

Multisource Selective Transfer Framework in Multiobjective Optimization Problems

Jun Zhang^{ID}, Weien Zhou, Xianqi Chen, Wen Yao, and Lu Cao

Abstract—For complex system design [e.g., satellite layout optimization design (SLOD)] in practical engineering, when launching a new optimization instance with another parameter configuration from the intuition of designers, it is always executed from scratch which wastes much time to repeat the similar search process. Inspired by transfer learning which can reuse past experiences to solve relevant tasks, many researchers pay more attention to explore how to learn from past optimization instances to accelerate the target one. In real-world applications, there have been numerous similar source instances stored in the database. The primary question is how to measure the transferability from numerous sources to avoid the notorious negative transferring. To obtain the relatedness between source and target instance, we develop an optimization instance representation method named centroid distribution, which is by the aid of the probabilistic model learned by elite candidate solutions in estimation of distribution algorithm (EDA) during the evolutionary process. Wasserstein distance is employed to evaluate the similarity between the centroid distributions of different optimization instances, based on which, we present a novel framework called multisource selective transfer optimization with three strategies to select sources reasonably. To choose the suitable strategy, four selection suggestions are summarized according to the similarity between the source and target centroid distribution. The framework is beneficial to choose the most suitable sources, which could improve the search efficiency in solving multiobjective optimization problems. To evaluate the effectiveness of the proposed framework and selection suggestions, we conduct two experiments: 1) comprehensive empirical studies on complex multiobjective optimization problem benchmarks and 2) a real-world SLOD problem. Suggestions for strategy selection coincide with the experiment results, based on which, we propose a mixed strategy to deal with the negative transfer in the experiments successfully. The results demonstrate that our proposed framework achieves competitive performance on most of the benchmark problems in convergence speed and hypervolume values and performs best on the real-world applications among all the comparison algorithms.

Manuscript received October 22, 2018; revised February 2, 2019, April 16, 2019, and June 24, 2019; accepted June 27, 2019. Date of publication July 1, 2019; date of current version May 29, 2020. This work was supported by the National Natural Science Foundation of China under Grant 11725211 and Grant 51675525. (Corresponding author: Wen Yao.)

J. Zhang is with the National Innovation Institute of Defense Technology, Chinese Academy of Military Science, Beijing 100071, China, and also with the Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China (e-mail: mcgrady150318@163.com).

W. Zhou, W. Yao, and L. Cao are with the National Innovation Institute of Defense Technology, Chinese Academy of Military Science, Beijing 100071, China (e-mail: wendy0782@126.com).

X. Chen is with the National University of Defense Technology, Changsha 410072, China.

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TEVC.2019.2926107

Index Terms—Estimation of distribution algorithm (EDA), multiobjective optimization, multisource transfer, transfer optimization, Wasserstein distance (WD).

I. INTRODUCTION

FOR COMPLEX system design problems in practical engineering, taking satellite system design for example, there are plenty of past experiences such as the different design of satellite layout solutions have been stored in the database before launching a new design. Exploiting past experiences are not only enhance optimization effectiveness but also improve the convergence rate. It is very important for complex system design as generally involving computationally expensive multidisciplinary analysis [1]. However, how to exploit the knowledge to accelerate the new design confuses the satellite designers. Transfer learning is concerned with the shift in data distribution from one domain to another. Various research has been applied to classical machine learning tasks successfully, e.g., classification task [2]–[4], like sentiment analysis [5], and digits recognition [6]. In recent years, researchers in the evolutionary algorithm community attempt to employ transfer learning to the optimization task [7]–[10].

When applying transfer learning methods to solve the optimization problems, three issues should be considered as follows.

Q1 (Transferability): In real-world case studies, a large number of similar problems have been solved and saved in the database. The critical question is how to identify the transferability of all the sources. Negative transfer occurs if the selected sources are not suitable.

Q2 (Transfer Component): In the following step, the transfer component of the problem should be determined, such as solutions, structures, features, parameters or something else.

Q3 (Transfer Algorithm): How to reuse the information extracted from the source problems is the final step. Most research focuses on one-to-one domain adaptation algorithms, such as transfer component analysis (TCA) [11], TrAdaBoost [12] in benchmarks like MNIST dataset [13] and WIFI dataset [14]. However, very few research concentrates on multisource transfer learning problems.

In transfer learning research field, most of the works are focused on Q2 and Q3, especially on Q3. But, Q1 is also very important as transferring from inappropriate sources will lead to negative transfer. In this paper, we pay special attention to the transferability of multisource problems and investigate how to measure the similarity of different optimization problems.

Liu *et al.* [15] showed that relatedness of tasks is crucial to the effectiveness of multitask learning in his work. Feng *et al.* [16] developed an evolutionary multitasking method based on autoencoder, and tasks in the experiments are chosen relied on two factors: 1) degree of the intersection of the Pareto optimal solutions and 2) similarity in the fitness landscape between tasks. The factors guarantee the relatedness. However, in actual scenes, we cannot obtain the information as mentioned above of the target problem at the beginning.

To evaluate the relatedness of different optimization problems, representation of different optimization instances should be determined first. Santana *et al.* [17] figured the optimization problem as a graph via the estimation of Bayesian networks algorithm (EBNA), which is one type of the estimation of distribution algorithms (EDAs). In his work, a task is represented by a graph. The relevance of tasks could be obtained by the network analysis method, but the representation method depends on human-made graph features very much, which may miss relevant information on tasks. Yu *et al.* [18] invented a new representation method of the reinforcement learning tasks with different parameters sampled by the uniform distributions. He uses reward vectors obtained by prototype policies via few iterations named shallow trials to represent the task, to make the past reinforcement learning tasks reusing. However, shallow trials in the evolution would make too many noises to represent the optimization instance accurately. Utilizing the population distribution produced by EDA in each generation, we can represent the instance by the compositive distribution, which is called centroid distribution here. Centroid means the centralization of all the population in each generation of evolution. Similar instances have similar centroid distribution, based on which, the similarity between different instances could be obtained easily via the Wasserstein distance (WD). By the similarities, we can make use of the three source selection strategies to produce transferred knowledge to the GA-style algorithm like NSGAII, to accelerate the searching process during the evolution. To use the strategy effectively, four selection suggestions are summarized.

As a summary, this paper makes the following contributions.

- 1) Invent a novel representation method named centroid distribution to measure the similarity of different optimization instances.
- 2) Propose a novel multisource selective transfer optimization framework with three strategies based on EDA and NSGA-II [19] to solve multiobjective optimization problems.
- 3) Summarize four suggestions of source selection strategy, based on which, we propose a mixed strategy to deal with the negative transfer issue.

The rest of this paper is organized as follows. A related work is surveyed in transfer optimization, EDAs, and evolutionary dynamic optimization (EDO) are presented in Section II. The general framework and workflow of multisource selective transfer optimization (MSSTO) based on EDA and NSGA-II is introduced in details in Section III. Section IV provides the comprehensive empirical studies on the commonly used multiobjective evolutionary benchmarks and summarizes four

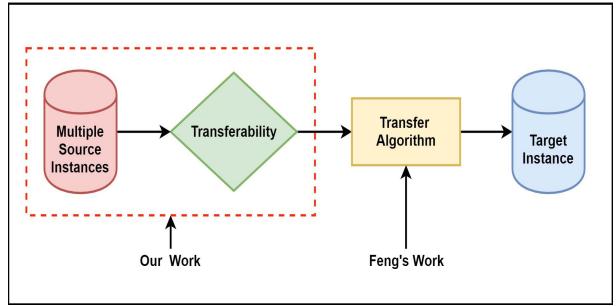


Fig. 1. Illustration of an MSSTO workflow.

using suggestions of source selection strategy. A real-world design optimization problem of satellite layout is used to evaluate our proposed framework and validate the selection suggestions in Section V. Finally, the concluding remarks of this paper are discussed in Section VI.

II. RELATED WORK

A. Transfer Optimization

In practical applications, engineers always restart a new optimization instance when the constraints and objective function parameters meet with a slice modification. Indeed there are many common features among these instances, searching from scratch will result in repeated computation. Recently, many works focusing on improving the efficacy of existing evolutionary algorithms through machine learning techniques are developed, especially reusing past search experiences from related problems. Feng *et al.* [7] proposed a new transfer method based on the autoencoder model to solve the multiobjective optimization problems, which accelerates the search process in the evolutionary algorithm. Santana *et al.* [10] puts forward four types of transfer learning methods when solving the multimarker tagging single-nucleotide polymorphism (SNP) selection problems, namely, the transfer of solutions, structure transfer, behavioral or algorithmic transfer, and combined transfer, and any combination of the previous three alternatives.

Transfer optimization's mathematical definition is given in Gupta's survey [20], which is defined as follows: given a knowledge base consists of numerous past experiences $M = \cup_{k \in K} m_k$, and target optimization instance (T), transfer optimization improves performance speedup measured as $Q_t(T|M) - Q_t(T) \geq 0$, where $Q_t(T|M)$ is the algorithmic efficiency based on the knowledge embedded in M .

Research in transfer optimization can be divided into two categories according to the source quantity, namely, single source and multisource. Plenty of research works concentrate on one-to-one transfer algorithms in popular benchmarks, but multisource problems are more common in practical applications. Multisource transfer optimization task is concentrated on in this paper. As illustrated in Fig. 1, a general multisource transfer optimization workflow usually contains source selection and transfer algorithm. Source selection problem represents the transferability, including the representation of instance and source selection strategy. Here, we focus on

Algorithm 1 EDAs General Procedure

-
- 1: $t = 0$.
 - 2: Initialize model $MO(0)$ to represent uniform distribution over admissible solutions
 - 3: **while** termination criteria not reaches **do**
 - 4: $P = \text{generate } N > 0 \text{ candidate solutions by sampling } MO(t)$
 - 5: $F = \text{evaluate all candidate solutions in } P$
 - 6: $MO(t + 1) = \text{adjust_model}(P, F, MO(t))$
 - 7: $t = t + 1$
 - 8: **end while**
-

studying the transferability especially instance representation method and multisource selection strategy. Feng's work [7] is taken as our transfer method.

B. Estimation of Distribution Algorithms

EDAs are stochastic optimization methods that guide the search for the optimum point by learning and sampling an explicit probabilistic model of elite candidate solutions. The main difference between EDAs and conventional evolutionary algorithms (GA-style algorithms) is that the latter one generates new candidate solutions via the mutation and crossover operation, which can be viewed as an implicit distribution. However, EDAs use an explicit distribution represented by a probabilistic model, such as multivariate normal distribution, Bayesian network, etc.

The general procedure of EDAs is outlined as Algorithm 1.

