.42e-4) -	**8.2183e-3 (6.27e-4)**	1.9606e-2 (8.26e-5) -	**7.9057e-3 (5.25e-4)**
	5000	**1.0773e-2 (2.27e-4) +**	1.1244e-2 (5.94e-4)	9.9333e-3 (5.86e-5) -	**6.1791e-3 (3.39e-4)**	9.6431e-3 (1.75e-5) -	**5.9400e-3 (3.45e-4)**
LSMOP3	500	1.7272e+1 (1.96e+0) -	**1.5664e+0 (8.96e-3)**	1.6484e+2 (2.02e+2) -	**1.5634e+0 (1.03e-3)**	2.2250e+0 (2.32e-1) -	**1.0670e+0 (6.51e-2)**
	1000	2.1078e+1 (1.05e+0) -	**1.5792e+0 (4.28e-3)**	3.2988e-2 (6.62e-4) -	**1.1001e-2 (3.59e-4)**	3.4005e-2 (3.71e-4) -	**1.1531e-2 (3.13e-4)**
	2000	2.4767e+1 (9.66e-1) -	**1.5867e+0 (1.94e-3)**	**3.0650e+1 (1.17e+0) =**	2.8413e+2 (4.30e+2)	9.0123e+0 (7.79e-1) -	**3.6272e+0 (2.88e-1)**
	5000	3.0266e+1 (6.41e-1) -	**1.5886e+0 (1.99e-3)**	7.6690e+2 (3.25e+3) -	**5.8456e+2 (1.81e+3)**	1.6856e+1 (5.92e-1) -	**9.9853e+0 (2.07e+0)**
LSMOP4	500	9.2576e-2 (2.06e-3) -	**4.4927e-2 (3.53e-3)**	7.8585e-2 (1.11e-3) -	**3.7962e-2 (1.57e-3)**	8.4395e-2 (1.12e-3) -	**3.8701e-2 (2.00e-3)**
	1000	6.0913e-2 (6.53e-4) -	**3.8470e-2 (1.92e-3)**	4.8910e-2 (6.60e-4) -	**2.6342e-2 (5.95e-4)**	5.0257e-2 (5.44e-4) -	**2.6694e-2 (5.96e-4)**
	2000	3.7177e-2 (3.33e-4) -	**3.0291e-2 (1.52e-3)**	3.0159e-2 (3.65e-4) -	**1.6606e-2 (5.12e-4)**	2.9916e-2 (2.78e-4) -	**1.7196e-2 (4.01e-4)**
	5000	1.9291e-2 (1.55e-4) -	**1.8193e-2 (7.93e-4)**	1.6651e-2 (1.39e-4) -	**9.3854e-3 (4.20e-4)**	1.5942e-2 (9.40e-5) -	**9.1821e-3 (3.39e-4)**
LSMOP5	500	5.3357e+0 (6.43e-1) -	**5.6178e-1 (1.11e-1)**	3.8460e+0 (4.92e-1) -	**3.4746e-1 (4.47e-3)**	2.0400e+0 (1.55e+0) -	**4.2981e-1 (1.39e-1)**
	1000	1.0534e+1 (5.85e-1) -	**7.1965e-1 (4.64e-2)**	1.0541e+1 (4.34e-1) -	**5.5310e-1 (1.16e-1)**	4.3867e+0 (1.86e+0) -	**4.7161e-1 (1.68e-1)**
	2000	1.3501e+1 (6.13e-1) -	**1.6241e+0 (2.77e+0)**	1.5779e+1 (5.71e-1) -	**1.5269e+0 (3.05e+0)**	8.0514e+0 (1.40e+0) -	**9.2503e-1 (3.81e-1)**
	5000	1.7714e+1 (3.93e-1) -	**2.4789e+0 (2.40e-1)**	1.9542e+1 (5.62e-1) -	**2.1956e+0 (3.35e-1)**	1.1433e+1 (3.85e-1) -	**2.1376e+0 (3.49e-1)**
LSMOP6	500	8.0917e-1 (1.39e-3) -	**6.9628e-1 (6.05e-2)**	1.0695e+1 (3.47e+0) -	**1.9563e+0 (2.16e+0)**	**6.2660e-1 (1.18e-1) =**	6.7307e-1 (5.32e-2)
	1000	7.7501e-1 (5.74e-4) =	**7.7139e-1 (1.58e-2)**	**3.6915e+1 (8.33e+1) =**	4.5736e+1 (1.30e+2)	**5.6075e-1 (1.37e-1) =**	6.0984e-1 (1.39e-1)
	2000	**7.5735e-1 (3.72e-5) =**	7.5736e-1 (2.80e-5)	**7.5698e-1 (6.36e-5) =**	7.5701e-1 (8.96e-5)	**4.9622e-1 (1.07e-1) =**	5.2737e-1 (1.37e-1)
	5000	**7.4774e-1 (1.22e-7) =**	7.4774e-1 (1.22e-7)	7.4772e-1 (1.09e-5) =	**7.4772e-1 (9.45e-6)**	**4.6032e-1 (1.18e-1) =**	5.2847e-1 (1.61e-1)
LSMOP7	500	1.9806e+2 (4.62e+2) -	**1.5080e+0 (1.83e-3)**	1.3026e+2 (3.50e+1) -	**1.4992e+0 (8.81e-4)**	7.0862e+1 (2.07e+1) -	**1.5035e+0 (1.38e-3)**
	1000	4.2147e+3 (2.61e+3) -	**1.5158e+0 (1.65e-3)**	2.3499e+3 (5.64e+2) -	**1.5147e+0 (2.71e-3)**	9.3069e+2 (1.56e+2) -	**1.5507e+0 (8.79e-2)**
	2000	2.1245e+4 (3.45e+3) -	**1.5226e+0 (1.12e-4)**	1.1164e+4 (1.05e+3) -	**1.2353e+3 (1.02e+3)**	5.6826e+3 (6.56e+2) -	**1.6399e+3 (1.90e+3)**
	5000	3.7263e+4 (2.59e+3) -	**1.5240e+0 (1.10e-3)**	2.9727e+4 (1.60e+3) -	**1.0369e+4 (4.58e+3)**	1.9993e+4 (1.14e+3) -	**6.4997e+3 (2.26e+3)**
LSMOP8	500	1.5677e+0 (1.40e-1) -	**2.7533e-1 (6.41e-2)**	1.8044e+0 (1.47e-1) -	**3.4330e-1 (2.08e-4)**	1.3772e+0 (2.17e-1) -	**1.9460e-1 (3.14e-2)**
	1000	4.7682e+0 (3.52e-1) -	**5.7812e-1 (7.42e-2)**	4.9361e+0 (2.53e-1) -	**3.8366e-1 (1.96e-2)**	4.0687e+0 (3.40e-1) -	**3.6415e-1 (3.27e-2)**
	2000	8.1027e+0 (2.88e-1) -	**7.7314e-1 (6.85e-2)**	7.8359e+0 (2.71e-1) -	**7.5965e-1 (9.57e-2)**	7.5674e+0 (3.41e-1) -	**5.7916e-1 (7.59e-2)**
	5000	1.1255e+1 (2.72e-1) -	**9.6911e-1 (2.11e-1)**	1.1780e+1 (2.47e-1) -	**1.1077e+0 (2.78e-1)**	1.0475e+1 (2.67e-1) -	**9.9985e-1 (2.89e-1)**
LSMOP9	500	1.2315e+0 (7.26e-3) -	**3.1330e-1 (2.80e-1)**	1.2149e+0 (6.05e-3) -	**2.9654e-1 (2.15e-1)**	1.1426e+0 (1.70e-1) -	**4.1032e-1 (2.83e-1)**
	1000	1.3871e+0 (1.18e-1) -	**2.6087e-1 (1.84e-1)**	1.1183e+0 (4.64e-2) -	**4.0715e-1 (2.27e-1)**	2.3443e+0 (2.08e-1) -	**2.7946e-1 (2.01e-1)**
	2000	7.5382e+0 (1.05e+0) -	**2.5731e-1 (1.72e-1)**	4.7646e+0 (7.16e-1) -	**3.9912e-1 (1.46e-1)**	6.8608e+0 (4.32e-1) -	**2.3687e-1 (1.46e-1)**
	5000	2.0503e+1 (1.27e+0) -	**4.7859e-1 (2.39e-1)**	1.9303e+1 (9.02e-1) -	**5.2563e-1 (1.99e-1)**	1.6865e+1 (1.14e+0) -	**4.1194e-1 (1.58e-1)**
$+/-/\approx$		1/32/3	——	0/32/4	——	0/32/4	——

to find. Since the original versions search randomly in the high-dimensional decision space, solving LSMOP9 has a slow convergence speed, making it difficult to obtain a high-quality offspring population. CLIA performs slightly better, but the IGD values are still unsatisfactory (about 1.1). G2S-based MOEAs obtain a high-quality population early in evolution. This demonstrates that G2S can guide the population to search efficiently in a high-dimensional decision space.

