How to Exploit Optimization Experience? Revisiting Evolutionary Sequential Transfer Optimization: Part A - Benchmark Problems

xiaoming xue [1], cuie yang [2], Lianghuan Feng [2], kai zhang [2], linqi song [2], and Kay Chen Tan [2]

[1]City University of Hong Kong
[2]Affiliation not available

October 30, 2023

Abstract

Evolutionary sequential transfer optimization (ESTO), which attempts to enhance the evolutionary search of a target task using the knowledge captured from several previously-solved source tasks, has been receiving increasing research attention in recent years. Despite the tremendous approaches developed, it is worth noting that existing benchmark problems for ESTO are not well designed, as they are often simply extended from other benchmarks in which the relationships between the source and target tasks are not well analyzed. Consequently, the comparisons conducted on these problems are not systematic and can only provide numerical results without a deeper analysis of how an ESTO algorithm performs on problems with different properties. Taking this clue, this two-part paper revisits a large body of solution-based ESTO algorithms on a group of newly developed test problems, to help researchers and practitioners gain a deeper understanding of how to better exploit optimization experience towards enhanced optimization performance. Part A of the series designs a problem generator based on several newly defined concepts to generate benchmark problems with diverse properties, which are competitive in resembling real-world problems. Part B of the series empirically revisits various algorithms by answering five key research questions related to knowledge transfer. The results demonstrated that the performance of many ESTO algorithms is highly problem-dependent, which suggest the necessity of more research efforts on transferability measurement and enhancement in ESTO algorithm design. The source code of the benchmark suite developed in part A is available at https://github.com/XmingHsueh/Revisiting-S-ESTOs-PartA.

How to Exploit Optimization Experience? Revisiting Evolutionary Sequential Transfer Optimization: Part A - Benchmark Problems

Xiaoming Xue, Cuie Yang, *Member, IEEE*, Liang Feng, Kai Zhang, *Member, IEEE*, Linqi Song, *Member, IEEE*, and Kay Chen Tan, *Fellow, IEEE*

***Abstract*—Evolutionary sequential transfer optimization (ESTO), which attempts to enhance the evolutionary search of a target task using the knowledge captured from several previously-solved source tasks, has been receiving increasing research attention in recent years. Despite the tremendous approaches developed, it is worth noting that existing benchmark problems for ESTO are not well designed, as they are often simply extended from other benchmarks in which the relationships between the source and target tasks are not well analyzed. Consequently, the comparisons conducted on these problems are not systematic and can only provide numerical results without a deeper analysis of how an ESTO algorithm performs on problems with different properties. Taking this clue, this two-part paper revisits a large body of solution-based ESTO algorithms on a group of newly developed test problems, to help researchers and practitioners gain a deeper understanding of how to better exploit optimization experience towards enhanced optimization performance. Part A of the series designs a problem generator based on several newly defined concepts to generate benchmark problems with diverse properties, which are competitive in resembling real-world problems. Part B of the series empirically revisits various algorithms by answering five key research questions related to knowledge transfer. The results demonstrated that the performance of many ESTO algorithms is highly problem-dependent, which suggest the necessity of more research efforts on transferability measurement and enhancement in ESTO algorithm design. The source code of the benchmark suite developed in part A is available at https://github.com/XminghSueh/Revisiting-S-ESTOs-PartA.**

***Index Terms*—optimization experience, sequential transfer optimization, test problems, benchmark suite.**

I. Introduction

Knowledge transfer-enhanced optimization, known as *transfer optimization* [1], has received increasing research interests in the past years. Similar to transfer learning [2, 3], transfer optimization aims to improve the performance of an optimizer on target task(s) using knowledge extracted from related tasks [4–6]. Since evolutionary algorithms (EAs) are easy to be implemented and do not require optimization problems to have the nice mathematical properties needed by mathematical programming algorithms, a variety of transfer optimization techniques using EAs as optimizers have been developed and recognized as *evolutionary transfer optimization* (ETO) in the literature [7]. According to the conceptual realizations in [1], ETO can be classified into three categories: evolutionary multitasking [8–10], evolutionary multiform optimization [11–13], and evolutionary sequential transfer optimization (ESTO) [5, 14–16]. Specifically, ESTO refers to an evolutionary search paradigm that leverages the searching experience captured from previously-solved optimization tasks to speed up the search when solving an unseen target task [16].

In the literature, a variety of ESTO algorithms have been proposed. In terms of "what to transfer", existing ESTO algorithms can be categorized into three types: algorithm-based methods [17], model-based methods [18], and solution-based methods [5]. Fig. 1 presents a timeline that illustrates the development of each type of ESTO. In each row, the height of each shaded area denotes the number of publications of each type over years. In algorithm-based ESTO, algorithm configuration [17, 19, 20], algorithm selection [21, 22], and algorithm portfolio [23–25] are three representative methods [26]. Algorithm configuration transfers the set of optimal parameters obtained on a group of source tasks to the target task [20]. Algorithm selection reuses the algorithm with the best performance among a group of candidate algorithms on the target task [27]. Algorithm portfolio transfers candidate algorithm(s) and its parameters simultaneously [23]. Moreover, knowledge can also be transferred in the form of model information in ESTO, i.e., model-based methods. Model biasing [28–30] and model aggregation [31, 32] are two well-known methods in this category. Model biasing captures the model information from source tasks and uses them to bias the initial search distribution on a target task [28], while model aggregation constructs a mixture distribution by aggregating the optimized source distributions into the target one [33]. Furthermore, in *solution-based ESTO* (S-ESTO), the knowledge is transferred in the form of solution(s) across tasks. Due to its