Based on the decision variable interdependence, EDAs are classified into three categories [21].

- 1) Univariate factorizations, which assumes each variable is mutually independent. The representative algorithms consist of population-based incremental learning (PBIL) [22], univariate marginal distribution algorithm (UMDA) [23], compact genetic algorithm (cGA) [24], etc.
- 2) Bivariate factorizations, in this class of models, dependencies between variables form a tree or forest graph. The representative algorithms contain mutual information maximizing input clustering (MIMIC) [25], EDA based on dependency trees and bivariate marginal distribution algorithm (BMDA) [26], etc.
- 3) Multivariate factorizations which represent dependencies using either directed acyclic graphs or undirected graphs. Two representative models are Bayesian networks [27] and Markov networks [28].

EDAs have been employed to numerous challenging optimization problems in recent years, especially in multiobjective optimization problems [29]–[33] and multimodal optimization problems [34], [35]. By the aid of the probabilistic model learned from elite candidate solutions, many types of research [36] focus on how to reuse the model from past experiences to accelerate the target instance searching. Inspired by the transfer learning research in EDA, we propose a novel method to represent instance called centroid distribution. Based on the representation, we propose three

source selection strategies to produce proper knowledge to transfer.

C. Evolutionary Dynamic Optimization

EDO is one of the most active research areas in the field of evolutionary computation. In certain cases of EDO, some common patterns could be found with the changes of environments. Inspired by this, a lot of research focus on reusing the past search experiences from the similar environments to accelerate the evolution process [37], [38].

One common approach is learning to predict the movement of optimum based on some special points. Li *et al.* [39] proposed a predictive strategy based on the special points (SPPS), such as feed-forward center point, boundary point, and knee point, which can do well in EDO. Ruan *et al.* [40] proposed a hybrid diversity maintenance method to improve prediction accuracy based on the center points. Zou *et al.* [41] proposed a method to predict the nondominated set based on the knee points to make population quickly converge to PF.

Another approach is to predict the locations which individuals or population should be reinitialized when a change occurs. Zhou's work [42] utilizes history search experiences to guide similar target search to solve the dynamic optimization problems efficiently. Two reinitialization strategies are proposed, one is predicting the new location of individual according to the history; the other one is perturbing the current population by some noise whose variance estimated by the previous changes.

The third kind of approach named memory-based method is often used for solving the EDO problems via reusing the relevant Pareto-optimal set (POS) information stored in the memory when the environment changes periodically [43]–[46]. Jiang and Yang [37] proposed a new method called steady-state and generational evolutionary algorithm (SGEA), which reuses some past solutions with good distribution and relocates some solutions related to the new Pareto front based on the information of the previous and current environments when a change is detected. Besides, the multipopulation schemes are useful methods to solve the problems [47], [48].

Some recent works integrated the transfer learning algorithms such as TCA with the dynamic optimization algorithms directly. Jiang presents two algorithm frameworks in this way. One is called Tr-DMOEA which integrates transfer learning and population-based evolutionary algorithms [8], and the other one is the combination of domain adaptation and nonparametric estimation-based EDA, called DANE-EDA [9]. Different from his works, we mainly focus on how to select the suitable source from multiple source tasks to speed up the evolutionary process.

For EDO methods, the information of past generations during the optimization process is reused to facilitate the optimum search in the upcoming generations. In EDO process, the optimization problem formulation is unchanged except for some environmental parameters. But for transfer optimization, it aims to reuse the information from past optimization instances, which are totally independent of current target optimization task. How to properly judge the

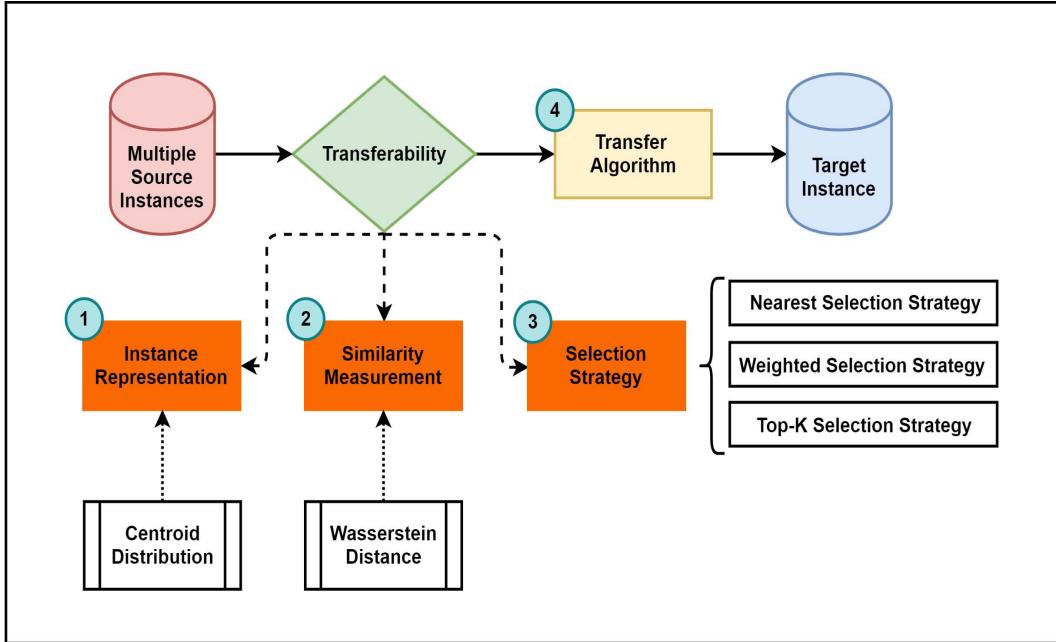


Fig. 2. Comprehensive illustration of an MSSTO workflow.

relevance between the sources and the target, based on which the appropriate sources can be selected to accelerate the target task, is an important and challenging issue. To solve this problem, this paper is focused on the similarity measurement methods and source select strategies, which will be discussed in Section III.

III. MULTISOURCE SELECTIVE TRANSFER OPTIMIZATION FRAMEWORK

In most practical applications [such as satellite layout optimization design (SLOD) problems], massive design instances have been launched, executed, and stored in the database for reuse. However, most researchers are more interested in one-to-one transfer optimization and neglect the multisource property in the real-world scenes. In this paper, we present a multisource selective transfer optimization framework to solve the many-reusable-instance problems.

EDAs are one type of black box optimization algorithms which have been introduced in Section II-B. Generic algorithms (GA-style) are heuristic-based searching algorithms, which are always inspired by human beings action or the other animals' action, free to the model during the searching process and depend on the crossover and mutation operation to find the better solution in the space. EDA performs better in describing the population distribution than GA-style via the explicit probabilistic model and GA-style retains an implicit one.

To solve the multisource transfer optimization problems in real-world applications, we combine the two types of optimization algorithms. As shown in Fig. 2, instance representation is introduced first. Next, the source-target similarity measurement method would be presented. Three source selection strategies are explored in the Section III-C. Then the

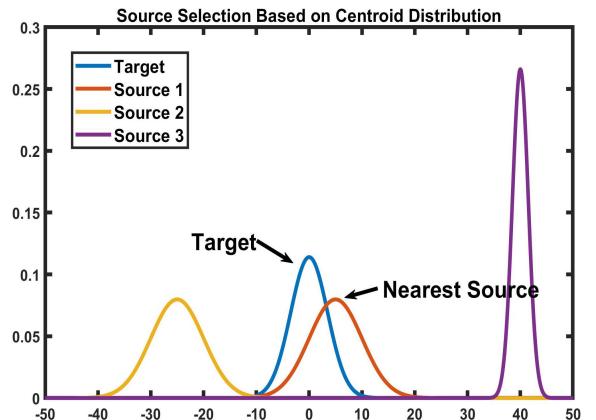


Fig. 3. Every optimization instance is represented by multivariate normal distribution.

transfer method is shown in Section III-D. Last, our MSSTO is presented.

A. Optimization Instance Representation

To represent optimization instance during the searching process, we propose a novel method called centroid distribution (see Fig. 3). With GA-style algorithm running, we can obtain the elite population through selection operation, the distribution of selected solutions can be learned as an explicit probabilistic model of EDA. We call the learned model centroid distribution, to represent the current population dispersal trending in the space.

Denote the centroid distribution as $C \sim N(\mu, \sigma^2)$, which is independent multivariate normal distribution, $\mu = (\mu_1, \mu_2, \dots, \mu_n)$, $\sigma^2 = (\sigma_1^2, \sigma_2^2, \dots, \sigma_n^2)$, where n represents the variable dimension, μ_i denotes the mean of i th dimension, and σ_i^2 represents the variance of i th dimension. Each variable

is independent of each other

$$\mu_i = \frac{1}{N} \sum_{j=1}^N (\text{elite}_{ij}) \quad (1)$$

$$\sigma_i^2 = \frac{1}{N-1} \sum_{j=1}^N (\text{elite}_{ij} - \mu_i)^2 \quad (2)$$

where elite denotes the selected population via the selection operation in GA-style algorithm. i represents the i th dimension and j means the j th individual. N denotes the population size.

To avoid the notorious negative transfer, suitable sources should be determined before the transfer algorithm is launched. Our proposed representation method could reflect some similarity information between the source and target during the evolution, which can assist target instance to choose suitable sources to transfer.

B. Source Target Similarity Measurement

WD is a distance function defined between probability distributions on a given metric space [50]. In this paper, we use WD to calculate the distance between the source and target centroid distribution.

Since centroid distribution C obeys the independent multivariate normal distribution law, WD between the source and target centroid distribution can be computed by

$$\text{WD}(C_s, C_t) = \sqrt{\|\mu_s - \mu_t\|_2^2 + \|\sigma_s - \sigma_t\|_2^2} \quad (3)$$

where C_s represents the source centroid distribution and C_t represents the target centroid distribution. μ_s denotes the mean vector of source distribution and μ_t denotes the mean vector of the target distribution. σ_s means the standard deviation of the source distribution and σ_t means the standard deviation of the target distribution.

The nearer distance of centroid distributions is, the more related instance is. We define the similarity between source and target instance as follows:

$$\text{sim}(s, t) = \frac{1}{\text{WD}(C_s, C_t)} \quad (4)$$

where $\text{sim}(s, t)$ denotes the similarity.

Kullback–Leibler divergence [51] can also be used for measuring the distance between two distributions, which will be investigated in the future.