D. Comparisons with State-of-the-Art LMOEAs

The effectiveness of G2S has been demonstrated by achieving significant optimization results even in early generations. To further validate the competitiveness of G2SLMOEA, six state-of-the-art LMOEAs are selected to compare with the proposed framework on LSMOP and LMF. The environment selection strategy in CLIA is used in our G2SLMOEA.

1) Comparison Results on LSMOP Problems: G2SLMOEA is compared with six competitive LMOEAs on 2- and 3-objective LSMOP1-9 with n from 500 to 5000. The six compared algorithms are IMMOEAD, LMOEADS, DGEA, FDV, ALMOEA and LERA. Table II shows the average IGD values of G2SLOMEA and the six compared algorithms on 3-objective LSMOPs. Table S10 and Table S11 (in Supplementary Document) present the results for IGD and HV on 2- and 3-objective LSMOP1-9, respectively. G2SLMOEA achieves the best (i.e., smallest) IGD value on 41 out of the 72 problems in LSMOP1-9. Among the compared six algorithms, the best two algorithms are LMOEADS and FDV, which achieve 19 and 12 best IGD results respectively. Any other algorithm does not obtain the best IGD values. Specifically, among the 72 problems, G2SLMOEA performs better than IMMOEAD, LMOEADS, DGEA, FDV, ALMOEA and LERD on 70, 47, 65, 54, 63 and 72 problems, respectively.

In Table III, Friedman tests are conducted on the results of IGD and HV. Friedman test indicates that the mean rankings effectively capture significant differences in performance. With the $\alpha = 0.05$ significance level, G2SLMOEA achieves the best rankings for both IGD and HV results for $m = 2$ and 3. Specifically, for the IGD values on the 72 problems, $\chi^2 = 288.68$ and $p = 2.18 \times 10^{-59}$. For the HV values on the 72 problems, $\chi^2 = 222.35$ and $p = 3.28 \times 10^{-45}$.

LSMOP1-9 test problems are also used to analyze the capabilities of each algorithm in solving problems with diverse characteristics. Fig. 5 presents the radar diagrams specifying the ranking of the performance of each algorithm on each type of problem. LMOEADS outperforms G2SLMOEA on