Xiaoming Xue and Linqi Song are with the Department of Computer Science, City University of Hong Kong, Hong Kong SAR, China and also with the City University of Hong Kong Shenzhen Research Institute, Shenzhen 518057, China (e-mail: xminghsueh@gmail.com; linqi.song@cityu.edu.hk).

Cuie Yang is with the State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University, Shenyang 110819, China (e-mail: cuieyang@outlook.com).

Liang Feng is with the College of Computer Science, Chongqing University, Chongqing 400044, China (e-mail: liangf@cqu.edu.cn).

K. Zhang is with the Civil Engineering School, Qingdao University of Technology, Qingdao 266520, China (e-mail: zhangkai@qut.edu.cn).

Kay Chen Tan is with the Department of Computing, Hong Kong Polytechnic University, Hong Kong SAR, China (e-mail: kctan@polyu.edu.hk).

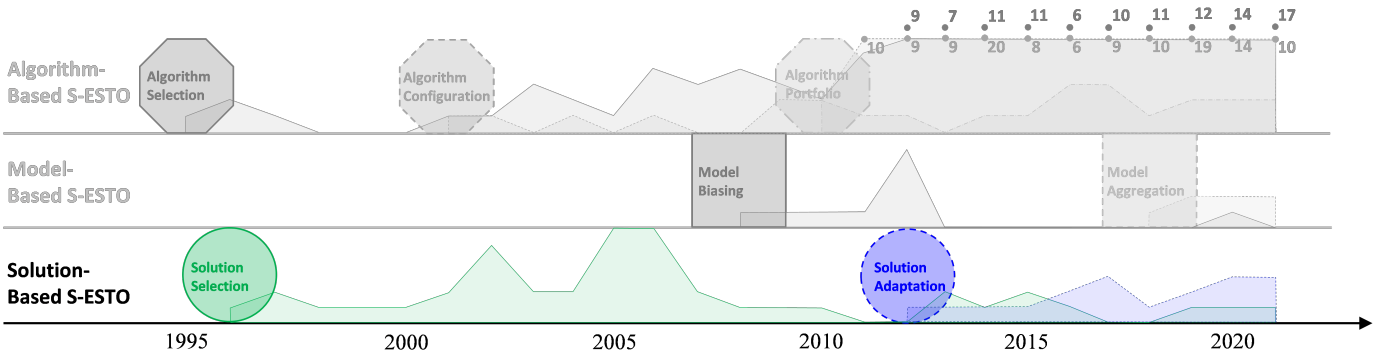


Fig. 1: A timeline graph of showing the development of various ESTO approaches. The height of each shaded area denotes the number of publications related with theoretical research or practical applications.

ease of implementability and optimizer-independent property[1], S-ESTO has obtained increasing research attention in the past decades, as shown in Fig. 1. However, exiting studies mainly focus on designing knowledge transfer methods, without rigorous analyses on the performance obtained by knowledge transfer and how different algorithms perform on problems with varying properties. This paper thus presents an attempt to fill this gap.

First of all, in evolutionary computation, using benchmarks is a common way of investigating various algorithms, which can provide valuable insights of the performance of different methods and arouse important concerns in designing better algorithms. However, existing test problems in S-ESTO are simply extended from other benchmarks [14, 34], in which the relationships between source and target tasks are not analyzed, limiting their ability to represent real-world problems. Besides, there are no common test problems for S-ESTO study, and existing S-ESTO methods are evaluated on different problems, which could result in unfair comparisons, since an algorithm that performs well on one problem set often shows unsatisfactory performance on another group of problems [35]. Thus, it is desirable to design a common benchmark suite containing problems with diverse properties to promote the competition between different algorithms and, as a consequence, boost the development of S-ESTO.

Next, algorithm design in S-ESTO mainly focuses on *solution selection* and *solution adaptation* [34, 36]. In solution selection, a similarity metric is often employed to identify transferable source solution(s) [5, 37–39]. However, the selected solutions are hard to ensure their usefulness for the target task due to the intrinsic uncertainty of similarity metric in measuring transferability. Towards positive knowledge transfer from source to target task, solution adaptation is proposed to adapt the source solutions(s) for a higher transferability [14, 40, 41]. Moreover, solution selection and solution adaptation are also integrated to curb the negative transfer in multi-source problems [34]. However, it is still unclear what factors contribute to the efficacies of solution selection and solution adaptation. Therefore, it is worth conducting an in-depth investigation of various S-ESTO algorithms on a common benchmark suite, to help researchers and practitioners gain a deeper understanding of how to exploit optimization experience more effectively.