C. Source Selection Strategy

Due to the different optimization methods are used or different stopping conditions are used, different results may be obtained even for the similar optimization tasks. Besides, for the complex highly nonlinear and multimodal problems with disconnected local optimal regions, the optimization results may be quite different even with the same optimization algorithms. Thus, given the sources which are quite different, they may be actually from the similar tasks. Actually, given the vast source data and without *a priori* knowledge about the detailed optimization formulation and the optimization solver used for each source, it is really difficult to correctly judge whether the

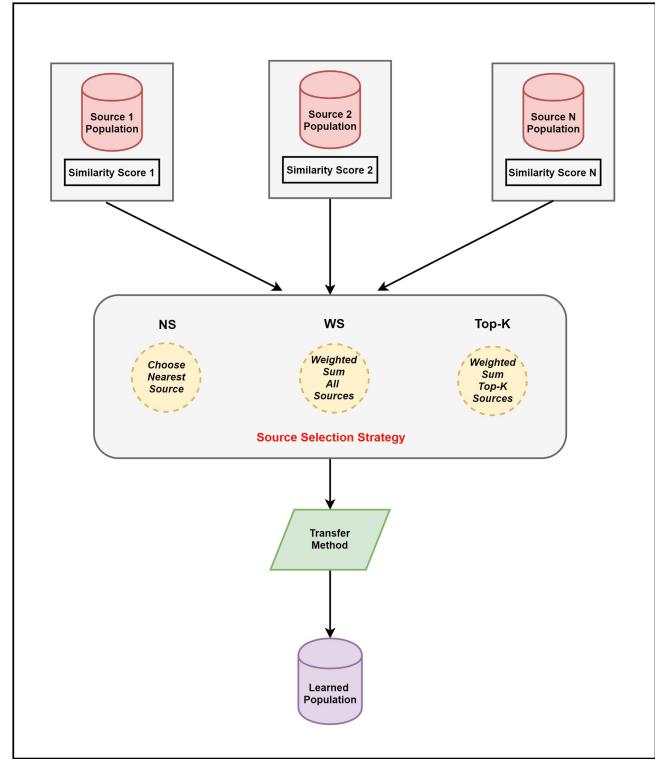


Fig. 4. Three strategies for recommending suitable sources to transfer method are proposed, including NSS, WSS, and Top-K.

dissimilarity between the source and the target results from the dissimilarity between the tasks or not. Therefore, in this paper, we propose the selection strategies that “the nearer distance of centroid distributions is, the more related instance is,” which actually is a sufficient but not necessary condition. If the distance is near, we judge that the instances are related and can be selected. But in the opposite, if the instances are related, we cannot judge that the distribution will be close.

Based on the analysis above, three source selection strategies are proposed to generate the transferred population to the target instance based on the similarities, see Fig. 4. Four suggestions of strategy selection are summarized in part 4.

1) *Nearest Selection Strategy*: During the evolution, the similarity between each source and target population can be calculated by 2-norm WD in (4). When all the similarities between sources and target centroid distributions are obtained, the strategy selects the nearest source as the transferred knowledge to target. We name it nearest selection strategy (NSS).

Denote the population in source instances as $P_{i(D \times N)}$, where $i = 1, 2, \dots, \text{num}_s$, num_s represents the number of source problems, N denotes the population size, and D denotes the variable dimension. When the transfer operation is called by the framework over the evolutionary process, NSS will take the largest score source as transferred one in the current generation, as follows:

$$P_{\text{transfer}} = P_{\arg \max_i(\text{sim}(s_i, t))} \quad (5)$$

where s_i means the i th source instance and t means the target instance. P_{transfer} denotes the transferred population produced

by NSS and $\arg \max_i(\text{sim}(s_i, t))$ represents the largest similarity score source index.

In NSS, we note that the most similar source contains the most useful knowledge to the target instance.

2) *Weighted Selection Strategy*: In certain circumstances, some sources may be located in the same similarity level almost. If we use one source to transfer, some useful information about other sources would be missed. Under the situation, a new strategy is proposed to exploit the knowledge of all the sources. Because of different sources making different level contributions to the transfer effectiveness, final source population is a weighted sum of all sources in the current generation. This strategy is called weighted selection strategy (WSS). Denote the source weight as w_i , the transferred population is given by

$$P_{\text{transfer}} = \sum_{i=1}^{\text{num}_s} w_i P_i \quad (6)$$

where

$$w_i = \frac{\text{sim}(s_i, t)}{\sum_{i=1}^{\text{num}_s} (\text{sim}(s_i, t))}. \quad (7)$$

Here, w_i is a scalar and P_i is a matrix.

In WSS, we note that every source is helpful to the target instance, but different sources have the different level of importance, reflected by source weights here.

3) *Top-K Selection Strategy*: Top-K selection strategy (Top-K) can be regarded as the particular form of WSS. In this situation, some dissimilar sources are eliminated, and the K most similar ones are chosen to transfer. All the calculation methods are the same as WSS.

In this strategy, the transferred source set is obtained as follows:

$$P_{\text{transfer}} = \sum_{i \in S} w_i P_i \quad (8)$$

where S denotes the index set of the K most similar sources.

4) *Suggestions for Strategy Selection*: In this part, four suggestions are summarized to guide to choose the suitable selection strategy under certain conditions.

First, a metric called max similarity rate is defined as follows:

$$\text{msr} = \frac{\max(\text{sim}(s_i, t))}{\sum_{i=1}^{\text{num}_s} (\text{sim}(s_i, t))} \quad (9)$$

where msr denotes the max score ratio which represents the ratio between the highest similarity score and the sum of all the similarity scores. When the selected source has a higher msr, it is more similar to the target than any other sources obviously. Under this situation, NSS could lead to better performance improvements.

Then, the variances of similarities between different sources are used for determining whether to choose WSS or Top-K. The variance determines the volatility of distribution. Smaller variances mean the equalization of all the sources, based on which, taking all the sources to combine into a learned population via WSS would lead to better performance. Larger

variances imply that part of sources is suitable to transfer via Top-K.

Based on the above analysis, suggestions of choosing source selection strategy can be summarized as follows.

- 1) When the msr is relatively high, we should make use of NSS to select one suitable source.
- 2) When the msr is relatively low, and the variances of similarities are small, we could take WSS as our selection strategy.
- 3) When the msr is relatively low, and the variances of similarities are large, the Top-K strategy is our best choice.
- 4) When the msr is relatively low, and the variances locate at the critical range between the WSS and Top-K strategy, the transfer operation is not recommended to call.

Based on the above suggestions, we propose a mixed version of the three strategies called mix selection strategy (MSS). To explore whether to transfer in each generation during the evolution, the source selection strategy part would be triggered every time. Whether to transfer or not would be determined by the suggestions above according to the similarity information between all the sources and target. More similarity calculation brings more computational cost, but it also exploits more information indeed. A separate experiment will also be conducted to evaluate MSS.

D. Transfer Method

In the whole framework pipeline, transfer method is used to receive final source population from three strategies in some generation. Then it produces the learned population via the mapping method proposed by Feng *et al.* [7], finally sends them to the selection operation of the GA-style as the candidate population. By using Feng's transfer method, for one thing, it can be injected into GA-style algorithm easily. For another, the mapping method can obtain a closed-form solution and is easy to solve during the evolution.

The source population and target population could be connected by one single layer denoising autoencoder [7] with solid theoretical derivations. As shown in Fig. 5, the key of autoencoder-based transfer method is learning a mapping from source to target population. We can obtain the closed-form mapping M by solving

$$L_{sq}(M) = \frac{1}{2N} \text{tr}[(P_t - MP_{\text{transfer}})^T (P_t - MP_{\text{transfer}})] \quad (10)$$

where $\text{tr}(\cdot)$ denotes the trace operation of a matrix, $L_{sq}(M)$ represents the squared reconstruction loss from the source population space to target, P_{transfer} represents the transferred population from selection strategies, and P_t denotes the target population during the evolution. So mapping M between transferred population P_{transfer} obtained by selection strategies in Section III-C and target population P_t is given by

$$M = (P_t P_{\text{transfer}}^T) (P_{\text{transfer}} P_{\text{transfer}}^T)^{-1}. \quad (11)$$

However, the learned mapping is changed with the evolution. The optimized solutions from past problems contain vast knowledge to improve the target optimization searching

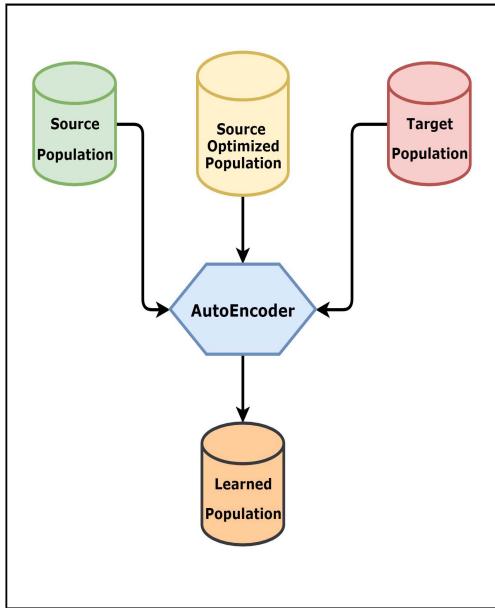


Fig. 5. Transfer method proposed by Feng which is based on autoencoder.

speed so that we can transfer the best solutions in the previous experiences into the target by the learned mapping.

Thought here is very similar to works in deep learning area which is popular in natural language processing [52], [53], computer vision tasks [54], [55], etc. The learned population can be obtained by

$$P_{\text{learned}} = MP_{s,\text{optimized}} \quad (12)$$

where $P_{s,\text{optimized}}$ denotes the best population in source instance for strategy NSS and P_{learned} represents the learned population via the transfer method. Especially for WSS and Top-K, best population and P_{transfer} are the weighted sums of all suitable sources.

E. Proposed Framework

Based on the four parts above, our MSSTO is proposed. As shown in Fig. 6, MSSTO inherits from the conventional evolutionary algorithm. The most important difference is that selected candidates consist of three parts: 1) parent population from the last generation; 2) offspring population from mutation and crossover operation; and 3) learned population from past experiences by the transfer method. The learned population is injected into the selection operation of GA-style.

The general procedure of MSSTO is outlined as Algorithm 2. G in step 5 denotes the transfer cycle and Feng sets it as 10 in his work [7].