TABLE II
THE IGD RESULTS OF SIX COMPARED ALGORITHMS AND G2SLMOEA ON 3-OBJECTIVE LSMOP1-9

Problem	n	IMMOEAD	LMOEADS	DGEA	FDV	ALMOEA	LERD	G2SLMOEA
LSMOP1	500	1.3975e+0 (4.47e-1) -	5.2188e-1 (4.15e-2) -	5.9423e-1 (1.34e-1) -	3.9803e-1 (7.37e-2) -	6.4091e-1 (2.11e-2) -	1.2664e+0 (1.98e-1) -	**1.7444e-1 (2.27e-2)**
	1000	4.5594e+0 (8.53e-1) -	5.6682e-1 (4.45e-2) -	6.7364e-1 (1.09e-1) -	4.3424e-1 (1.23e-1) -	5.5446e-1 (5.39e-3) -	1.3108e+0 (1.94e-1) -	**2.9557e-1 (2.29e-2)**
	2000	7.8916e+0 (4.97e-1) -	5.8634e-1 (5.48e-2) -	6.5131e-1 (1.53e-1) -	5.0714e-1 (1.41e-1) -	5.6835e-1 (9.82e-3) -	1.3825e+0 (1.35e-1) -	**3.4320e-1 (1.97e-2)**
	5000	1.0022e+1 (2.42e-1) -	6.2199e-1 (3.70e-2) -	6.5348e-1 (1.17e-1) -	5.6366e-1 (1.06e-1) -	5.7791e-1 (9.60e-3) -	1.4208e+0 (1.80e-1) -	**4.0605e-1 (8.94e-2)**
LSMOP2	500	6.5661e-2 (3.96e-4) -	4.4977e-2 (1.37e-3) =	5.0042e-2 (2.22e-3) -	4.4876e-2 (8.89e-4) =	5.7338e-2 (3.82e-4) -	6.1261e-2 (1.36e-3) -	**4.4418e-2 (7.57e-4)**
	1000	5.2075e-2 (5.70e-4) -	3.7844e-2 (9.81e-4) -	4.0358e-2 (1.08e-3) -	**3.7007e-2 (2.67e-4) =**	4.1974e-2 (1.20e-4) -	4.4634e-2 (2.15e-4) -	3.7118e-2 (2.89e-4)
	2000	4.6868e-2 (9.23e-4) -	3.4958e-2 (1.16e-3) -	3.5240e-2 (5.07e-4) -	**3.3380e-2 (1.82e-4) +**	3.5459e-2 (9.18e-5) -	3.6935e-2 (1.45e-4) -	3.3683e-2 (2.43e-4)
	5000	4.7321e-2 (1.00e-3) -	3.3141e-2 (1.04e-3) -	3.2606e-2 (1.79e-4) -	**3.1737e-2 (6.81e-5) +**	3.2511e-2 (5.61e-5) -	3.3361e-2 (1.17e-4) -	3.2012e-2 (1.12e-4)
LSMOP3	500	6.4655e+0 (1.63e+0) -	8.5239e-1 (2.42e-2) -	9.2806e-1 (4.44e-1) -	8.6586e-1 (3.23e-2) -	1.0493e+0 (2.39e-1) -	9.4161e+0 (4.43e-1) -	**8.1516e-1 (4.29e-2)**
	1000	1.2159e+1 (1.87e+0) -	8.6144e-1 (2.80e-3) -	1.1165e+0 (5.74e-1) =	1.6826e+0 (9.56e-1) -	8.6069e-1 (2.20e-5) =	9.7162e+0 (4.97e-1) -	**8.5898e-1 (7.51e-3)**
	2000	1.6849e+1 (1.38e+0) -	8.6081e-1 (8.54e-4) =	1.1427e+0 (5.83e-1) -	2.5909e+0 (1.46e+0) -	8.6070e-1 (1.50e-5) =	9.9319e+0 (3.21e-1) -	**8.5743e-1 (8.22e-3)**
	5000	3.9615e+1 (2.48e+1) -	8.6148e-1 (1.98e-3) =	1.8296e+0 (2.15e+0) -	2.2588e+0 (1.10e+0) -	8.6100e-1 (1.33e-3) =	1.0107e+1 (3.83e-1) -	**8.5953e-1 (4.55e-3)**
LSMOP4	500	1.6340e-1 (3.44e-3) -	1.0523e-1 (1.94e-3) -	1.1767e-1 (3.14e-3) -	1.0317e-1 (4.46e-3) -	1.6121e-1 (2.62e-3) -	1.4420e-1 (6.36e-3) -	**9.5710e-2 (3.62e-3)**
	1000	1.0755e-1 (8.85e-4) -	6.8196e-2 (1.32e-3) -	7.1294e-2 (2.18e-3) -	**6.3765e-2 (2.89e-3) +**	9.4893e-2 (2.01e-3) -	9.7535e-2 (2.52e-3) -	6.5259e-2 (9.91e-4)
	2000	7.2424e-2 (6.52e-4) -	4.8193e-2 (1.09e-3) -	5.0833e-2 (2.81e-3) -	**4.4647e-2 (1.02e-3) +**	6.0765e-2 (9.34e-4) -	6.3946e-2 (1.39e-3) -	4.7018e-2 (7.46e-4)
	5000	5.4695e-2 (1.60e-3) -	3.6887e-2 (1.01e-3) -	3.7611e-2 (1.04e-3) -	**3.5083e-2 (2.74e-4) +**	4.0297e-2 (2.53e-4) -	4.1512e-2 (3.02e-4) -	3.6345e-2 (3.16e-4)
LSMOP5	500	8.4194e-1 (7.28e-2) -	5.2765e-1 (1.22e-2) -	6.6350e-1 (3.23e-1) -	3.7499e-1 (1.57e-1) -	5.4330e-1 (3.64e-4) -	2.2159e+0 (9.35e-1) -	**2.8636e-1 (3.26e-2)**
	1000	2.3273e+0 (4.79e-1) -	5.3321e-1 (8.32e-3) -	8.7273e-1 (5.40e-1) -	4.8963e-1 (2.06e-1) -	5.4232e-1 (2.13e-4) -	3.0850e+0 (5.44e-1) -	**3.9624e-1 (4.56e-2)**
	2000	6.9958e+0 (7.50e-1) -	5.3695e-1 (4.11e-3) -	8.7274e-1 (5.68e-1) -	6.1577e-1 (2.95e-1) =	5.4233e-1 (1.31e-4) -	3.2609e+0 (3.09e-1) -	**4.9924e-1 (1.98e-2)**
	5000	1.4335e+1 (8.49e-1) -	**5.3855e-1 (3.30e-3) =**	7.9392e-1 (3.68e-1) -	8.8822e-1 (4.04e-1) =	5.4224e-1 (2.34e-4) -	3.0994e+0 (4.34e-1) -	5.4158e-1 (2.51e-2)
LSMOP6	500	2.2027e+1 (7.03e+0) -	**7.5699e-1 (2.28e-2) +**	1.2334e+1 (2.43e+1) =	1.2725e+0 (4.81e-1) =	1.3000e+0 (2.76e-3) -	4.6421e+0 (1.83e+0) -	1.2843e+0 (1.46e-1)
	1000	3.2229e+2 (9.68e+1) -	**7.6065e-1 (2.92e-2) +**	2.5297e+1 (5.15e+1) -	3.0044e+0 (6.98e+0) =	1.3153e+0 (3.39e-5) -	2.6635e+1 (3.80e+1) -	1.2981e+0 (1.44e-1)
	2000	3.3092e+3 (6.72e+2) -	**7.9243e-1 (4.20e-2) +**	2.3308e+1 (6.90e+1) -	7.5123e+0 (2.13e+1) -	1.3257e+0 (1.15e-3) +	1.2968e+2 (1.27e+2) -	1.3271e+0 (1.14e-2)
	5000	1.5322e+4 (2.46e+3) -	**7.7887e-1 (3.50e-2) +**	1.2822e+2 (1.70e+2) =	4.1317e+1 (6.35e+1) -	1.3315e+0 (5.84e-4) +	4.0727e+2 (1.83e+2) -	1.3372e+0 (2.97e-2)
LSMOP7	500	9.0277e-1 (9.14e-2) -	8.7613e-1 (2.79e-2) -	9.5750e-1 (7.69e-2) -	9.5183e-1 (1.48e-2) -	1.0128e+0 (8.71e-2) -	1.0103e+0 (1.10e-1) -	**8.3934e-1 (5.10e-2)**
	1000	8.4700e-1 (7.55e-2) =	8.5529e-1 (6.41e-3) -	9.2582e-1 (7.54e-2) -	9.5860e-1 (1.49e-2) -	8.5941e-1 (7.76e-3) -	1.0100e+0 (6.14e-2) -	**8.1388e-1 (7.91e-2)**
	2000	5.9909e+0 (1.65e+1) -	8.3750e-1 (2.81e-2) -	8.5159e-1 (6.85e-2) -	9.6313e-1 (2.93e-2) -	8.4831e-1 (3.63e-4) -	9.8189e-1 (2.49e-2) -	**7.9978e-1 (4.45e-2)**
	5000	8.1457e+1 (1.20e+2) -	8.3463e-1 (1.81e-2) -	8.7331e-1 (7.70e-2) -	9.5549e-1 (9.16e-3) -	8.4132e-1 (1.44e-4) -	9.6168e-1 (1.60e-3) -	**7.8670e-1 (7.44e-2)**
LSMOP8	500	7.0899e-1 (1.04e-1) -	2.6233e-1 (3.72e-2) -	1.9753e-1 (1.60e-1) -	1.0799e-1 (7.33e-2) -	2.7039e-1 (2.22e-2) -	3.0562e-1 (5.01e-2) -	**7.8023e-2 (9.55e-3)**
	1000	6.3812e-1 (6.85e-2) -	2.1495e-1 (2.91e-2) -	1.7319e-1 (1.17e-1) -	1.3990e-1 (9.31e-2) -	1.9848e-1 (5.75e-3) -	2.9228e-1 (1.18e-1) -	**8.3011e-2 (1.15e-2)**
	2000	5.8764e-1 (2.89e-2) -	2.1349e-1 (3.37e-2) -	3.2217e-1 (2.14e-1) -	2.0860e-1 (1.03e-1) -	1.9931e-1 (4.41e-3) -	5.4458e-1 (4.70e-2) -	**7.9958e-2 (3.35e-3)**
	5000	6.0296e-1 (4.02e-2) -	2.1375e-1 (1.17e-2) -	3.6801e-1 (2.45e-1) -	2.6806e-1 (1.62e-1) -	1.9654e-1 (5.08e-3) -	5.2713e-1 (5.52e-2) -	**8.0016e-2 (8.18e-3)**
LSMOP9	500	2.5204e+0 (5.70e-1) -	5.8381e-1 (2.04e-3) +	1.6755e+1 (6.06e+0) -	**5.5881e-1 (7.29e-2) +**	1.1378e+0 (1.11e-1) =	1.8372e+0 (2.01e-1) -	1.1030e+0 (1.66e-1)
	1000	1.5132e+1 (3.52e+0) -	**5.8193e-1 (1.85e-3) +**	2.8964e+1 (9.85e+0) -	2.7510e+0 (8.34e-1) -	1.2003e+0 (3.73e-2) -	3.5391e+0 (1.75e+0) -	1.1305e+0 (2.98e-1)
	2000	4.3989e+1 (5.48e+0) -	**5.8032e-1 (1.95e-3) +**	4.4108e+1 (1.50e+1) -	1.5387e+1 (4.61e+0) -	1.2051e+0 (2.81e-2) =	1.2963e+1 (5.17e+0) -	1.1955e+0 (3.92e-1)
	5000	9.1632e+1 (5.20e+0) -	**5.7892e-1 (2.41e-3) +**	5.2788e+1 (1.13e+1) -	4.0891e+1 (6.96e+0) -	1.2185e+0 (3.45e-2) +	1.8884e+1 (5.29e+0) -	3.5235e+0 (2.90e+0)
$+/-/\approx$		0/35/1	8/24/4	0/33/3	6/24/6	3/28/5	0/36/0	—

5000-dimensional disconnected and convex problems. Overall, G2SLMOEA performs slightly worse than LMOEADS on disconnected problems but excels in other characteristics.

TABLE III
FRIEDMAN TESTS FOR IGD AND HV ON LSMOP1-9 FOR SIX COMPARED ALGORITHMS AND G2SLMOEA

Metric	M	IMMOEAD	LMOEADS	DGEA	FDV	ALMOEA	LERD	G2SLMOEA
IGD	2	6.39	2.19	4.5	3.61	4.11	5.67	**1.53**
	3	6.58	2.64	4.67	3.03	3.64	5.89	**1.56**
	all	6.49	2.42	4.58	3.32	3.88	5.78	**1.54**
HV	2	2.01	4.99	4.08	4.14	4.18	2.53	**6.07**
	3	1.51	4.85	4.28	4.21	5.03	2.35	**5.78**
	all	1.76	4.92	4.18	4.17	4.6	2.44	**5.92**