In the light of the above, in this two-part paper, we intend to design a common benchmark suite for S-ESTO, and experimentally revisit and analyze existing S-ESTO algorithms on the designed test problems. Part A of the series covers important concepts, design guidelines, and key ingredients for generating S-ESTO benchmarks. Firstly, we introduce the basic concepts to characterize sequential transfer optimization problems. Based on the newly defined concepts, we discuss an important design feature named similarity distribution, which describes the similarity relationship between tasks. Then, a set of design guidelines and a problem generator for designing benchmarks are proposed. Lastly, a benchmark suite containing 12 individual problems is built. To provide deeper insights on the design of S-ESTO algorithms, part B employs the new test problems developed in part A to empirically revisit a wide variety of knowledge transfer techniques by addressing the following five research questions (RQs):

1) *RQ1:* How do existing similarity metrics perform in solution selection for S-ESTO?
2) *RQ2:* Which factor is essential to the effectiveness of similarity metrics in S-ESTO?
3) *RQ3:* How do existing adaptation techniques perform in solution adaptation for S-ESTO?
4) *RQ4:* What contributes to the effectiveness of solution adaptation models in S-ESTO?
5) *RQ5:* How to integrate solution selection and solution adaptation for S-ESTO?

Part A is organized as follows. Section II presents the problem definitions and discusses key transfer strategies in S-ESTO. Then, we introduce several important concepts to characterize sequential transfer optimization problems in Section III. After that, Section IV summarizes the crucial design features and general guidelines for designing S-ESTO benchmarks. In Section V, we build a S-ESTO benchmark suite containing 12 individual problems based on the discussions in Section IV. Lastly, Section VI concludes part A of this series.

[1]As knowledge is transferred in the form of solution(s), S-ESTO does not rely on the configuration of EA solvers.

II. Preliminaries

Firstly, the definitions of ESTO and S-ESTO are presented. Thereafter, we review a number of representative S-ESTO algorithms and the key transfer strategies involved in these methods. Lastly, the deficiencies of existing test problems are analyzed.

A. Definitions of ESTO and S-ESTO

With a number of previously-solved source tasks, *sequential transfer optimization* (STO) aims to improve the solving efficiency of a target task with the aid of knowledge captured from the source tasks, which can be formally portrayed as [1]:

$$\min_{\mathbf{x}\in\Omega} \left[f^t(\mathbf{x}) \,|\mathcal{M}, \mathcal{P}^t \right] \tag{1}$$

where $\mathbf{x}$ represents the decision vector, Ω is the decision space, f^t denotes the scalar objective function of the target task, $\mathcal{M}$ represents the database containing available information drawn from the k source tasks, $\mathcal{P}^t = \{P^{tj}, F^{tj}, j = 1, ..., t\}$ denotes the evaluated solutions of the target task from the first generation to generation t (P^{tj} is the population data at the jth generation and F^{tj} represents the fitness values of the solutions at the jth generation). STO using EAs as its optimizer is termed *evolutionary sequential transfer optimization* (ESTO).

Given a sequential transfer optimization problem[2] (STOP), its knowledge database often contains evaluated solutions on source tasks and the collected partial information of the source and target tasks. This is particularly the case for many combinatorial optimization problems [42]. For example, partial information such as nodes' locations in a traveling salesman problem is readily accessible [43]. Leveraging such information for better solving efficiency is very common in EA and STO [41, 44]. In this study, we will not cover ESTO methods using problem information since these studies are usually problem dependent. Instead, we focus on black-box ESTO in which the knowledge is extracted from searched solutions. This type of ESTO is expected to benefit a wider range of practical applications [5]. Particularly, ESTO that transfers solution(s) is recognized as *solution-based evolutionary sequential transfer optimization* (S-ESTO). The available source data in S-ESTO is simply evaluated solutions of the k source tasks:

$$\mathcal{M} = \{\mathcal{P}^{s1}, \mathcal{P}^{s2}, ..., \mathcal{P}^{sk}\} \tag{2}$$

where $\mathcal{P}^{si} = \{P^{sij}, F^{sij}, j = 1, ..., G\}$ is the population data of the ith source task, P^{sij} denotes the evaluated solutions of the ith source task at the jth generation and F^{sij} represents the fitness vector of the ith source task at the jth generation.

B. S-ESTO Algorithms

The general framework of existing S-ESTO algorithms is shown in Fig. 2. The knowledge transfer module is often independent of evolutionary optimizers, making an S-ESTO algorithm can be easily modularized. Without loss of generality, any of other population-based optimizers (e.g., particle swarm optimization [45], PSO) can be used to replace the conventional EAs in Fig. 2. In the knowledge transfer module, task information in the form of evaluated solutions is fed into a knowledge extraction phase for generating transferable solutions. For tasks with different decision spaces, one can transform them into a common space and extract the transferable solution(s) therein. Then, the transferable solution(s) in the common space is transformed to the target task for further evaluation. For convenience, all the subsequent discussions about solution are assumed in the context of common space. In the selection phase, the transferred solutions with competitive or superior fitnesses will be automatically reserved to the next generation, while those with poor fitness values will be discarded. In the past years, many S-ESTO approaches have been proposed in the literature, which focus either on the selection or adaptation of solutions for transfer across tasks. Generally, existing S-ESTO approaches can be classified into the following three categories:

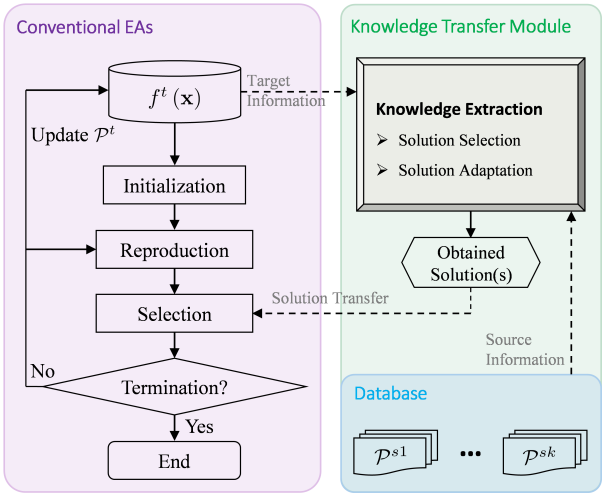


Fig. 2: The general framework of S-ESTO algorithms.

1) Solution selection-based S-ESTO [5, 46]: To identify the most transferable solution(s), solution selection employs a similarity metric to select the most similar source task to the target task and then extracts solution(s) for transfer.
2) Solution adaptation-based S-ESTO [14, 47]: To overcome the task heterogeneity, solution adaptation adapts transferable solution(s) for a higher transferability, where the source task that provides the solution(s) is often randomly selected.
3) S-ESTO that integrates selection and adaptation [32, 34]: Towards more effective knowledge transfer in a multi-source scenario, solution selection and solution adaptation are integrated in S-ESTO. Firstly, the most similar source task to the target task is identified by solution selection to provide transferable solution(s). Then, the transferable solution(s) is adapted and transferred to the target task via solution adaptation.

More detailed technical reviews of the above three types of S-ESTO approaches are provided in part B of this series.

[2]We distinguish "problem" from "task" in this work. A sequential transfer optimization problem contains source and target optimization tasks.

C. Existing Test Problems

To the best of our knowledge, there is no common benchmark suite for ESTO in the literature. Existing test problems considered in the literature are often different in each study, which are either extended from other benchmark problems or generated from particular practical problems with limited variations.

1) Problems Extended from Other Benchmarks: In [14, 47], the test problems are from two commonly used benchmarks, including 5 MOPs in ZDT family [48] and 7 MOPs in DTLZ family [49]. It is important to note that these problems are designed for the research of multi-objective optimization rather than S-ESTO. The optima of all the problems are set to be zero vectors for convenience, as they are solved independently. However, due to the closer optima of source and target tasks, one should scrutinize these problems before applying them to evaluate S-ESTO algorithms. On these problems, the direct reuse of source solutions often leads to a significant speedup of the search in the target task. However, the closer optima of source and target tasks are very rare in real-world problems. Moreover, in [16], the test problems are extended from the benchmark functions in single-objective and multi-objective optimization, in which the optima of source and target tasks are randomly configured. However, the relationship between the randomly configured optima can only represent the relationship of a particular type of STOPs. As shown in Part B, an S-ESTO algorithm that shows superior performance on the problems in [16] does not work well on many other STOPs.

2) Problems Generated from Practical Instances: In [5], the empirical studies are conducted on three practical applications: combinational circuit design, strike force asset allocation, and job shop scheduling. To ensure a certain degree of solution similarity, they generate source tasks based on the parameters of the target task. Similarly, in [34], source tasks are synthesized via sampling the parameters in objectives and constraints of the target task. As a result, the optima of the generated source tasks are close to the target one. However, an underlying assumption in Eq. (1) is that the source tasks are independently accumulated before the target task. Thus, the prespecified similarities in [5, 34] are unrealistic in practice.

In summary, the relationships between the optima of source and target tasks in existing test problems are configured in an unsystematic manner. The limited patterns of such relationships make the test problems insufficient for evaluating various S-ESTO algorithms. In Section III, we will introduce a number of basic concepts to characterize STOPs, which can help us figure out an important feature ignored by previous studies, i.e., similarity distribution.

III. Basic Concepts

To characterize STOPs, we first define a number of basic concepts. The newly defined concepts and their relations with several well-established notions in ESTO are illustrated in Fig. 3. Particularly, in view of multiple individual tasks (i.e., source and target tasks) in an STOP, we define a concept named *task family* to refer to a task set from which the source and target tasks are sampled. Next, since S-ESTO transfers knowledge in the form of solution(s), it is necessary to analyze the relationship between the optimal solutions of source and target tasks. A novel concept named *task-optimum mapping* is defined to connect a task to its optimum. Thus, given a task family, there will be a set of optima corresponding to the elementary tasks, which is termed the image of task-optimum mapping. Then, we define *optimum coverage* to describe the relative size of the image over the decision space. Finally, by measuring the similarity between two tasks based on the distance of their optimal solutions, we introduce an important problem feature of STOPs named *similarity distribution*, which describes the similarity relationship between the k source tasks and the target task in an STOP. In part B, the experimental results reveal that the performance of an S-ESTO algorithm is very susceptible to similarity distribution. To help readers digest all the newly defined concepts, we present an illustrative example at the end of this section.