After the initialization, the framework starts a loop with the population reproduction, transfer and selection operation. Transfer occurs when the generation can divide exactly G . During the transfer operation, the source and target instance are represented by centroid distribution defined by (1) and (2) first. Then the similarity score between the source and target could be calculated by (3) and (4). Next, suitable source would be selected according to (5)–(9). Last, the learned population

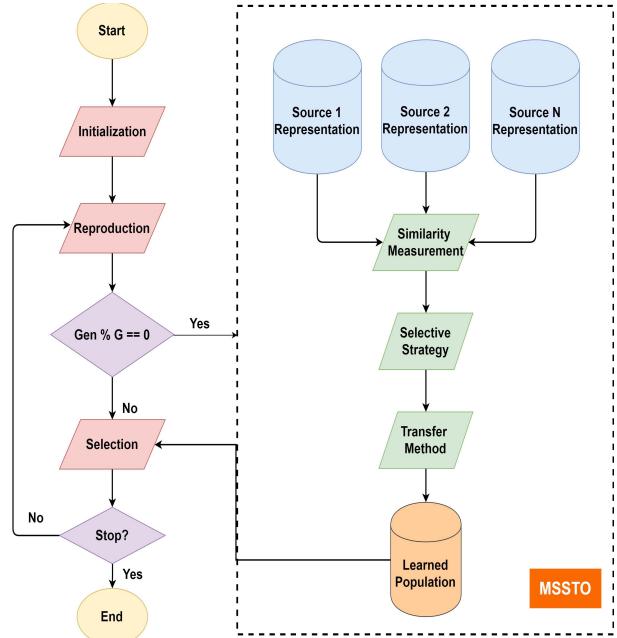


Fig. 6. Illustration of an MSSTO framework procedure.

Algorithm 2 MSSTO Framework General Procedure

- 1: Set $Gen = 0$.
 - 2: Initialization.
 - 3: **while** condition is not satisfied **do**
 - 4: Reproduction, including crossover and mutation operation.
 - 5: If $Gen \bmod G == 0$, go to step 6, else go to step 11.
 - 6: Transfer, execute Step 7-10.
 - 7: Represent source and target instances with centroid distribution defined by the formulas (1)(2) in Section III-A.
 - 8: Calculate the similarity score by the formulas (3)(4) in Section III-B.
 - 9: Select suitable source according to the formulas (5)–(9) in Section III-C.
 - 10: Produce the learned population by the formulas (10)(11) in Section III-D.
 - 11: Population Selection.
 - 12: $Gen = Gen + 1$
 - 13: **end while**
-

to transfer is produced by (10) and (11). The loop stops when the final fitness evaluation is done.

Three selection strategies defined in (5)–(9) could produce the final source population for the later process with transfer method. The strategies may not only improve optimization effect but also increase the convergence speed, so as to decrease the fitness evaluation quantity. Fewer evaluations in real-world applications mean fewer simulations or experiments which can save a lot of computational or experimental cost.

IV. EMPIRICAL STUDY

In this section, the experiments are setup up to validate the proposed framework.

Before a new optimization instance is launched in experiments, a large number of similar synthetic instances should be run by the same algorithm. Here the “similar” means that instances objectives and constraints have the same form, but the parameters in objectives and constraints are different. To simplify the research, parameters are sampled from uniform distribution here. We empirically evaluate MSSTO, particularly, addressing the following points.

- 1) Can MSSTO with three source selection strategies accelerate new similar instance searching process via the past experiences?
- 2) Are the suggestions of strategy selection effective?
- 3) Can the mixed strategy based on the suggestions solve the negative transfer issues?

A. Experiment Settings

Twelve commonly used multiobjective problem (MOP) benchmarks, consisting of five MOPs in ZDT family problems with two objectives [56] and seven MOPs in DTLZ family problems with three objectives [57] are considered here. For comparison, a popular multiobjective evolutionary algorithm (MOEA) named NSGA-II [19] and Fengs transferred method based on autoencoder (NSGAII+AE) [7] are employed as our baseline MOP solvers. NSGA-II is also used as GA-style in MSSTO.

In Feng’s method, the past experiences come from different problems. When the ZDT problem is regarded as the target, other problems in the ZDT family or problems in the DTLZ family will be regarded as the source problems without selection. In this paper, to validate the relatedness measurement method via the distance of source population distributions and the effectiveness of MSSTO, we synthesize new similar problems based on the given target problem through some transformation tricks. Transformation means that the optimization problem template keeps constant, but the parameters in objectives and constraints change randomly. Every target instance can accumulate knowledge learned from ten similar sources which are transformed by the target via a random disturbance to the parameters in objectives and constraints.

In the experimental results, we denote the comparison algorithms as follows.

- 1) NSGAII denotes the NSGAII algorithm.
- 2) NSGAII+AE denotes the algorithm proposed by Feng.
- 3) NSGAII+ST1 denotes the proposed algorithm with the NSS.
- 4) NSGAII+ST2 denotes the proposed algorithm with the WSS.
- 5) NSGAII+ST3 denotes the proposed algorithm with the Top-K.
- 6) NSGAII+MIX denotes the proposed algorithm with the MSS.

Furthermore, our experiments are developed based on the evolutionary multiobjective optimization platform (PlatEMO) [58], which is an open source and free MATLAB-based platform for evolutionary multiobjective optimization, that includes more than 60 existing MOEAs and 100 popular

benchmark MOPs, and the specific experimental settings are outlined as follows.

- 1) Population size N is configured as 50 in all the algorithms.
- 2) Evolutionary operators in NSGAII [19]: SBX crossover: $p_c = 0.9$, $\eta_c = 20$, polynomial mutation: $p_m = (1/n)$, $\eta_m = 20$.
- 3) Maximum function evaluation is MaxFES = 75000, and the stopping criterion is reaching the maximum function evaluation.
- 4) Independent number of runs is runs = 20.
- 5) Interval of injecting solution obtained by source problem learning is $G = 10$.
- 6) Source instance parameters distribution. For the given target instance, similar sources would be synthesized via sampling the parameters in objectives and constraints. Suppose the value of the objective parameter is r , the parameters are sampled concerning the uniform distribution of $[r - 5, r + 5]$.
- 7) Source instance number $\text{num}_s = 10$, each target instance has ten similar sources synthesized by the random processing method.
- 8) Top 30% source instances are selected in the Top-K strategy.

Other parameters are kept as the default values in PlatEMO [58].

In the evaluation step, all the algorithms are evaluated by the metric hypervolume (HV), which is the only single set quality measure that is known to be strictly monotonic with regards to Pareto dominance [59], [60]. The motivation of this paper is trying to improve Feng’s work (NSGAII+AE). For comparing easily, we take the same indicator as Feng’s work, namely HV.

The numerical experiments include three steps. First, the benchmark test problems are solved by the proposed MSSTO framework with three different source selection strategies, respectively and compared against the standard NSGAII result. Second, the source selection suggestions are evaluated by the intermediate process of some typical benchmark test problems. Third, the MSS based on the suggestions would be evaluated by one problem which occurs negative transfer.

B. Experimental Results

1) Evaluation of MSSTO With Three Strategies on Benchmark: Five algorithms are compared in the experimental study, consisting of one classical multiobjective algorithm NSGA-II and four variants of NSGA-II. In ZDT/DTLZ MOPs, all algorithms execute 20 independent runs with 75 000 function evaluations. The results are tabulated in Table I. The best HV values are bold in Table I.

As illustrated in Table I, MSSTO, including NSGAII-ST1, NSGAII-ST2, and NSGAII-ST3 achieve nine champions in all of the 12 MOPs, NSGA-II wins three remained problems. Here, the champion is defined as achieving the most considerable HV value. Since the standard deviation magnitude is much smaller than HV value, it could be neglected in the comparison. Our framework has achieved similar performance as

TABLE I
HV MEDIAN AND STANDARD DEVIATION VALUES OBTAINED BY NSGAII, NSGAII+AE, NSGAII+ST1, NSGAII+ST2, AND NSGAII+ST3 ON ZDT/DTLZ MOPs ACROSS 20 INDEPENDENT RUNS WITH 75 000 FUNCTION EVALUATIONS

MOPs	NSGAII	NSGAII+AE	NSGAII+ST1	NSGAII+ST2	NSGAII+ST3
ZDT1	8.627E-1±6.0E-7	8.514E-1±3.7E-5	8.633E-1±1.3E-6	8.637E-1±1.1E-6	8.632E-1±7.0E-7
ZDT2	5.297E-1±7.0E-7	5.149E-1±1.0E-4	5.305E-1±9.0E-7	5.286E-1±4.8E-6	0.530E-1±2.5E-6
ZDT3	1.016E-0±9.0E-5	1.006E-0±2.0E-5	1.019E-0±3.0E-7	1.019E-0±1.0E-7	1.019E-0±3.0E-7
ZDT4	8.584E-1±7.7E-6	8.622E-1±4.6E-6	8.640E-1±2.2E-6	8.640E-1±1.1E-6	8.647E-1±3.0E-7
ZDT6	4.277E-1±2.0E-7	4.172E-1±1.5E-5	4.280E-1±2.0E-7	4.270E-1±5.0E-7	4.277E-1±1.0E-6
DTLZ1	1.334E-1±0.0E-0	1.312E-1±0.0E-0	1.330E-1±0.0E-0	1.332E-1±0.0E-0	1.324E-1±0.0E-0
DTLZ2	6.697E-1±2.0E-7	6.564E-1±1.0E-7	6.683E-1±1.0E-7	6.705E-1±2.0E-7	6.697E-1±1.0E-7
DTLZ3	6.739E-1±2.0E-7	6.681E-1±2.0E-7	6.757E-1±2.0E-7	6.736E-1±2.0E-7	6.732E-1±1.0E-7
DTLZ4	5.646E-1±5.2E-5	5.673E-1±1.6E-5	6.818E-1±1.0E-7	5.944E-1±4.2E-5	6.236E-1±3.0E-5
DTLZ5	1.301E-1±0.0E-0	1.168E-1±0.0E-0	1.297E-1±0.0E-0	1.298E-1±0.0E-0	1.298E-1±0.0E-0
DTLZ6	1.304E-1±0.0E-0	1.159E-1±0.0E-0	1.300E-1±0.0E-0	1.293E-1±0.0E-0	1.297E-1±0.0E-0
DTLZ7	1.490E-0±6.3E-6	1.390E-0±5.4E-6	1.503E-0±4.0E-7	1.473E-0±6.1E-6	1.500E-0±3.0E-7

NSGA-II on some problems, e.g., DTLZ1, DTLZ2, DTLZ5, DTLZ6, DTLZ7, ZDT2, ZDT3, and ZDT6, but MSSTO shows superior performance over NSGA-II and NSGAII+AE on ZDT1, ZDT4, DTLZ3, and DTLZ4, which demonstrates our framework can learn the best population among all the comparison algorithms.

In all problems we evaluate, MSSTO consistently outperforms NSGAII-AE, which means selective and combination strategies in our framework can learn more useful knowledge to guide the search of target instance evolution than transferring from all the sources without source selection operation in NSGAII-AE.