To more intuitively demonstrate the performance of G2SLMOEA and the other algorithms in solving LSMOPs, the IGD convergence curves of the algorithms on some test problems are presented in Fig. 6. The abscissae represent the percentage of consumed FEs. It can be observed from the IGD convergence curve in Fig. 6(a) that the performance of IMMOEAD is poor. Although LERD reduces the computing load through the reformulated decision variable analysis method, the performance is still not satisfactory. Although LMOEADS and ALMOEA converge faster than G2SLMOEA in early generations, they tend to level off after a rapid decline. G2SLMOEA can make more effective use of computing resources and achieve better results. When the percentage of consumed FEs is less than $\theta = 10\%$, G2SLMOEA emphasizes the convergence of the population by utilizing GAN to generate guiding solutions. Conversely, when the percentage of consumed FEs exceeds $\theta = 10\%$, the G2S strategy emphasizes the diversity of the population. Therefore, it can be observed that the IGD curve of G2SLMOEA exhibits multiple distinct descending phases, indicating that the G2S strategy effectively uses FEs to achieve better optimization performance. The convergence speed of FDV is slower than some algorithms. However, it still achieves good IGD values. Fig. 6(b) plots the IGD convergence curves of each algorithm on 3-objective 1000-dimensional LSMOP5. It can be observed that G2SLMOEA performs best. The convergence speed of LMOEADS and ALMOEA is faster than that of G2SLMOEA. However, G2SLMOEA ensures both the convergence and diversity of solutions. Therefore, in the region of the consumed FEs percentage from 10% to 20%, there is a noticeable decrease in the IGD values, achieving the best results. Fig. 6(c) shows the IGD convergence curves on 3-objective 1000-dimensional LSMOP8. Similarly, G2SLMOEA demonstrates a significant decrease in IGD values in the region of the consumed FEs percentage from 10% to 20% and gets the best optimization performance. It can be found from Fig. 6 that G2S guides evolution in terms of both convergence and diversity and can make full use of computing resources. G2SLMOEA first obtains a solution set with good convergence and diversity through GAN-generated guiding solutions and then focuses on diversity to achieve a well-balanced exploration and development. Additionally, the shaded areas in Fig.6 represent the standard deviation of IGD values for 20

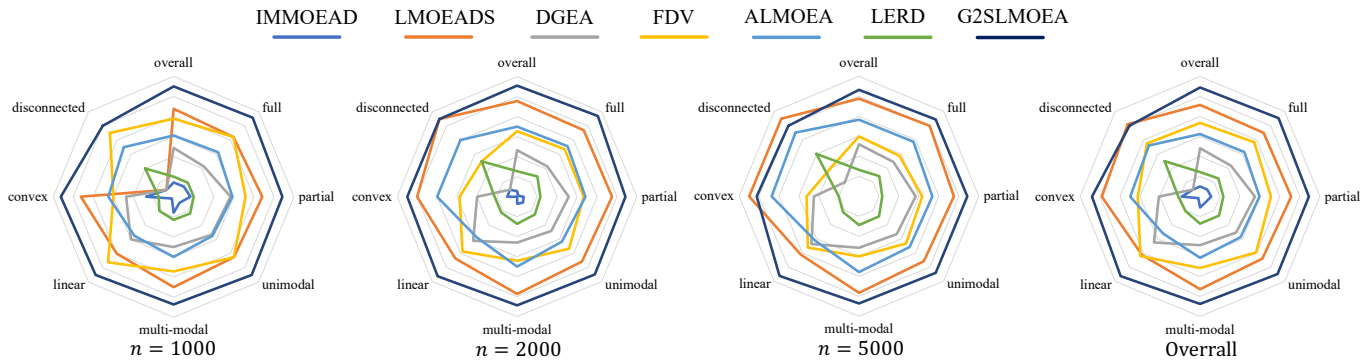
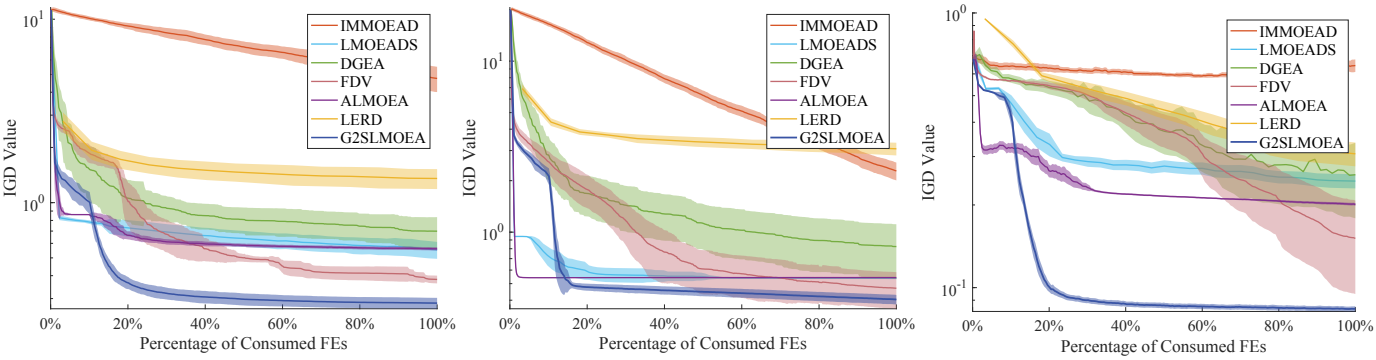


Fig. 5. Redar diagrams for the performance on different characteristics of LSMOP1-9 with $m = 2$ and 3. The first three radar diagrams are categorized performance on $n = 1000$, $n = 2000$ and $n = 5000$, respectively. The fourth diagram shows the overall performance. The data in each dimension is $|A| - r$, where r is the Friedman rank of the IGD for each category and $|A|$ is the number of all algorithms compared, $|A| = 7$. Some problems have mixed characteristics and therefore will be repeated when categorizing. $F_{\text{full}} = \text{LSMOP}1, 3, 4, 5, 7, 8$. $F_{\text{partial}} = \text{LSMOP}2, 3, 4, 6, 7, 8$. $F_{\text{unimodal}} = \text{LSMOP}1, 2, 4, 5, 6, 8, 9$. $F_{\text{multi-modal}} = \text{LSMOP}2, 3, 4, 5, 8, 9$. $F_{\text{linear}} = \text{LSMOP}1, 2, 3, 4$. $F_{\text{convex}} = \text{LSMOP}5, 6, 7, 8$. $F_{\text{disconnected}} = \text{LSMOP}9$.



(a) LSMOP1 (b) LSMOP5 (c) LSMOP8

Fig. 6. IGD convergence curves with standard deviation on 3-objective 1000-dimensional LSMOPs. The abscissae are the percentage of consumed FEs.

independent runs of the algorithms. It can be observed that G2SLMOEA has better stability.

In Fig. S3 (in Supplementary Document), DM [50] is used to evaluate the diversity of the population. The DM convergence curve demonstrates that the diversity of the population increases significantly when $\theta > 10\%$. G2S quickly finds good converged solutions in early generations and improves the diversity of solutions in later generations.

2) Runtime Comparison and Analysis: Fig. 7 provides the average runtime of each algorithm on 3-objective LSMOPs with n = 500, 1000, 2000 and 5000. ALMOEA has the longest average runtime because of training an MLP during evolution. LMOEADS and G2SLMOEA show the shortest overall runtime. G2S uses GAN only in the first stages of evolution to generate guiding solutions. That is, the runtime of G2SLMOEA can be decreased by decreasing the value of theta. In Fig. 7, θ is specified as $\theta = 10\%$.

3) Comparison Results on LMF Problems: To further validate the performance of G2SLMOEA, experiments are also carried out on 2-objective LMF1-8 with n = 1000, 3000 and 5000. In Table S8 (in Supplementary Document), G2SLMOEA achieves the best IGD in 22 out of the 24 problems. Only LMOEADS and FDV achieve performance close to G2SLMOEA on a few problems. On most problems, the performance of the six compared algorithms is poor when

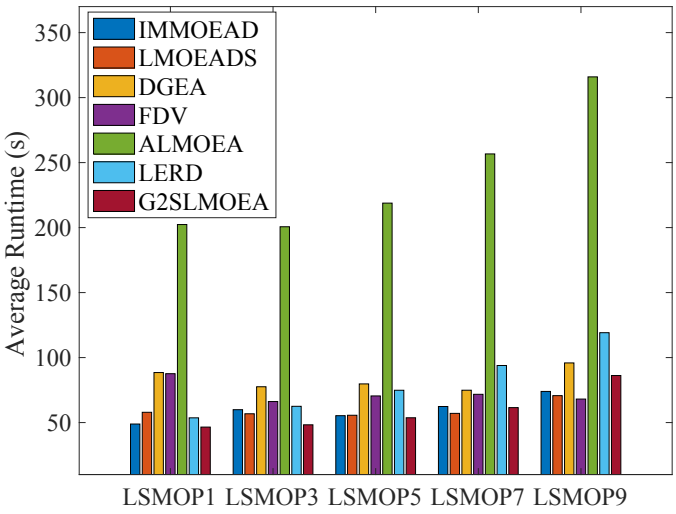


Fig. 7. The average runtime of G2SLMOEA and six compared algorithms on 3-objective 1000-dimensional LSMOPs.