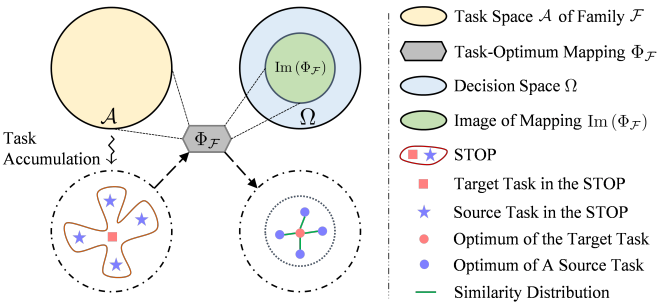


Fig. 3: An illustration of the newly defined concepts.

A. Task Family

Firstly, we introduce the concept of task family referring to a task set from which the tasks in an STOP are sampled, which is given by:

Definition 1. *A task family $\mathcal{F}(\mathbf{x};\mathcal{A})$ consists of a set of elementary optimization tasks $f(\mathbf{x};\mathbf{a})$ with the decision variable $\mathbf{x}$ in a decision space Ω, which is given by:*

$$\mathcal{F}(\mathbf{x};\mathcal{A}) = \{f(\mathbf{x};\mathbf{a}) \,|\mathbf{a} \in \mathcal{A}\}, \quad \mathbf{x} \in \Omega \tag{3}$$

where $\mathcal{A}$ is a task space that contains all the realizations of an elementary task; $\mathbf{a}$ denotes a feature vector that characterizes a particular optimization task in $\mathcal{F}$.

The concept of task family can be exemplified by different optimization problems. For instance, minimization tasks of n-dimensional quadratic functions with distinct coefficients can be regarded as a task family. A realization of the coefficients deterministically characterizes a specific quadratic optimization task. Moreover, in combinational optimization, traveling salesman optimization tasks with n cities can be treated as a task family. The coordinate information deterministically characterizes a specific task. Notably, the task family here is a fairly general concept whose elementary tasks depend on the application of interest. Given a task family $\mathcal{F}$, we assume that an elementary task $f \in \mathcal{F}$ is a random variable with a latent distribution, which is given by:

$$f(\mathbf{x};\mathbf{a}) \sim f(\mathbf{x};p_{\mathcal{A}}) \tag{4}$$

where $p_{\mathcal{A}}$ denotes the distribution of the parametric features.

According to the definition of task family, knowledge transfers in S-ESTO can be divided into two categories: *intra-family transfer* and *inter-family transfer*. Specifically, in the intra-family transfer, source and target tasks are drawn from the same family, thus sharing similar landscape modalities. By contrast, source and target tasks in the inter-family transfer are from different families.

B. Task-Optimum Mapping

Given a task family $\mathcal{F}(\mathbf{x};\mathcal{A})$, $f(\mathbf{x};\mathbf{a}) \in \mathcal{F}(\mathbf{x};\mathcal{A})$ is a regular optimization task [50]. Without loss of generality, the elementary tasks are considered to be single-objective minimization optimization tasks[3], which can be formulated as follows:

$$\mathbf{x}^* = \min_{\mathbf{x}\in\Omega} f(\mathbf{x};\mathbf{a}) \tag{5}$$

where $\mathbf{x}^*$ denotes the optimal solution. Generally, for a task with given fixed features, its optimum is unique. Thus, we define task-optimum mapping to describe the relation between the features and optimum of a task.

Definition 2. *Given a task family $\mathcal{F}(\mathbf{x};\mathcal{A})$, its task-optimum mapping $\Phi_{\mathcal{F}} : \mathcal{A} \to \Omega$ models the relationship between features and optimum:*

$$\Phi_{\mathcal{F}}(\mathbf{a}) = \mathbf{x}^*, \ \ \mathbf{x}^* \subseteq \Omega \tag{6}$$

where $\Phi_{\mathcal{F}}(\cdot)$ denotes the task-optimum mapping of $\mathcal{F}(\mathbf{x};\mathcal{A})$.

The task-optimum mapping defined here is a fairly general concept that can refer to the optimization process of any optimization task of interest. For tasks with an explicit relation between features and optimum, an analytical solution of task-optimum mapping is often available. For example, the optimal solution of a convex quadratic optimization task can be analytically derived based on its coefficients [51]. More often, features and optimum of a task are implicitly related. In this case, an optimizer is usually employed to search for the optimum by iteratively evaluating the objective function. Importantly, the concept of task-optimum mapping still holds despite the implicity, allowing us to learn optimization experience. In particular, given a set of previously-solved tasks $\{f(\mathbf{x};\mathbf{a}_s^i), i = 1,...,k\}$ from a task family $\mathcal{F}(\mathbf{x};\mathcal{A})$, their optima are represented as $\{\mathbf{x}_{si}^*, i = 1,...,k\}$. The task-optimum mapping of this family can be estimated via a supervised learning algorithm [52–55]:

$$\widetilde{\Phi}_{\mathcal{F}} = \min_{\Phi\in\Theta} \sum_{i=1}^{k} l\left(\Phi\left(\mathbf{a}_s^i\right), \mathbf{x}_{si}^*\right) \tag{7}$$

where Θ represents a mapping space with candidate mappings, $l(\cdot,\cdot)$ denotes a metric function for measuring the discrepancy between the predicted optimum and the true optimum. Now, given an unsolved target task $f(\mathbf{x};\mathbf{a}_t) \in \mathcal{F}$, its optimum can be predicted as follows:

$$\widetilde{\mathbf{x}}_t^* = \widetilde{\Phi}_{\mathcal{F}}(\mathbf{a}_t) \tag{8}$$