Compared with our three proposed strategies, ST1 wins in six problems, ST2 wins three problems, while ST3 wins twice. The selective strategy performs better than combining with all the sources strategy in most of the situations, ST2 is on par with ST3 which means combining all the sources has a similar effect as mixing the Top-K in all the MOPs here. Though the results show that ST1 is the best, we think three strategies adapt to different situations. We will make a further study on the issue later.

To access the efficiency of our proposed framework, we plot the convergence graphs of the NSGA-II, NSGAII-AE and three NSGAII variants on the representative MOPs. In particular, Fig. 7 shows the convergence graphs of NSGAII, NSGAII+AE, NSGAII+ST1, NSGAII-ST2, and NSGAII-ST3. In the figures, the *x*-axis denotes the fitness evaluations which represents the algorithm efficiency, and fewer function evaluations imply a better computation performance. *Y*-axis denotes the mean HV values obtained across 20 independent runs.

Compared with NSGAII and NSGAII-AE, MSSTO with three different selection strategies achieves the competitive performance in most of the MOPs. NSGAII-ST1 outperforms the other two strategies in most MOPs of ZDT family, but in DTLZ MOPs, NSGAII-ST3 is best-performing.

Furthermore, we track the transfer process of compared algorithms during the evolution. As illustrated in Fig. 8, we define a metric called average contribution rate to evaluate the transfer effectiveness of NSGAII+AE and our MSSTO. The *x*-axis denotes the evolutionary generation, and the *y*-axis denotes the average contribution rate obtained across 20 independent runs, which is given as follows:

$$\text{acr}_i = \frac{N_{i,\text{selected}}}{N_{i,\text{total}}} \quad (13)$$

where acr_i means the average contribution rate of the *i*th generation during evolution, N_{selected} represents the size of selected population in *i*th generation, and N_{total} denotes the total size of population sent by the transfer method in *i*th generation. Here, we choose ZDT1 as our tracking problem to validate the algorithm transfer effect. As seen in Fig. 8, NSGAII+AE performs worst among all of the transferred algorithms, NSGAII+ST1 achieves the best performance, and NSGAII+ST2 performs on par with NSGAII+ST3. This explains the observations shown in Fig. 7(a) and Table I that NSGAII+ST1 achieves the best convergence speed, and NSGAII+AE performs worst in HV value. NSGAII+ST2 has achieved similar performance as NSGAII+ST3 in both HV value and convergence speed.

2) *Evaluation of Source Selection Suggestions*: To validate the suggestions of choosing the different source selection strategies, we make a further study on the similarities among all the sources and target.

First, we consider when to use NSGAII+ST1 during the evolution. Observed from Fig. 7(d)–(f) and (h) that on ZDT4 and ZDT6, NSGAII+ST1 performs better than the other two algorithms, but on DTLZ1 and DTLZ3, NSGAII+ST1 achieves the worse performance than the other two. So we choose ZDT4, ZDT6, DTLZ1, and DTLZ3 as our experimental problems. As shown in Fig. 9, the *x*-axis denotes the evolutionary generation, and the *y*-axis denotes the max similarity rate obtained across 20 independent runs.

On ZDT4 and ZDT6, the msr values are significantly higher than the performance on DTLZ1 and DTLZ3, which brings into correspondence with the previous results. It is suggested when the msr is relatively high, using the WSS and Top-K would lead to a slower convergence speed.

Then, when to use NSGAII+ST2 is considered in our discussion. Observed from Fig. 7, we find that NSGAII+ST2 performs better than NSGAII+ST3 on DTLZ1, but NSGAII+ST3 performs better on ZDT4 and ZDT6. In Fig. 10, the *x*-axis denotes the evolutionary generation, and the *y*-axis denotes the variances of sources obtained across 20 independent runs. For convenient comparison, we plot the logarithm of variances in the figure. As shown in Fig. 10, the variance values on ZDT4 and ZDT6 are several orders of magnitudes greater than DTLZ1. It concludes when the msr is relatively low and the variances of similarities are large, WSS should be chosen. Contrarily, the Top-K would be the most suitable choice. If NSS is selected under this condition, it will miss more useful knowledge.

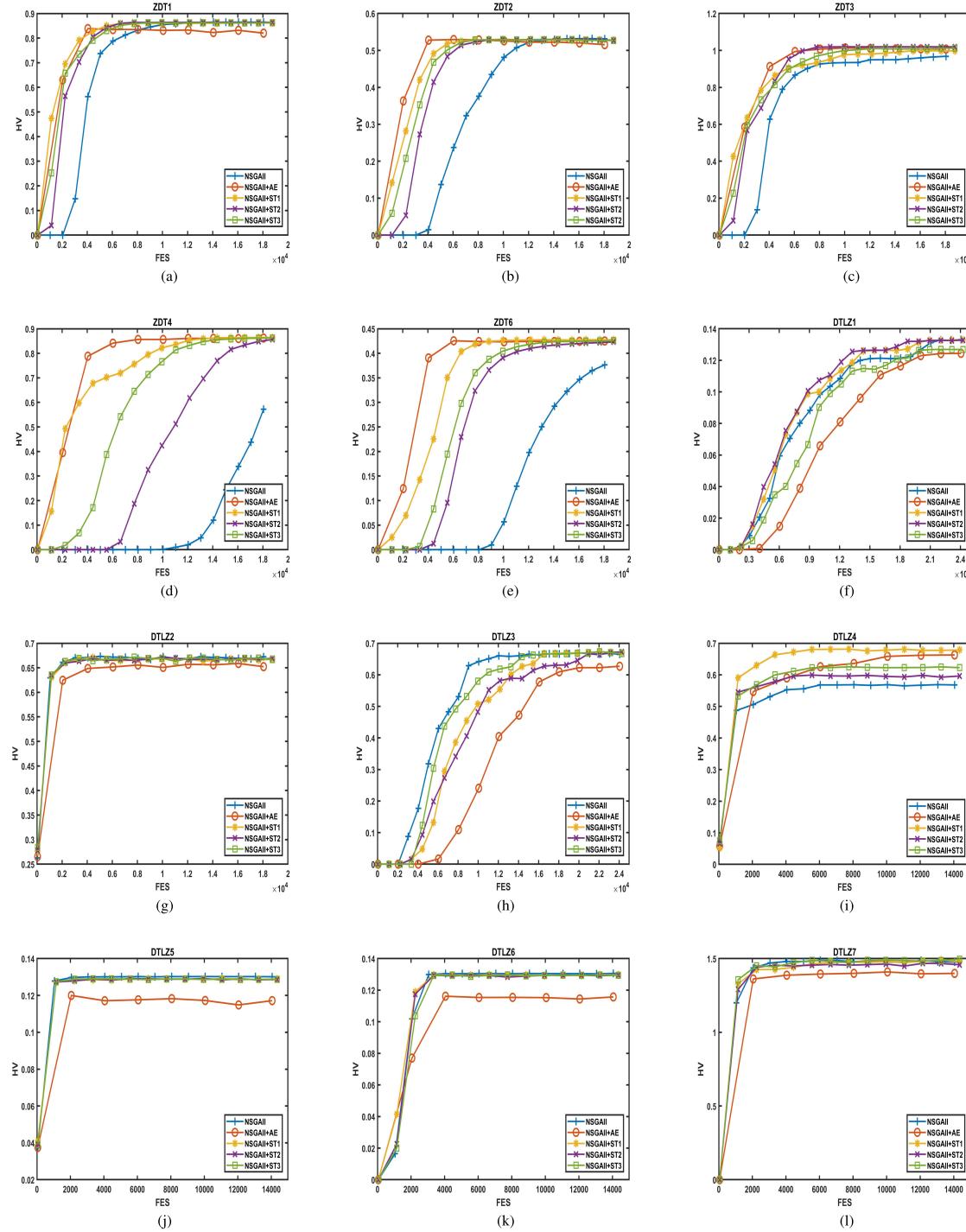


Fig. 7. Mean HV values obtained by NSGAI, NSGAI+AE, NSGAI+ST1, NSGAI+ST2, and NSGAI+ST3 with respect to FES on representative ZDT/DTLZ MOPs across 20 independent runs. (a) ZDT1. (b) ZDT2. (c) ZDT3. (d) ZDT4. (e) ZDT6. (f) DTLZ1. (g) DTLZ2. (h) DTLZ3. (i) DTLZ4. (j) DTLZ5. (k) DTLZ6. (l) DTLZ7.

3) Evaluation of Mix Selection Strategy: To evaluate NSGAI+MIX, a separate experiment is conducted on DTLZ3. Consider that on DTLZ3, the negative transfer occurs in all of the transferred algorithms. In Fig. 11, x -axis and y -axis have the same meaning as Fig. 7. In this experiment, we set three hyperparameters to denote the msr threshold, the upper and lower bound of the variance according to the strategy selection suggestions. Different from the previous

experiments, the transfer strategy part is called in each generation. In each generation, the strategy aims to decide whether to transfer [here principle (4) means no transfer occurs] and which strategy should be taken if the transfer is needed. NSGAI+MIX is a dynamic decision process during the evolution. It is designed for taking the best decisions in each generation according to the current information. As illustrated in Fig. 11, NSGAI+MIX achieves better convergence speed

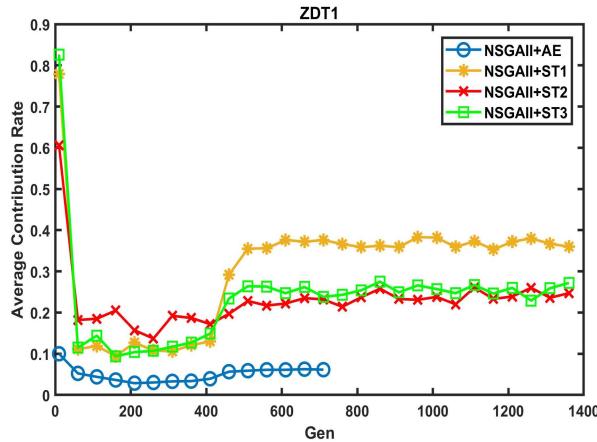


Fig. 8. Average contribution rate varies along with the evolutionary process on ZDT1.

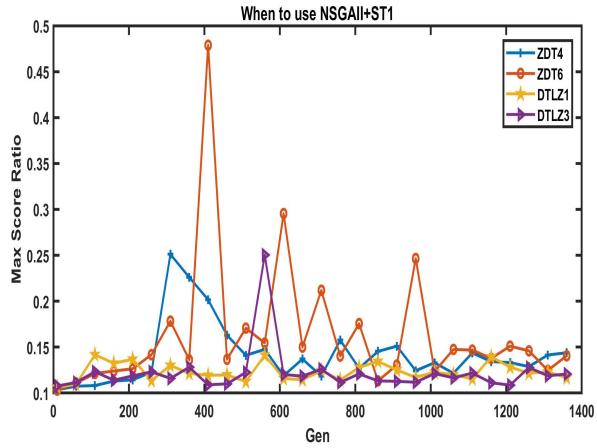


Fig. 9. Max similarity score ratio between the selected source and target via NSGAII+ST1 on ZDT4, ZDT6, DTLZ1, and DTLZ3.