solving LMF, which suggests that only G2SLMOEA can solve difficult LSMOPs. G2S uses GAN to generate the distribution of guiding points in the decision space and to guide evolution through guiding solutions. Friedman tests are conducted on

TABLE IV
THE HV RESULTS OF SIX COMPARED ALGORITHMS AND G2SLMOEA ON TREE1-5

Problem	n	IMMOEAD	LMOEADS	DGEA	FDV	ALMOEA	LERD	G2SLMOEA
TREE1	3000	NaN (NaN)	8.3689e-1 (7.25e-5) -	8.0153e-1 (2.26e-2) -	7.3691e-1 (2.40e-2) -	NaN (NaN)	7.8074e-1 (8.46e-3) -	**8.4070e-1 (2.38e-4)**
TREE2	3000	NaN (NaN)	8.4993e-1 (5.43e-5) -	8.3857e-1 (9.09e-3) -	7.7325e-1 (2.20e-3) -	NaN (NaN)	7.9765e-1 (4.01e-3) -	**8.5255e-1 (1.06e-3)**
TREE3	6000	NaN (NaN)	**8.8743e-1 (9.89e-8) +**	7.4798e-1 (1.63e-1) -	NaN (NaN)	NaN (NaN)	7.6939e-1 (1.47e-2) -	8.8742e-1 (1.00e-5)
TREE4	6000	NaN (NaN)	**9.6433e-1 (2.36e-5) +**	3.4506e-1 (3.70e-1) -	NaN (NaN)	NaN (NaN)	1.7635e-1 (1.37e-1) -	9.6424e-1 (3.61e-8)
TREE5	6000	NaN (NaN)	9.2728e-1 (1.56e-5) -	8.1376e-1 (8.23e-2) -	NaN (NaN)	NaN (NaN)	7.6002e-1 (2.41e-2) -	**9.2908e-1 (1.63e-3)**
$+/-/\approx$		0/5/0	2/3/0	0/5/0	0/5/0	0/5/0	0/5/0	—

the results of IGD with the significance level of $\alpha = 0.05$, $\chi^2 = 76.52$ and $p = 1.87 \times 10^{-14}$. G2SLMOEA achieves the top overall ranking considering all 24 test problems.

4) Comparison on Real-World Problems: The purpose of designing LMOEAs is to solve various complex LSMOPs for real-world applications. Time-varying ratio error estimation (TREE) [46] in real-world scenarios is selected as a test problem. G2SLMOEA is compared with state-of-the-art algorithms on TREE to verify whether G2SLMOEA has better performance in solving real-world LSMOPs. Since the true Pareto Front of the TREE problem is unknown, the methods from the original paper [46] are adopted to calculate IGD and HV. IGD results on the TREE benchmark are presented in the Supplementary Document.

Table IV gives the HV results of six compared algorithms and G2SLMOEA on TREE1-5, with the experimental results for TREE6 omitted since all algorithms fail to solve TREE6 efficiently at the given FEs. The "NaN" in Table IV indicates that the corresponding algorithm does not obtain a usable final solution set. The algorithm that is competitive with G2SLMOEA is LMOEADS, with higher HV values than G2SLMOEA on TREE3 and TREE4. However, the HV values of LMOEADS on these two problems are very close to those of G2SLMOEA, while the performance gap is significant on the remaining three problems. The performance of other algorithms in obtaining satisfactory solution sets under the given FEs is challenging, especially FDV and ALMOEA. Experimental results on TREE demonstrate that G2SLMOEA is more effective in solving complex real-world LSMOPs.

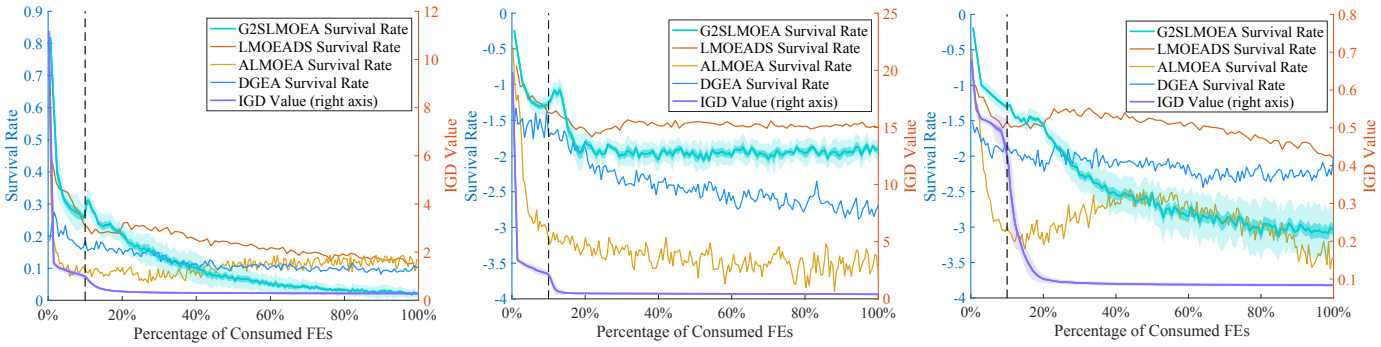
5) More Discussions: Fig. S4, Fig. S5, and Fig. S6 (in Supplementary Document) show the distribution of solutions of the eight algorithms on 3-objective LSMOP1, LSMOP4, and LSMOP8, which can observe the convergence and diversity of the solution set obtained by each algorithm more directly. IMMOEAD, LMOEADS, DGEA, ALMOEA, and LERD show poor convergence and diversity on LSMOP1 and LSMOP8. The diversity of IMMOEAD and LMOEADS on LSMOP4 is poor. The diversity of the solution set obtained by FDV is close to that of G2SLMOEA. However, as shown in Table II, G2SLMOEA's solution set is closer to the true Pareto front. The solution set obtained by G2SLMOEA is closer to the true Pareto front on these problems.

The performance of G2SLMOEA on LSMOP6 and LSMOP9 still has some shortcomings. The two problems are mixed modality, convex and partially separable or fully separable respectively. G2SLMOEA tends to fall into the local optima in solving these two problems.

To prevent mode collapse during the training process, G2SLMOEA uses GAN only in early generations ($\theta < 10\%$). Suppose the current value of simplex is large (i.e., $f_1 + \ldots + f_m = 1000$) and the number of associated reference vectors (i.e., each solution is associated with the reference vector with the smallest acute angle) is less than $1/3$ of the current population. This situation means that the convergence and diversity of the population are poor (Fig. S2 in Supplementary Document). When this happens, directly stop the training of GAN. G2SLMOEA can trigger this protection mechanism when solving LSMOP3 and LSMOP7. Training is stopped if the current population quality is very poor to prevent GAN from generating low-quality guiding solutions. After entering the second stage of G2S, nondominated solutions are used as guiding solutions. The effectiveness of this strategy is confirmed by the convergence curve in Fig. 6 and Table V. Therefore, the second-best IGD value is obtained on 2-objective LSMOP3 and LSMOP7, and the solution set obtained on the 3-objective problem has the best IGD value.

E. Ablation Study of G2S

Due to the distinct optimization emphasis of the G2S strategy before and after exceeding the threshold $\theta = 10\%$, a more detailed analysis is provided. When the percentage of consumed FEs is less than θ, GAN is used to generate guiding solutions, and an individual is associated with each guiding solution with the maximum cosine similarity, ensuring the convergence of the offspring population. When greater than θ, the nondominated solution set is used as guiding solutions, and individuals are associated with the guiding solution with the minimum cosine similarity, emphasizing the exploration of diversity. G2SLMOEA1 optimizes using only the first approach, while G2SLMOEA2 uses only the second approach. Table V presents the IGD results of G2SLMOEA1, G2SLMOEA2 and G2SLMOEA on 3-objective LSMOP1-9. In most problems, G2SLMOEA1 performs worse than G2SLMOEA but outperforms G2SLMOEA2. Due to the use of nondominated solutions as guiding solutions in G2SLMOEA2, its performance is notably poor in many problems. Providing accurate search directions in early generations, and prioritizing population convergence before exploring diversity, ensures that the G2S strategy can generate a high-quality offspring population. Fig. 8 shows the survival rate of offspring generated by each algorithm. G2S can generate good offspring with a high survival rate in early generations. The decreasing trend in the IGD curve further indicates the effectiveness of the G2S strategy. In the region of the first 10% FEs in Fig. 8, the survival rate of the offspring population generated by G2S



(a) LSMOP1 (b) LSMOP5 (c) LSMOP8

Fig. 8. Comparison of survival rate of generated offspring populations on 3-objective 1000-dimensional LSMOPs. The survival rate and IGD are depicted with solid curves representing the mean value, dark bands indicating the confidence interval, and light bands illustrating the standard deviation intervals. Only the IGD curve corresponds to the right ordinate, the remaining curves correspond to the left ordinate.