Furthermore, a task-optimum mapping is uniformly continuous if it is continuous on the task space (i.e., $\mathcal{A}$). The uniform continuity states that $\forall x, y \in \mathcal{A}$, $\exists\delta > 0$, $\forall\epsilon > 0$:

$$|x - y| < \delta \Rightarrow |\Phi(x) - \Phi(y)| < \epsilon \tag{9}$$

This property suggests that two similar tasks in the feature space possess closer optimal solutions, which forms the foundation of many STO algorithms using problem information (i.e., $\mathbf{a}_t$) for similarity measurement. Several representative methods can be found in [38, 56–59]. Given a task family, the image of its task-optimum mapping is the set of all optimal solutions in the decision space [60], as illustrated in Fig. 3. The following subsection defines a concept named optimum coverage used for describing the relative size of this image.

C. Optimum Coverage

Given a task family $\mathcal{F}(\mathbf{x};\mathcal{A})$, its task-optimum mapping $\Phi_{\mathcal{F}}$ often has the following two properties:

- *Non-injective*: $\exists \mathbf{a}_i, \mathbf{a}_j \in \mathcal{A} : \Phi(\mathbf{a}_i) = \Phi(\mathbf{a}_j) \nRightarrow \mathbf{a}_i = \mathbf{a}_j$.
- *Non-surjective*: $\Phi(\mathcal{A}) \neq \Omega$.

The first property implies that different tasks from the same family may share a common optimal solution. For instance, a minor change of edge information in a short-path optimization task may not result in the alteration of the shortest path. The second property means that the image of task-optimum mapping, i.e., $\mathrm{Im}(\Phi_{\mathcal{F}})$, is distributed over a subregion of the decision space. Fig. 3 shows the relationship between the task space $\mathcal{A}$ and the decision space Ω connected by the task-optimum mapping $\Phi_{\mathcal{F}}$. The image of $\Phi_{\mathcal{F}}$ is the set of all optimal solutions corresponding to the tasks in $\mathcal{A}$. According to Eq. (4) and the definition of task-optimum mapping, we have the distribution of optimal solutions as follows:

$$\Phi_{\mathcal{F}}(\mathbf{a} \sim p_{\mathcal{A}}) = \mathbf{x}^* \sim q_{\mathcal{A}}(\mathbf{x}^*) \tag{10}$$

where $q_{\mathcal{A}}$ represents the optimum distribution of the elementary tasks in $\mathcal{F}$. Here, the event space of optimum $\mathbf{x}^*$ is equivalent to the image of $\Phi_{\mathcal{F}}$. Next, we define the concept of optimum coverage to describe the relative size of such image over the decision space.

Definition 3. *Given a set of task families $\Theta = \{\mathcal{F}_1,...,\mathcal{F}_m\}$ from which the source and target tasks of an STOP are obtained, where m is the number of task families, optimum coverage is defined as the ratio of the image $\mathrm{Im}(\Phi_\Theta)$ to the common space Ω_c:*

$$\gamma = \frac{\int_{x\in\mathrm{Im}(\Phi_\Theta)} \mathrm{d}x}{\int_{x\in\Omega_c} \mathrm{d}x} \tag{11}$$

where γ denotes the optimum coverage. $\mathrm{Im}(\Phi_\Theta)$ is $\mathrm{Im}(\Phi_{\mathcal{F}})$ in the common space for a single family, while it equals to $\mathrm{Im}(\Phi_{\mathcal{F}_1}) \cup \mathrm{Im}(\Phi_{\mathcal{F}_2}) \cup ... \cup \mathrm{Im}(\Phi_{\mathcal{F}_m})$ for multiple families. Apparently, m is equal to 1 in intra-family transfer while it is greater than 1 in inter-family transfer.

[3]The generalization to multi- and many-objective tasks is trivial. A maximization task can be converted to the minimization task via multiplying the objective function by -1.

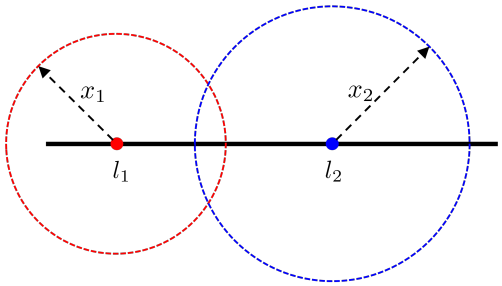


Fig. 4: A schematic diagram of the interval coverage task.