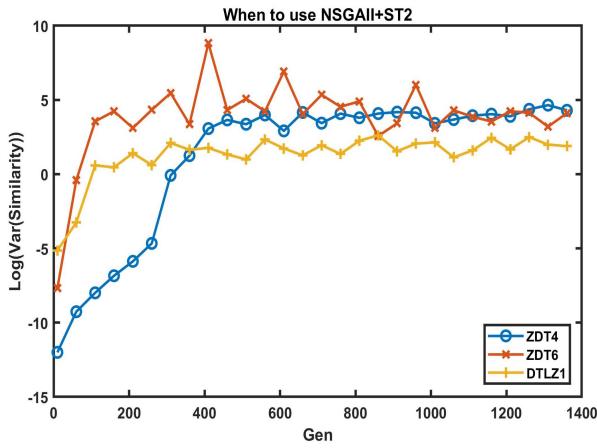


Fig. 10. Variance of similarity score between the selected source and target via NSGAII+ST1 on ZDT4, ZDT6, and DTLZ1.

and HV values than all the algorithms (including NSGAII). It concludes that our summarized suggestions can be used in the source selection problems well.

4) Discussion: After the empirically evaluations, we can answer the three questions at the beginning of this section.

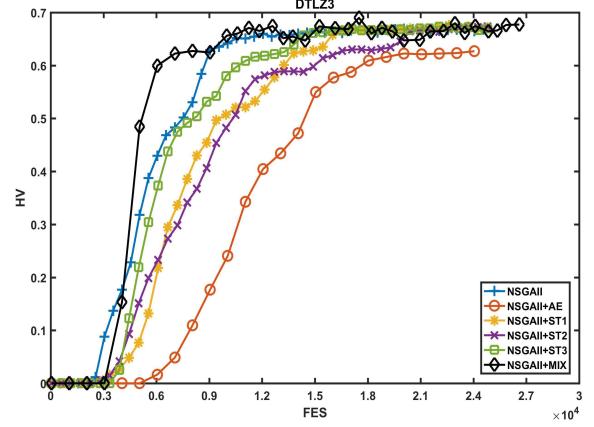


Fig. 11. HV value on DTLZ3 with NSGAII+MIX compared to the above algorithms across independent 20 runs.

- 1) Observed from Fig. 7 and Table I, it is convinced that MSSTO can accelerate the new similar instance by our proposed three source selection strategies in most situations.
- 2) Observed from Figs. 8–10, we validated the effectiveness of our strategy suggestions. Based on the suggestions, we propose a mixed strategy NSGAII+MIX, to solve the negative transfer issue occurs in the DTLZ3 problem successfully (see in Fig. 11).

This paper is a preliminary research on multisource transfer optimization. In this paper, we explore the simplest homogeneous situation. Here, the homogeneity means the same variable number and type in both source and target instance. In the real-world engineering scenes, our proposed method can be used well. Take SLOD task in Section V for example, the design is processed iteratively. In different design phases, the objectives and constraints would change slightly, but the design variables (the locations of the components) are kept unchanged. We can accumulate large amounts of historical optimization experiences to reuse in the new design phase. In the future research, we will expand our method to the heterogeneous situation which is more complex.

V. APPLICATION ON SATELLITE LAYOUT OPTIMIZATION DESIGN PROBLEM

In this section, we consider the application of the proposed framework on SLOD problems. In particular, this application aims to carry out the computationally demanding simulation-based optimization of two different satellite layout design tasks. It would be another critical improvement in shortening the satellite design cycle and enhancing the quality of satellite layout design, thereby playing a pivotal role in realizing the intelligent system design for aerospace vehicles.

The purpose of satellite layout design is to place the satellite components in the proper position and orientation so that various engineering requirements or constraints can be satisfied to ensure the satellite in a normal and smooth in-orbit operation state [61]. In most engineering cases, such layout schemes can be determined through repeated adjustment and modification,

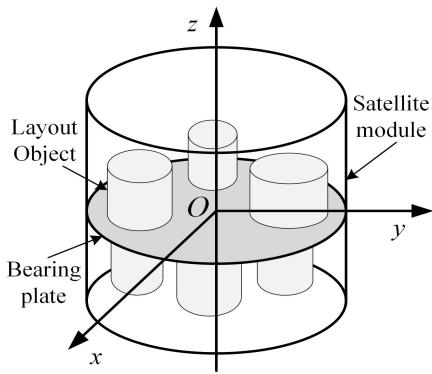


Fig. 12. Illustration of the SLOD model.

which dramatically extends the design cycle and thus hampers the rapid deployment of micro- and nano-satellites. Under this circumstance, the SLOD problem is proposed as an automatic optimization approach to generate some better layout candidate schemes for engineers to choose from [62].

The SLOD problem is a 3-D layout optimization problem with complex performance constraints (see Fig. 12). It has a nonlinear and nonconvex layout design space with numerous local optima, which is known as a class of NP-hard problems [63]. The SLOD model concerning the satellite mass characteristics, attitude dynamics issues, and thermal performance are considered in this real-world application. This model has two objectives: one is to minimize the moments of inertia of the whole satellite so that the satellite attitude controllability can be enhanced as much as possible, and the second is to minimize the nonuniformity of satellite heat flux so that a uniform thermal distribution within the satellite can be reached. The second objective function is represented using the intersection area of the active areas determined by the component dissipation powers, which is first proposed by Hengeveld *et al.* [64] and later combined in the SLOD research [62]. The corresponding layout constraints can be summarized as the performance constraints (system centroid constraint and inertia angle constraint) and the primary geometric constraint (nonoverlap constraint).

Assume that there are total N cylindrical layout components to be placed in the simplified cylindrical satellite module as shown in Fig. 12. All these components are installed on two surfaces of the bearing plate which is fixed to the shell of the satellite. The SLOD model in this section mainly focuses on optimizing the component centroid coordinates within the 2-D layout domain. Therefore, the bi-objective constrained SLOD model can be formulated as follows:

$$\begin{aligned}
 & \text{Find } X = \{X_1, X_2, \dots, X_N\} = \{X_i = (x_i, y_i) | i = 1 \dots N\} \\
 & \text{Min } \begin{cases} f_1(X) = J_x(X) + J_y(X) + J_z(X) \\ f_2(X) = \sum_{k=1}^n \left(\sum_{i=1}^{N_k-1} \sum_{j=i+1}^{N_k} \text{int}(A_{ij}) \right) \end{cases} \\
 & \text{s.t. } \begin{cases} g_1(X) = \sum_{i=0}^{N-1} \sum_{j=i+1}^N \Delta V_{ij} \leq 0 \\ g_2(X) = |x_c - x_e| - \delta x_e \leq 0 \\ g_3(X) = |y_c - y_e| - \delta y_e \leq 0 \\ g_4(X) = |\theta_x(X)| - \delta \theta_x \leq 0 \\ g_5(X) = |\theta_y(X)| - \delta \theta_y \leq 0 \\ g_6(X) = |\theta_z(X)| - \delta \theta_z \leq 0 \end{cases} \quad (14)
 \end{aligned}$$

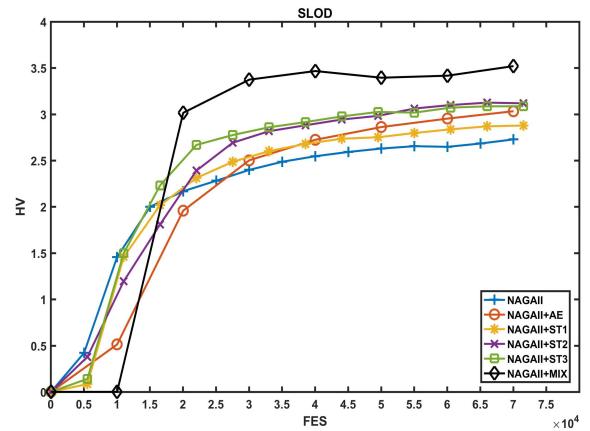


Fig. 13. Mean HV values obtained by NSGAII, NSGAII+AE, NSGAII+ST1, NSGAII+ST2, NSGAII+ST3, and NSGAII+MIX with respect to FES on SLOD across 20 independent runs.

where X represents the design variables and refers to the layout scheme to be optimized. f_1 means the summation of the moments of inertia of the whole satellite. n denotes the number of layout surfaces and here $n = 2$. N_k is the number of layout components that are placed on the k th ($k = 1 \dots n$) layout surface. According to [64], f_2 can represent the heat flux nonuniformity approximately. Regarding the nonoverlap constraint g_1 , ΔV_{ij} ($i, j > 0, i \neq j$) refers to the amount of overlap between components. When $i = 0$, the component i denotes the satellite module shell, and it means the container protrusion is also not allowed. g_{23} mean the system centroid error constraints, where δx_e and δy_e are the maximum permissible centroid errors and set as 3 mm here. And $g_{4,5,6}$ describe the inertia angle constraints, where $\delta \theta_x$, $\delta \theta_y$, and $\delta \theta_z$ are the maximum permissible inertia angles and set as 0.03 rad. Their relevant calculation can be found in [62].

Hence, a multiobjective multiconstraint numerical optimization problem is formed to represent the SLOD case in this paper.

Noted that when calculating the HV values, the reference set representing the approximation of the ideal and nadir points is set as (28.0, 3.6). Observed from Fig. 13 and Table II (best result is bold), NSGAII+MIX based on the selection suggestions is top-performing, and our proposed framework with three strategies achieves better performance than NSGAII+AE and NSGAII, respectively. It is also verified the effectiveness of our summarized selection suggestions. However, the standard deviation of NSGAII+MIX is larger than other methods a lot, which means the mixed method is not stable. We need to take much time to tune the hyperparameters. We will make a further study on the stableness of NSGAII+MIX in the future work.

Based on the selection from multisource instances in real-world applications, the target instance can achieve an impressive boosting in HV values and convergence speed. The effect of reusing more knowledge from similar instances is powerful for practical applications in the actual scenes, e.g., SLOD problem. Better searching the performance implies the saving more time and costs.