TABLE V
GAN GENERATION VERSUS NONDOMINATED SOLUTIONS AS GUIDING SOLUTIONS

Problem	n	G2SLMOEA1 ($\theta = 100\%$)	G2SLMOEA2 ($\theta = 0\%$)	G2SLMOEA ($\theta = 10\%$)†
LSMOP1	500	2.7054e-1 (3.03e-2) -	6.3028e-1 (1.82e-2) -	**1.7932e-1 (2.75e-2)**
	1000	3.7238e-1 (3.04e-2) -	6.7059e-1 (5.77e-2) -	**2.9024e-1 (2.10e-2)**
	2000	4.9391e-1 (4.61e-2) -	7.0855e-1 (2.65e-2) -	**3.4115e-1 (1.06e-2)**
	5000	6.0201e-1 (7.47e-2) -	7.1030e-1 (4.27e-2) -	**3.6525e-1 (5.75e-3)**
LSMOP2	500	4.6016e-2 (4.09e-4) -	6.3432e-2 (3.78e-4) -	**4.4429e-2 (5.66e-4)**
	1000	3.7444e-2 (1.57e-4) -	4.5010e-2 (8.72e-5) -	**3.7232e-2 (2.14e-4)**
	2000	3.3774e-2 (9.06e-5) -	3.7002e-2 (6.26e-5) -	**3.3693e-2 (2.31e-4)**
	5000	**3.1963e-2 (4.57e-5) =**	3.3308e-2 (2.65e-5) -	3.2012e-2 (1.09e-4)
LSMOP3	500	8.6071e-1 (8.44e-6) -	**8.3709e-1 (4.17e-2) =**	8.4438e-1 (3.06e-2)
	1000	1.1783e+0 (9.90e-1) -	8.5654e-1 (1.57e-2) =	**8.3756e-1 (4.09e-2)**
	2000	9.5986e-1 (4.35e-1) -	8.6066e-1 (1.84e-4) =	**8.6060e-1 (3.77e-4)**
	5000	3.2161e+0 (2.42e+0) -	8.6071e-1 (8.00e-6) =	**8.6066e-1 (1.96e-4)**
LSMOP4	500	1.0802e-1 (2.77e-3) -	1.0802e-1 (2.77e-3) -	**9.5285e-2 (3.49e-3)**
	1000	7.2676e-2 (9.21e-4) -	1.0696e-1 (1.42e-3) -	**6.5312e-2 (8.72e-4)**
	2000	5.0263e-2 (6.25e-4) -	6.8307e-2 (6.67e-4) -	**4.6981e-2 (7.14e-4)**
	5000	3.7477e-2 (2.56e-4) -	4.2794e-2 (1.27e-4) -	**3.6418e-2 (2.37e-4)**
LSMOP5	500	3.3942e-1 (3.74e-2) -	5.2279e-1 (2.50e-2) -	**2.7626e-1 (3.93e-2)**
	1000	4.2937e-1 (4.57e-2) =	5.4148e-1 (1.51e-3) -	**4.0977e-1 (3.79e-2)**
	2000	5.2206e-1 (1.45e-2) -	5.4223e-1 (5.03e-4) -	**5.0017e-1 (2.96e-2)**
	5000	6.3822e-1 (1.60e-1) -	5.4241e-1 (3.84e-4) -	**5.3948e-1 (1.43e-2)**
LSMOP6	500	1.3791e+0 (7.60e-2) -	1.2865e+0 (1.45e-2) =	**1.2863e+0 (1.34e-1)**
	1000	1.4999e+0 (1.38e-1) -	**1.3129e+0 (1.14e-3) =**	1.3314e+0 (3.57e-2)
	2000	1.1970e+2 (3.13e+2) -	1.3233e+0 (1.00e-3) -	**1.2594e+0 (1.89e-1)**
	5000	3.4629e+3 (7.64e+2) -	1.2995e+0 (1.33e-1) -	**1.2716e+0 (1.76e-1)**
LSMOP7	500	8.9569e-1 (5.43e-2) =	**8.3552e-1 (5.66e-2) =**	8.4469e-1 (4.69e-2)
	1000	8.8245e-1 (5.82e-2) -	**8.1408e-1 (6.67e-2) =**	8.2562e-1 (9.21e-2)
	2000	9.6777e-1 (5.21e-2) -	**7.8694e-1 (7.60e-2) =**	8.0185e-1 (7.68e-2)
	5000	9.6386e-1 (9.15e-4) -	7.9952e-1 (5.42e-2) =	**7.8567e-1 (9.24e-2)**
LSMOP8	500	1.2419e-1 (1.33e-2) -	2.0484e-1 (1.71e-2) -	**7.8900e-2 (1.01e-2)**
	1000	1.5311e-1 (1.62e-2) -	1.8642e-1 (2.75e-2) -	**8.0087e-2 (5.54e-3)**
	2000	1.6774e-1 (1.47e-2) -	1.8974e-1 (2.17e-2) -	**7.9364e-2 (3.78e-3)**
	5000	1.6563e-1 (2.43e-2) -	1.9394e-1 (5.12e-2) -	**7.8047e-2 (2.59e-3)**
LSMOP9	500	1.1764e+0 (8.19e-2) =	1.0658e+1 (1.22e+0) -	**1.0483e+0 (2.65e-1)**
	1000	**1.0643e+0 (2.39e-1) +**	1.7635e+1 (1.33e+0) -	1.1576e+0 (2.73e-1)
	2000	1.2662e+0 (2.32e-1) =	2.5403e+1 (1.78e+0) -	**1.1894e+0 (4.37e-1)**
	5000	**1.8299e+0 (7.87e-1) +**	3.3168e+1 (4.17e+0) -	3.5851e+0 (3.02e+0)
+/ − / ≈		0/26/10	2/29/5	—

† $\theta = 10\%$ is used in our experiments and is placed in the last column as the compared parameter.

ranges from high to low. When the percentage of consumed FEs exceeds 10%, G2S focuses on exploring the diversity. Correspondingly, the survival rate curve shows an increasing trend in the region where the percentage of consumed FEs is greater than 10%, and the IGD values also decrease more rapidly. The consistency between the survival rate curve and the IGD value curve indicates that the G2S strategy can sequentially accelerate the improvement of IGD values. Compared to other algorithms, G2S produces a high-quality offspring population with higher survival rates. The survival rate curves of ALMOEA and DGEA confirm our explanations in the motivation section, indicating their search directions are dependent on the population quality, resulting in a lower offspring survival rate. In the later part of the first 10% FEs in Fig. 8 (a) and (b), the survival rate of LMOEADS is higher than that of G2S. However, G2S still demonstrates an accelerated downward trend in IGD values after the algorithm used 10% of the total FEs.

TABLE VI
COMPARISON FOR STATISTICAL ANALYSIS OF DIFFERENT VALUES OF θ ON 3-OBJECTIVE LSMOP1-9

Parameter	IGD ranking ↓ p-value=5.99E-13	HV ranking ↑ p-value=3.75E-11	Runtime ranking ↓ p-value=3.93E-30
$\theta = 0\%$	4.67	2.24	**1.00**
$\theta = 25\%$	2.54	**4.44**	3.00
$\theta = 50\%$	2.72	4.11	4.00
$\theta = 75\%$	3.64	3.64	5.00
$\theta = 100\%$	5.03	2.28	6.00
$\theta = 10\%$†	**2.40**	4.29	2.00

† $\theta = 10\%$ is used in our experiments and is placed in the last row as the compared parameter.