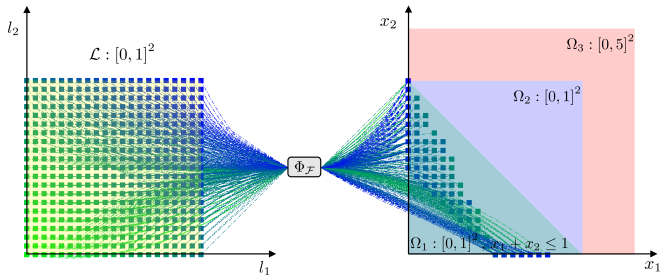


Fig. 5: A great number of interval coverage tasks in $\mathcal{F}(\mathbf{x};\mathcal{L})$ and the locations of their optima in Ω_1, Ω_2 and Ω_3.

In particular, $\gamma \approx 0$ if all the tasks in a family set share a common optimum, $\gamma = 1$ when the optimal solutions spread over the whole search space. It is noted that most problems fall between these two extreme cases. To some degree, the optimum coverage of a task family can reflect one's prior knowledge about this family. Generally, for a task family with little prior knowledge, one often adopts a large decision space due to his weak prior beliefs about the optimal solutions. In this case, the optimum coverage is small. By contrast, one always adopts a small decision space if some priors about the optima are available, resulting in a large optimum coverage. Therefore, optimum coverage is a highly problem-dependent property of task family. When using S-ESTO algorithms, one can readily obtain the target optimum by reusing the source optima if $\gamma \approx 0$. However, if γ is large, it is challenging to acquire the target optimum based on a limited number of diversely distributed source optima. Next, to characterize the relationship between the optimal solutions of source and target tasks more clearly, we define a concept named similarity distribution in what follows.

D. Similarity Distribution

When the source and target tasks of an STOP are obtained from a family set $\Theta = \{\mathcal{F}_1, ..., \mathcal{F}_m\}$, the accumulation of the tasks can be seen as an increasing number of independent samples drawn from the joint task distribution of the m task families. Suppose the source and target tasks are obtained as $\{f(\mathbf{x};\mathbf{a}_s^i), i = 1, ..., k\}$ and $f(\mathbf{x};\mathbf{a}_t)$, we have their optima as $\{\mathbf{x}_{si}^*, i = 1, ..., k\}$ and $\mathbf{x}_t^*$, which are denoted as $\{\mathbf{o}_{si}, i = 1, ..., k\}$ and $\mathbf{o}_t$ in the common space. Since S-ESTO transfers optimized solution(s), the similarity between the optimal solutions of two tasks largely reflects their transferability[4]. Therefore, we measure the similarity between the ith source task and the target task using the Chebyshev distance[5] between their optimal solutions, which is given by,

$$S_i = 1 - D_{Cheb}(\mathbf{o}_t, \mathbf{o}_{si}) = 1 - \max_j \left(|\mathbf{o}_t^j - \mathbf{o}_{si}^j|\right) \quad (12)$$

where $S_i \in [0, 1]$ denotes the similarity value between the ith source task and the target task, $D_{Cheb}(\cdot,\cdot)$ is the Chebyshev distance function, $\mathbf{o}_t^j$ and $\mathbf{o}_{si}^j$ represent the jth variables of $\mathbf{o}_t$ and $\mathbf{o}_{si}$, respectively.

[4]A formal definition of transferability in S-ESTO is provided in part B.

[5]Without loss of generality, other distance functions can also be used here.

With Eq. (12), we can characterize the similarity relationship between the k source tasks and the target task using the set of similarity values $\{S_i,\ i = 1, 2, ..., k\}$. To eliminate the effect of the varying number of source tasks, one can estimate a probability density function of the set of similarity values using the kernel smoothing technique [61], which is given by,

$$\hat{p}(S) = \mathbb{K}(S_1, ..., S_k) \quad (13)$$

where $\hat{p}(S)$ represents the estimated probability density function, $\mathbb{K}(\cdot)$ denotes the kernel smoothing estimate. For brevity, we term the estimated function similarity distribution.

According to Fig. 3 and Eq. (13), we can see that the similarity distribution of an STOP explicitly describes the similarity relationship between the k source tasks and the target task in the search space. This problem feature quantitatively reflects the distribution of source tasks with different degrees of similarities to the target task, which is influenced by the following three main factors:

- Optimum coverage of the family set(s) of interest.
- Optimum distribution induced by the task distribution.
- Randomness of sampling tasks for constituting the STOP.

Next, we present a toy example to help readers digest all the newly defined concepts and show how similarity distribution changes with the three factors.

E. A Toy Example

Given an 1-dimensional interval with unit length and two stations within the interval, one aims to minimize the radiuses of the two stations with the satisfaction of covering the whole interval. Fig. 4 shows a schematic diagram of this optimization task. More formally, an interval coverage optimization task $f(\mathbf{x};\boldsymbol{l})$ with the feature vector $\boldsymbol{l} = [l_1, l_2]$ (i.e., the locations of the two stations) can be formulated as follows:

$$\min_{\mathbf{x}\in\Omega} x_1 + x_2 \quad (14)$$
$$s.t.\ \text{The interval is covered.}$$

where $\mathbf{x} = [x_1, x_2]$ denotes the decision vector that represents the radiuses of the two stations, Ω represents the decision space configured by a practitioner. It is noteworthy that the box-constrained spaces from $[0, 1]^2$ to $[0, \infty]^2$ are all feasible settings with no influence on the optimal solutions. Next, we use this example to demonstrate the concepts defined earlier and show the necessities of introducing them into S-ESTO.