TABLE II

HV MEDIAN AND STANDARD DEVIATION VALUES OBTAINED BY NSGAI^I, NSGAI^I+AE, NSGAI^I+ST1, NSGAI^I+ST2, NSGAI^I+ST3, AND NSGAI^I+MIX ON SLOD ACROSS 20 INDEPENDENT RUNS WITH 75 000 FUNCTION EVALUATIONS

MOPs	NSGAI ^I	NSGAI ^I +AE	NSGAI ^I +ST1	NSGAI ^I +ST2	NSGAI ^I +ST3	NSGAI ^I +MIX
SLOD	2.7318±0.1880	3.0613±0.2454	2.9263±0.1494	3.1547±0.1084	3.1064±0.2917	3.5201±0.5610

VI. CONCLUSION

In this paper, to reuse knowledge from past searching experiences for accelerating the target optimization instance in a new parameter configuration, we proposed a novel multisource selective transfer optimization framework combining GA-style and EDA. Employing EDA probabilistic model learned by elite candidate solutions from the selection operation of GA-style EA, we present an optimization instance representation method named centroid distribution to measure the relatedness of different optimization instances during the evolution. The relatedness can assist the target instance to select the most suitable sources to learn and transfer. In the source selection step, we propose three strategies: 1) NSS; 2) WSS; and 3) Top-K. The first one aims at selecting the nearest source, but the latter two produce a combined source from multiple sources with different weights. To guide the use of the suitable selection strategy during the evolution, four using suggestions are summarized. To evaluate the effectiveness of the proposed framework, comprehensive studies on the multiobjective optimization problem benchmarks and a real-world application from the SLOD problem are conducted. The obtained results confirm the efficacy of our proposed framework for accelerating the searching process via the multisource selective transfer step when compared with the classical EA (NSGA-II) without selection among the sources. Our further study validates the effectiveness of strategy selection suggestions and the ability to solve negative transfer issues by our proposed MSS.

In the future, our research will focus as follows.

- 1) Improve the stableness of mixed and dynamic selection strategy.
- 2) Verify the effectiveness of MSSTO on multiple heterogeneous sources problems.
- 3) Evaluate the applicability of MSSTO in other types of optimization problems, e.g., multimodel optimization.

REFERENCES

- [1] W. Yao, X. Chen, Q. Ouyang, and M. V. Tooren, “A reliability-based multidisciplinary design optimization procedure based on combined probability and evidence theory,” *Struct. Multidiscipl. Optim.*, vol. 48, no. 2, pp. 339–354, 2013.
- [2] C. B. Do and A. Y. Ng, “Transfer learning for text classification,” in *Proc. 18th Adv. Neural Inf. Process. Syst.*, Vancouver, BC, Canada, Dec. 2005, pp. 299–306. [Online]. Available: <http://papers.nips.cc/paper/2843-transfer-learning-for-text-classification>
- [3] A. Quattoni, M. Collins, and T. Darrell, “Transfer learning for image classification with sparse prototype representations,” in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Anchorage, AK, USA, Jun. 2008, pp. 1–8. [Online]. Available: <https://doi.org/10.1109/CVPR.2008.4587637>
- [4] S. J. Pan and Q. Yang, “A survey on transfer learning,” *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010. [Online]. Available: <https://doi.org/10.1109/TKDE.2009.191>
- [5] Y. Yoshida, T. Hirao, T. Iwata, M. Nagata, and Y. Matsumoto, “Transfer learning for multiple-domain sentiment analysis—Identifying domain dependent/independent word polarity,” in *Proc. 25th AAAI Conf. Artif. Intell.*, San Francisco, CA, USA, Aug. 2011, pp. 1286–1291. [Online]. Available: <http://www.aaai.org/ocs/index.php/AAAI/AAAI11/paper/view/3597>
- [6] H. Hosseinzadeh and F. Razzaqi, “LMDT: A weakly-supervised large-margin-domain-transfer for handwritten digit recognition,” *Eng. Appl. Artif. Intell.*, vol. 52, pp. 119–125, Jun. 2016. [Online]. Available: <https://doi.org/10.1016/j.engappai.2016.02.014>
- [7] L. Feng, Y.-S. Ong, S. Jiang, and A. Gupta, “Autoencoding evolutionary search with learning across heterogeneous problems,” *IEEE Trans. Evol. Comput.*, vol. 21, no. 5, pp. 760–772, Oct. 2017. [Online]. Available: <https://doi.org/10.1109/TEVC.2017.2682274>
- [8] M. Jiang, Z. Huang, L. Qiu, W. Huang, and G. G. Yen, “Transfer learning-based dynamic multiobjective optimization algorithms,” *IEEE Trans. Evol. Comput.*, vol. 22, no. 4, pp. 501–514, Aug. 2018. [Online]. Available: <https://doi.org/10.1109/TEVC.2017.2771451>
- [9] M. Jiang, L. Qiu, Z. Huang, and G. G. Yen, “Dynamic multi-objective estimation of distribution algorithm based on domain adaptation and nonparametric estimation,” *Inf. Sci.*, vol. 435, pp. 203–223, Apr. 2018. [Online]. Available: <https://doi.org/10.1016/j.ins.2017.12.058>
- [10] R. Santana, A. Mendiburu, and J. A. Lozano, “Structural transfer using EDAs: An application to multi-marker tagging SNP selection,” in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Brisbane, QLD, Australia, Jun. 2012, pp. 1–8. [Online]. Available: <https://doi.org/10.1109/CEC.2012.6252963>
- [11] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, “Domain adaptation via transfer component analysis,” *IEEE Trans. Neural Netw.*, vol. 22, no. 2, pp. 199–210, Feb. 2011. [Online]. Available: <https://doi.org/10.1109/TNN.2010.2091281>
- [12] W. Dai, Q. Yang, G.-R. Xue, and Y. Yu, “Boosting for transfer learning,” in *Proc. 24th Int. Conf. Mach. Learn. (ICML)*, Corvallis, OR, USA, Jun. 2007, pp. 193–200. [Online]. Available: <https://doi.org/10.1145/1273496.1273521>
- [13] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
- [14] J. G. Rohra, B. Perumal, S. J. Narayanan, P. Thakur, and R. B. Bhatt, “User localization in an indoor environment using fuzzy hybrid of particle swarm optimization & gravitational search algorithm with neural networks,” in *Proc. 6th Int. Conf. Soft Comput. Problem Solving*, 2017, pp. 286–295.
- [15] T. Liu, Q. Yang, and D. Tao, “Understanding how feature structure transfers in transfer learning,” in *Proc. 26th Int. Joint Conf. Artif. Intell. (IJCAI)*, Melbourne, VIC, Australia, Aug. 2017, pp. 2365–2371. [Online]. Available: <https://doi.org/10.24963/ijcai.2017/329>
- [16] L. Feng *et al.*, “Evolutionary multitasking via explicit autoencoding,” *IEEE Trans. Cybern.*, vol. 49, no. 9, pp. 3457–3470, Sep. 2019.
- [17] R. Santana, C. Bielza, and P. Larrañaga, “Network measures for re-using problem information in EDAs,” Dept. Artif. Intell., Faculty Informat., Tech. Univ. Madrid, Madrid, Spain, Rep. UPM-FI/DIA/2010-3, 2010.
- [18] Y. Yu, S.-Y. Chen, Q. Da, and Z.-H. Zhou, “Reusable reinforcement learning via shallow trails,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 6, pp. 2204–2215, Jun. 2018. [Online]. Available: <https://doi.org/10.1109/TNNLS.2018.2803729>
- [19] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: NSGA-II,” *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.
- [20] A. Gupta, Y.-S. Ong, and L. Feng, “Insights on transfer optimization: Because experience is the best teacher,” *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 2, no. 1, pp. 51–64, Feb. 2018. [Online]. Available: <https://doi.org/10.1109/TETCI.2017.2769104>
- [21] M. Hauschild and M. Pelikan, “An introduction and survey of estimation of distribution algorithms,” *Swarm Evol. Comput.*, vol. 1, no. 3, pp. 111–128, 2011. [Online]. Available: <https://doi.org/10.1016/j.swevo.2011.08.003>