F. Parameter Sensitivity Analysis of G2SLMOEA

The threshold value for the percentage of consumed FEs during GAN-guided evolution is a crucial parameter. Therefore, further experimental studies are conducted. Table VI presents the Friedman tests for IGD, HV and runtime with different values of θ on LSMOPs with $n = 500$, 1000, 2000 and 5000. Among them, $\theta = 10\%$ is the setting used in our experiments. The best values for IGD, HV, and runtime are achieved at $\theta = 10\%$, 25%, and 0%, respectively. When $\theta = 50\%$ and 75%, the performance is not the best, and the runtime is excessively long. Therefore it is reasonable to set the value of θ between 10% and 25%.

TABLE VII
COMPARISON OF IGD VALUES FOR DIFFERENT θ AND N_S ON 3-OBJECTIVE LSMOP1-9

Parameter	$N_S = N$	$N_S = 3N$	$N_S = 4N$	$N_S = 2N$†	p-value
$\theta = 25\%$	2.75	2.89	2.75	**1.61**	4.00E-05
$\theta = 50\%$	2.83	2.92	2.72	**1.53**	4.00E-06
$\theta = 75\%$	2.5	3.06	3	**1.44**	7.02E-08
$\theta = 10\%$†	2.43	2.82	3	**1.75**	1.59E-04

† $N_S = 2N$ is used in our experiments and is placed in the penultimate column as the compared parameter.

The number of guiding points sampled on the simplex N_S is also an important parameter. Table VII presents the Friedman tests for the IGD performance on different numbers of guiding points under various θ in solving 3-objective LSMOP1-9. The most effective training of GAN with $2N$ guiding points is achieved under different θ values in Table VII. The trained GAN generates guiding solutions in the decision space to guide the population to evolve. Then, each guiding solution is evaluated. Whereas the increase in the number of guiding points increases the number of input-output pairs for GAN training, it also increases the number of consumed FEs. Therefore, blindly increasing the number of guiding points may not be advisable. The value of N_S is set to $2N$, which balances learning the distribution of guiding points in the decision space and avoiding excessive FEs consumption.

V. CONCLUSIONS

This paper proposes a GAN-guided large-scale multiobjective evolutionary framework for solving LSMOPs. In order to guide the population to evolve efficiently, G2S uses GAN to learn the distribution of guiding points in the decision space to provide more precise search directions. The search direction of G2S is independent of the quality of the population and can generate high-quality guiding solutions in early generations. The G2S strategy of first ensuring convergence and then exploring diversity makes the population fully optimized. G2S embedded as a plug-and-play component into existing MOEAs improves the performance of solving LSMOPs. Our experimental results show that the effectiveness of the G2S strategy is thoroughly verified and G2SLMOEA has competitive performance on LSMOPs problems.

In future research, we will investigate more in-depth mechanisms of diversity optimization. In addition, improving the stability of optimizing LSMOPs with different characteristics by GAN is also an important research direction in the future.

REFERENCES

[1] J. Duan, Z. He, and G. G. Yen, "Robust multiobjective optimization for vehicle routing problem with time windows," *IEEE Transactions on Cybernetics*, vol. 52, no. 8, pp. 8300–8314, 2022.

[2] Z. Li and Z. Ding, "Distributed multiobjective optimization for network resource allocation of multiagent systems," *IEEE Transactions on Cybernetics*, vol. 51, no. 12, pp. 5800–5810, 2021.

[3] R. Zhang, S. Song, and C. Wu, "The hot strip mill scheduling problem with uncertainty: Robust optimization models and solution approaches," *IEEE Transactions on Cybernetics*, vol. 53, no. 7, pp. 4079–4093, 2023.

[4] W. Wei, R. Yang, H. Gu, W. Zhao, C. Chen, and S. Wan, "Multi-objective optimization for resource allocation in vehicular cloud computing networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 12, pp. 25 536–25 545, 2021.

[5] P. Wang, B. Xue, J. Liang, and M. Zhang, "Differential evolution-based feature selection: A niching-based multiobjective approach," *IEEE Transactions on Evolutionary Computation*, vol. 27, no. 2, pp. 296–310, 2022.

[6] H. Zhang, Y. Jin, and K. Hao, "Evolutionary search for complete neural network architectures with partial weight sharing," *IEEE transactions on evolutionary computation*, vol. 26, no. 5, pp. 1072–1086, 2022.

[7] S. Chen, S. Islam, A. Bolufé-Röhler, J. Montgomery, and T. Hendtlass, "A random walk analysis of search in metaheuristics," in *2021 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2021, pp. 2323–2330.

[8] Y. Tian, L. Si, X. Zhang, R. Cheng, C. He, K. C. Tan, and Y. Jin, "Evolutionary large-scale multi-objective optimization: A survey," *ACM Computing Surveys (CSUR)*, vol. 54, no. 8, pp. 1–34, 2021.

[9] L. M. Antonio and C. A. C. Coello, "Use of cooperative coevolution for solving large scale multiobjective optimization problems," in *2013 IEEE Congress on Evolutionary Computation*. IEEE, 2013, pp. 2758–2765.

[10] L. Miguel Antonio and C. A. Coello Coello, "Decomposition-based approach for solving large scale multi-objective problems," in *Parallel Problem Solving from Nature–PPSN XIV: 14th International Conference, Edinburgh, UK, September 17-21, 2016, Proceedings 14*. Springer, 2016, pp. 525–534.

[11] M. Li and J. Wei, "A cooperative co-evolutionary algorithm for large-scale multi-objective optimization problems," in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, 2018, pp. 1716–1721.

[12] X. Ma, F. Liu, Y. Qi, X. Wang, L. Li, L. Jiao, M. Yin, and M. Gong, "A multiobjective evolutionary algorithm based on decision variable analyses for multiobjective optimization problems with large-scale variables," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 2, pp. 275–298, 2015.

[13] X. Zhang, Y. Tian, R. Cheng, and Y. Jin, "A decision variable manifold-based evolutionary algorithm for large-scale many-objective optimization," *IEEE Transactions on evolutionary Computation*, vol. 22, no. 1, pp. 97–112, 2016.

[14] L. Ma, M. Huang, S. Yang, R. Wang, and X. Wang, "An adaptive localized decision variable analysis approach to large-scale multiobjective and many-objective optimization," *IEEE Transactions on Cybernetics*, vol. 52, no. 7, pp. 6684–6696, 2021.

[15] Q. Zhang and H. Li, "Moea/d: A multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on evolutionary computation*, vol. 11, no. 6, pp. 712–731, 2007.

[16] C. He, R. Cheng, L. Li, K. C. Tan, and Y. Jin, "Large-scale multiobjective optimization via reformulated decision variable analysis," *IEEE transactions on evolutionary computation*, 2022.

[17] H. Zille, H. Ishibuchi, S. Mostaghim, and Y. Nojima, "A framework for large-scale multiobjective optimization based on problem transformation," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 2, pp. 260–275, 2017.

[18] C. He, L. Li, Y. Tian, X. Zhang, R. Cheng, Y. Jin, and X. Yao, "Accelerating large-scale multiobjective optimization via problem reformulation," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 6, pp. 949–961, 2019.

[19] H. Qian and Y. Yu, "Solving high-dimensional multi-objective optimization problems with low effective dimensions," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, no. 1, 2017.

[20] Y. Tian, C. Lu, X. Zhang, F. Cheng, and Y. Jin, "A pattern mining-based evolutionary algorithm for large-scale sparse multiobjective optimization problems," *IEEE transactions on cybernetics*, vol. 52, no. 7, pp. 6784–6797, 2020.

[21] R. Liu, R. Ren, J. Liu, and J. Liu, "A clustering and dimensionality reduction based evolutionary algorithm for large-scale multi-objective problems," *Applied Soft Computing*, vol. 89, p. 106120, 2020.

[22] H. Chen, R. Cheng, J. Wen, H. Li, and J. Weng, "Solving large-scale many-objective optimization problems by covariance matrix adaptation evolution strategy with scalable small subpopulations," *Information Sciences*, vol. 509, pp. 457–469, 2020.

[23] C. He, S. Huang, R. Cheng, K. C. Tan, and Y. Jin, "Evolutionary multiobjective optimization driven by generative adversarial networks (gans)," *IEEE transactions on cybernetics*, vol. 51, no. 6, pp. 3129–3142, 2020.