- [22] S. Baluja, "Population-based incremental learning. A method for integrating genetic search based function optimization and competitive learning," Tech. Rep., 1994. [Online]. Available: <https://dl.acm.org/citation.cfm?id=865123>
- [23] H. Mühlenbein, "The equation for response to selection and its use for prediction," *Evol. Comput.*, vol. 5, no. 3, pp. 303–346, 1997. [Online]. Available: <https://doi.org/10.1162/evo.1997.5.3.303>
- [24] G. R. Harik, F. G. Lobo, and D. E. Goldberg, "The compact genetic algorithm," *IEEE Trans. Evol. Comput.*, vol. 3, no. 4, pp. 287–297, Nov. 1999.
- [25] J. S. D. Bonet, C. L. Isbell, and P. Viola, "MIMIC: Finding optima by estimating probability densities," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, Denver, CO, USA, 1996, pp. 424–430.
- [26] M. Pelikan and H. Muehlenbein, *The Bivariate Marginal Distribution Algorithm*. London, U.K.: Springer, 1999.
- [27] M. Pelikan, D. E. Goldberg, and E. Cantú-Paz, "BOA: The Bayesian optimization algorithm," in *Proc. Conf. Genet. Evol. Comput.*, Orlando, FL, USA, 1999, pp. 525–532.
- [28] S. Shakya, J. McCall, and D. Brown, "Using a Markov network model in a univariate EDA: An empirical cost-benefit analysis," in *Proc. Genet. Evol. Comput. Conf. (GECCO)*, Washington, DC, USA, Jun. 2005, pp. 727–734.
- [29] Y. Gao, L. Peng, F. Li, M. Liu, and X. Hu, "EDA-based multi-objective optimization using preference order ranking and multivariate Gaussian copula," in *Proc. 10th Int. Symp. Neural Netw. Adv. Neural Netw. II (ISNN)*, Dalian, China, Jul. 2013, pp. 341–350. [Online]. Available: https://doi.org/10.1007/978-3-642-39068-5_42
- [30] H. Karshenas, R. Santana, C. Bielza, and P. Larrañaga, "Multiobjective estimation of distribution algorithm based on joint modeling of objectives and variables," *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 519–542, Aug. 2014. [Online]. Available: <https://doi.org/10.1109/TEVC.2013.2281524>
- [31] V. A. Shim, K. C. Tan, and C. Y. Cheong, "A hybrid estimation of distribution algorithm with decomposition for solving the multiobjective multiple traveling salesman problem," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 42, no. 5, pp. 682–691, Sep. 2012. [Online]. Available: <https://doi.org/10.1109/TSMCC.2012.2188285>
- [32] V. A. Shim, K. C. Tan, J. Y. Chia, and A. A. Mamun, "Multi-objective optimization with estimation of distribution algorithm in a noisy environment," *Evol. Comput.*, vol. 21, no. 1, pp. 149–177, Mar. 2013. [Online]. Available: https://doi.org/10.1162/EVCO_a_00066
- [33] J. Luo, Y. Qi, J. Xie, and X. Zhang, "A hybrid multi-objective PSO-EDA algorithm for reservoir flood control operation," *Appl. Soft Comput.*, vol. 34, pp. 526–538, Sep. 2015. [Online]. Available: <https://doi.org/10.1016/j.asoc.2015.05.036>
- [34] P. Lipinski, "Solving the firefighter problem with two elements using a multi-modal estimation of distribution algorithm," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Donostia-San Sebastián, Spain, Jun. 2017, pp. 2161–2168. [Online]. Available: <https://doi.org/10.1109/CEC.2017.7969566>
- [35] Q. Yang, W.-N. Chen, Y. Li, C. L. P. Chen, X.-M. Xu, and J. Zhang, "Multimodal estimation of distribution algorithms," *IEEE Trans. Cybern.*, vol. 47, no. 3, pp. 636–650, Mar. 2017. [Online]. Available: <https://doi.org/10.1109/TCYB.2016.2523000>
- [36] M. Pelikan, M. W. Hauschild, and P. L. Lanzi, "Transfer learning, soft distance-based bias, and the hierarchical BOA," in *Proc. 12th Int. Conf. Parallel Problem Solving Nat. I (PPSN XII)*, Taormina, Italy, Sep. 2012, pp. 173–183. [Online]. Available: https://doi.org/10.1007/978-3-642-32937-1_18
- [37] S. Jiang and S. Yang, "A steady-state and generational evolutionary algorithm for dynamic multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 21, no. 1, pp. 65–82, Feb. 2017.
- [38] S. Jiang and S. Yang, "Evolutionary dynamic multiobjective optimization: Benchmarks and algorithm comparisons," *IEEE Trans. Cybern.*, vol. 47, no. 1, pp. 198–211, Jan. 2017.
- [39] Q. Li, J. Zou, S. Yang, J. Zheng, and R. Gan, "A predictive strategy based on special points for evolutionary dynamic multi-objective optimization," *Soft Comput.*, vol. 23, no. 11, pp. 3723–3739, 2019.
- [40] G. Ruan, G. Yu, J. Zheng, J. Zou, and S. Yang, "The effect of diversity maintenance on prediction in dynamic multi-objective optimization," *Appl. Soft Comput.*, vol. 58, pp. 631–647, Sep. 2017. [Online]. Available: <https://doi.org/10.1016/j.asoc.2017.05.008>
- [41] J. Zou, Q. Li, S. Yang, H. Bai, and J. Zheng, "A prediction strategy based on center points and knee points for evolutionary dynamic multi-objective optimization," *Appl. Soft Comput.*, vol. 61, pp. 806–818, Dec. 2017. [Online]. Available: <https://doi.org/10.1016/j.asoc.2017.08.004>
- [42] A. Zhou, Y. Jin, Q. Zhang, B. Sendhoff, and E. P. K. Tsang, "Prediction-based population re-initialization for evolutionary dynamic multi-objective optimization," in *Proc. 4th Int. Conf. Evol. Multi Criterion Optim. (EMO)*, Matsushima, Japan, Mar. 2007, pp. 832–846. [Online]. Available: https://doi.org/10.1007/978-3-540-70928-2_62
- [43] J. Branke, "Memory enhanced evolutionary algorithms for changing optimization problems," in *Proc. Congr. Evol. Comput. (CEC)*, vol. 3, Washington, DC, USA, 1999, pp. 1875–1882.
- [44] C. K. Goh and K. C. Tan, "A competitive-cooperative coevolutionary paradigm for dynamic multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 13, no. 1, pp. 103–127, Feb. 2009.
- [45] I. Hatzakis and D. Wallace, "Dynamic multi-objective optimization with evolutionary algorithms: A forward-looking approach," in *Proc. 8th Annu. Conf. Genet. Evol. Comput.*, Seattle, WA, USA, 2006, pp. 1201–1208.
- [46] Z. Zhang and S. Qian, "Artificial immune system in dynamic environments solving time-varying non-linear constrained multi-objective problems," *Soft Comput.*, vol. 15, no. 7, pp. 1333–1349, 2011.
- [47] L. T. Bui, Z. Michalewicz, E. Parkinson, and M. B. Abello, "Adaptation in dynamic environments: A case study in mission planning," *IEEE Trans. Evol. Comput.*, vol. 16, no. 2, pp. 190–209, Apr. 2012.
- [48] R. Shang, L. Jiao, Y. Ren, L. Li, and L. Wang, "Quantum immune clonal coevolutionary algorithm for dynamic multiobjective optimization," *Soft Comput.*, vol. 18, no. 4, pp. 743–756, 2014.
- [49] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, "Reading digits in natural images with unsupervised feature learning," 2011. [Online]. Available: <https://ai.google/research/pubs/pub37648>
- [50] I. Olkin and F. Pukelsheim, "The distance between two random vectors with given dispersion matrices," *Linear Algebra Appl.*, vol. 48, no. 82, pp. 257–263, Dec. 1982.
- [51] C. M. Bishop, *Pattern Recognition and Machine Learning* (Information Science and Statistics). New York, NY, USA: Springer-Verlag, 2006, p. 55.
- [52] S. Semeniuta, A. Severyn, and E. Barth, "A hybrid convolutional variational autoencoder for text generation," in *Proc. Conf. Empirical Methods Nat. Lang. Process. (EMNLP)*, Copenhagen, Denmark, Sep. 2017, pp. 627–637. [Online]. Available: <https://aclanthology.info/papers/D17-1066/d17-1066>
- [53] G. Zhou, Z. Zhu, T. He, and X. T. Hu, "Cross-lingual sentiment classification with stacked autoencoders," *Knowl. Inf. Syst.*, vol. 47, no. 1, pp. 27–44, 2016. [Online]. Available: <https://doi.org/10.1007/s10115-015-0849-0>
- [54] E. Kodirov, T. Xiang, and S. Gong, "Semantic autoencoder for zero-shot learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 4447–4456. [Online]. Available: <https://doi.org/10.1109/CVPR.2017.473>
- [55] X. Hou, L. Shen, K. Sun, and G. Qiu, "Deep feature consistent variational autoencoder," in *Proc. Appl. Comput. Vis.*, Santa Rosa, CA, USA, 2017, pp. 1133–1141.
- [56] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: Empirical results," *Evol. Comput.*, vol. 8, no. 2, pp. 173–195, 2000.
- [57] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, *Scalable Test Problems for Evolutionary Multiobjective Optimization*. London, U.K.: Springer, 2001.
- [58] Y. Tian, R. Cheng, X. Zhang, and Y. Jin, "PlatEMO: A MATLAB platform for evolutionary multi-objective optimization [educational forum]," *IEEE Comp. Int. Mag.*, vol. 12, no. 4, pp. 73–87, Nov. 2017. [Online]. Available: <https://doi.org/10.1109/MCI.2017.2742868>
- [59] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach," *IEEE Trans. Evol. Comput.*, vol. 3, no. 4, pp. 257–271, Nov. 1999.
- [60] J. Bader and E. Zitzler, "Hype: An algorithm for fast hypervolume-based many-objective optimization," *Evol. Comput.*, vol. 19, no. 1, pp. 45–76, 2011.
- [61] B. Zhang, H.-F. Teng, and Y.-J. Shi, "Layout optimization of satellite module using soft computing techniques," *Appl. Soft Comput.*, vol. 8, no. 1, pp. 507–521, 2008.
- [62] X. Chen, W. Yao, Y. Zhao, X. Chen, and X. Zheng, "A practical satellite layout optimization design approach based on enhanced finite-circle method," *Struct. Multidiscipl. Optim.*, vol. 58, no. 6, pp. 2635–2653, Dec. 2018. [Online]. Available: <https://doi.org/10.1007/s00158-018-2042-z>
- [63] H.-F. Teng, Y. Chen, W. Zeng, Y.-J. Shi, and Q.-H. Hu, "A dual-system variable-grain cooperative coevolutionary algorithm: Satellite-module layout design," *IEEE Trans. Evol. Comput.*, vol. 14, no. 3, pp. 438–455, Jun. 2010.

- [64] D. W. Hengeveld, J. E. Braun, E. A. Groll, and A. D. Williams, "Optimal placement of electronic components to minimize heat flux nonuniformities," *J. Spacecraft Rockets*, vol. 48, no. 4, pp. 556–563, 2015.



Jun Zhang received the B.Eng. degree in aircraft systems and engineering and the M.Sc. degree in aeronautical and astronautical science and technology from the National University of Defense Technology, Changsha, China, in 2011 and 2016, respectively. He is currently pursuing the Ph.D. degree in computer science with the Department of Computer Science and Technology, Tsinghua University, Beijing, China.

He is currently an Assistant Professor with the National Innovation Institute of Defense Technology, Chinese Academy of Military Science, Beijing. His current research interests include transfer learning, reinforcement learning, Bayesian machine learning, and multiobjective evolutionary algorithms.



Weien Zhou received the B.Sc. degree in mathematics from Nanjing University, Nanjing, China, in 2012, and the Ph.D. degree in computational mathematics from the National University of Defense Technology, Changsha, China, in 2017.

He is currently an Assistant Professor with the National Innovation Institute of Defense Technology, Chinese Academy of Military Science, Beijing, China. His current research interests include numerical methods for stochastic systems, uncertainty quantification, and large-scale optimization.



Xianqi Chen received the B.Eng. degree in aerospace engineering from the National University of Defense Technology, Changsha, China, in 2016, where he is currently pursuing the master's degree with the College of Aerospace Science and Engineering.

His current research interests include satellite layout optimization design and evolutionary algorithms in aerospace engineering.



Wen Yao received the M.Sc. and Ph.D. degrees in aerospace engineering from the National University of Defense Technology, Changsha, China, in 2007 and 2011, respectively.

She is currently a Professor with the National Innovation Institute of Defense Technology, Chinese Academy of Military Science, Beijing, China. Her current research interests include spacecraft systems engineering, multidisciplinary design optimization, uncertainty-based optimization, and data-driven surrogate modeling and evolutionary optimization.



Lu Cao received the M.Sc. and Ph.D. degrees in aerospace engineering from the National University of Defense Technology, Changsha, China, in 2010 and 2014, respectively.

He is currently an Associate Professor with the National Innovation Institute of Defense Technology, Chinese Academy of Military Science, Beijing, China. His current research interests include spacecraft systems engineering, controlling system, and Kalman filtering.