[24] R. Cheng, Y. Jin, K. Narukawa, and B. Sendhoff, "A multiobjective evolutionary algorithm using gaussian process-based inverse modeling," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 6, pp. 838–856, 2015.

[25] L. R. Farias and A. F. Araújo, "Im-moea/d: an inverse modeling multi-objective evolutionary algorithm based on decomposition," in *2021 ieee international conference on systems, man, and cybernetics (SMC)*. IEEE, 2021, pp. 462–467.

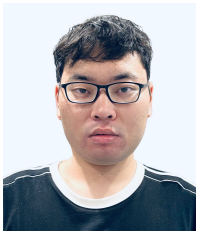
[26] Q. Deng, Q. Kang, L. Zhang, M. Zhou, and J. An, "Objective space-based population generation to accelerate evolutionary algorithms for large-scale many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 27, no. 2, pp. 326–340, 2022.

[27] S. Qin, C. Sun, Y. Jin, Y. Tan, and J. Fieldsend, "Large-scale evolutionary multiobjective optimization assisted by directed sampling," *IEEE transactions on evolutionary computation*, vol. 25, no. 4, pp. 724–738, 2021.

[28] Y. Tian, X. Zheng, X. Zhang, and Y. Jin, "Efficient large-scale multi-objective optimization based on a competitive swarm optimizer," *IEEE Transactions on Cybernetics*, vol. 50, no. 8, pp. 3696–3708, 2019.

[29] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.
[30] S. Liu, J. Li, Q. Lin, Y. Tian, and K. C. Tan, "Learning to accelerate evolutionary search for large-scale multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 27, no. 1, pp. 67–81, 2022.
[31] C. He, S. Huang, R. Cheng, K. C. Tan, and Y. Jin, "Evolutionary multiobjective optimization driven by generative adversarial networks (gans)," *IEEE transactions on cybernetics*, vol. 51, no. 6, pp. 3129–3142, 2020.
[32] Z. Wang, H. Hong, K. Ye, G.-E. Zhang, M. Jiang, and K. C. Tan, "Manifold interpolation for large-scale multiobjective optimization via generative adversarial networks," *IEEE Transactions on Neural Networks and Learning Systems*, 2021.
[33] Y. Tian, C. Lu, X. Zhang, K. C. Tan, and Y. Jin, "Solving large-scale multiobjective optimization problems with sparse optimal solutions via unsupervised neural networks," *IEEE Transactions on Cybernetics*, vol. 51, no. 6, pp. 3115–3128, 2021.
[34] S. Liu, Q. Lin, L. Feng, K.-C. Wong, and K. C. Tan, "Evolutionary multitasking for large-scale multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 27, no. 4, pp. 863–877, 2023.
[35] Z.-H. Zhan, J.-Y. Li, S. Kwong, and J. Zhang, "Learning-aided evolution for optimization," *IEEE Transactions on Evolutionary Computation*, vol. 27, no. 6, pp. 1794–1808, 2023.
[36] Y. Jiang, Z.-H. Zhan, K. C. Tan, and J. Zhang, "Knowledge learning for evolutionary computation," *IEEE Transactions on Evolutionary Computation*, pp. 1–1, 2023.
[37] J.-Y. Li, Z.-H. Zhan, K. C. Tan, and J. Zhang, "A meta-knowledge transfer-based differential evolution for multitask optimization," *IEEE Transactions on Evolutionary Computation*, vol. 26, no. 4, pp. 719–734, 2022.
[38] C. He, R. Cheng, and D. Yazdani, "Adaptive offspring generation for evolutionary large-scale multiobjective optimization," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 52, no. 2, pp. 786–798, 2020.
[39] X. Yang, J. Zou, S. Yang, J. Zheng, and Y. Liu, "A fuzzy decision variables framework for large-scale multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, 2021.
[40] R. Cheng, Y. Jin, M. Olhofer *et al.*, "Test problems for large-scale multiobjective and many-objective optimization," *IEEE transactions on cybernetics*, vol. 47, no. 12, pp. 4108–4121, 2016.
[41] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *Proceedings of the 34th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, D. Precup and Y. W. Teh, Eds., vol. 70. PMLR, 06–11 Aug 2017, pp. 214–223.
[42] H. Li and Q. Zhang, "Multiobjective optimization problems with complicated pareto sets, moea/d and nsga-ii," *IEEE transactions on evolutionary computation*, vol. 13, no. 2, pp. 284–302, 2008.
[43] E. Zitzler and S. Künzli, "Indicator-based selection in multiobjective search," in *International conference on parallel problem solving from nature*. Springer, 2004, pp. 832–842.
[44] H. Ge, M. Zhao, L. Sun, Z. Wang, G. Tan, Q. Zhang, and C. P. Chen, "A many-objective evolutionary algorithm with two interacting processes: Cascade manifold and reference point incremental learning," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 4, pp. 572–586, 2018.
[45] S. Liu, Q. Lin, K.-C. Wong, Q. Li, and K. C. Tan, "Evolutionary large-scale multiobjective optimization: Benchmarks and algorithms," *IEEE Transactions on Evolutionary Computation*, 2021.
[46] C. He, R. Cheng, C. Zhang, Y. Tian, Q. Chen, and X. Yao, "Evolutionary large-scale multiobjective optimization for ratio error estimation of voltage transformers," *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 5, pp. 868–881, 2020.
[47] A. Zhou, Y. Jin, Q. Zhang, B. Sendhoff, and E. Tsang, "Combining model-based and genetics-based offspring generation for multi-objective optimization using a convergence criterion," in *2006 IEEE international conference on evolutionary computation*. IEEE, 2006, pp. 892–899.
[48] L. While, L. Bradstreet, and L. Barone, "A fast way of calculating exact hypervolumes," *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 1, pp. 86–95, 2011.
[49] Y. Tian, R. Cheng, X. Zhang, and Y. Jin, "Platemo: A matlab platform for evolutionary multi-objective optimization [educational forum]," *IEEE Computational Intelligence Magazine*, vol. 12, no. 4, pp. 73–87, 2017.
[50] K. Deb and S. Jain, "Running performance metrics for evolutionary multi-objective optimizations," in *Proceedings of the Fourth Asia-Pacific Conference on Simulated Evolution and Learning (SEAL'02), Singapore*, 2002, pp. 13–20.

Hongwei Ge received the B.S. and M.S. degrees in mathematics from Jilin University, China, and the Ph.D. degree in computer application technology from Jilin University, in 2006. He is currently a professor with the College of Computer Science and Technology, Dalian University of Technology, Dalian, China. His research interests are artificial intelligence, machine learning, deep learning, swarm intelligence, intelligent system, etc. He has published more than 150 papers in prestigious journals, such as IEEE Transactions on Cybernetics, IEEE Transactions on Evolutionary Computation, IEEE Transactions on Intelligent Transportation Systems, IEEE Computational Intelligence Magazine, IEEE Transactions on Multimedia and IEEE Transactions on Circuits and Systems for Video Technology.



Zhi Zheng received the B.S. degree from the Dalian University of Technology, Dalian, China, in 2021, where he is currently pursuing the M.S. degree in computer science and technology with the Dalian University of Technology, Dalian, China. His research interests include evolutionary computation and machine learning.

Yaqing Hou (Member, IEEE) received the Ph.D. degree in artificial intelligence from Interdisciplinary Graduate School, Nanyang Technological University, Singapore, in 2017. He is currently an Associate Professor with the College of Computer Science and Technology, Dalian University of Technology, Dalian, China. His research interests include computational and artificial intelligence, memetic computing, transfer learning, and optimization. His publications have appeared in IEEE Transactions on Evolutionary Computation, and IEEE Computational Intelligence Magazine.

Xia Wang received the M.S. degree in computer science and technology from Henan Normal University, Henan, China, in 2019, where she is currently pursuing the Ph.D. degree in computer application technology with the Dalian University of Technology, Dalian, China. Her research interests include evolutionary computation, machine learning and swarm intelligence.

Hisao Ishibuchi (Fellow, IEEE) received the B.S. and M.S. degrees from Kyoto University, Kyoto, Japan, in 1985 and 1987, respectively, and the Ph.D. degree from Osaka Prefecture University, Sakai, Japan, in 1992. He is a Chair Professor with the Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen, China. His research interest is evolutionary multiobjective optimization.