Task family: The task family $\mathcal{F}(\mathbf{x};\mathcal{L})$ contains an infinite number of elementary tasks characterized by the feature vector $\boldsymbol{l}$ in the feature space $\mathcal{L}:[0,1]^2$. An elementary task $f(\mathbf{x};\boldsymbol{l})$ can be drawn from $\mathcal{F}(\mathbf{x};\mathcal{L})$. The feature space $\mathcal{L}$ here implies that the two stations of an elementary task can be anywhere within the interval. To illustrate, we sample a large number of elementary tasks that are uniformly distributed over the feature space, as shown in the left part of Fig. 5. Each solid square within the shaded area (i.e., $\mathcal{L}$) represents a particular elementary interval coverage optimization task.

For a practitioner who routinely optimizes many such interval coverage tasks, it is the case of STO where a few previously-solved source tasks can be used for better solving a target task. That is why we introduce task family to refer to the task set from which these tasks are obtained.

Task-optimum mapping: The task-optimum mapping $\Phi_{\mathcal{F}}$ of this family maps the features of an interval coverage task to its optimum. To illustrate, we optimize all the elementary tasks sampled before and show the locations of their optima as solid squares in the right part of Fig. 5. The task-optimum mapping is displayed in the form of many dotted parabolas that connect the features of elementary tasks to their optima.

As mentioned earlier, it is desirable to analyze the relationship between the optimal solutions of source and target tasks in S-ESTO. To this end, we can use task-optimum mapping to map a collection of tasks in the family to their optimal solutions for such analysis. The set of optimal solutions of all the elementary tasks is known as the image of task-optimum mapping, i.e., $\mathrm{Im}(\Phi_{\mathcal{F}})$.

Optimum coverage: Optimization coverage is defined as the ratio of the image $\mathrm{Im}(\Phi_{\mathcal{F}})$ to the decision space. It is noted that the size of decision space relies on our prior beliefs about the optimal solutions of a task family. If some priors are available, one always prefers to adopt a smaller decision space to enable an optimizer find the optimum effortlessly. In this example, we consider three cases of decision spaces (i.e., Ω_1, Ω_2, and Ω_3) and calculate the optimum coverages by discretizing the decision spaces. The optimum coverages under the three cases of decision spaces are $\gamma_1 = 0.34$, $\gamma_2 = 0.17$, and $\gamma_3 = 0.01$, respectively. Intuitively, a larger decision space results in a lower optimum coverage.

According to Definition 3, we observe that optimum coverage is a problem-dependent property of a task family, which is controlled by two factors: the intrinsic nature of task family that determines the distribution of optimal solutions and the human priors that influences the size of decision space. Since the source and target tasks in an STOP are alway a subset of elementary tasks in a task family (or multiple families in the inter-family transfer), their similarity relationship is sensitive to optimum coverage, which will be discussed further in similarity distribution below.

Similarity distribution: In this example, we can compose an STOP by sampling an arbitrary number of elementary tasks from $\mathcal{F}(\mathbf{x};\mathcal{L})$. Specifically, 10^3 tasks are optimized and stored into the database to serve as the source tasks, while an unsolved task serves as the target task. The similarity relationship between the 10^3 source tasks and the target task in terms of optimum can be described by similarity distribution defined in Eq. (13). Next, we employ the variable-controlling method to investigate the three aforementioned factors responsible for the variation of similarity distribution in different STOPs.

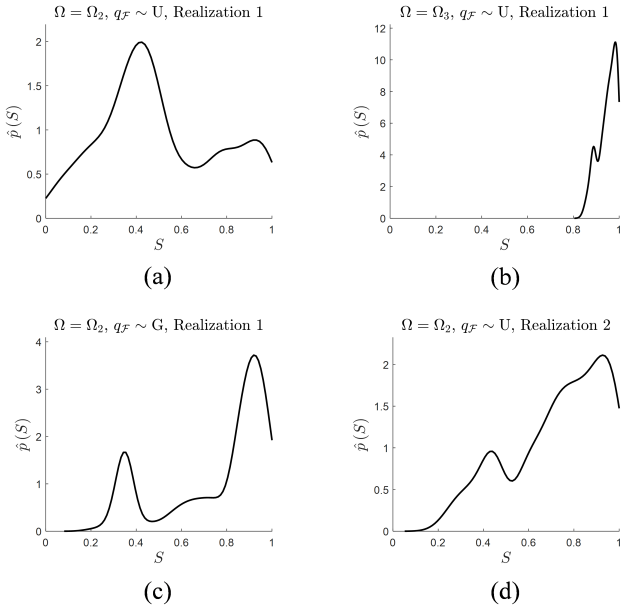
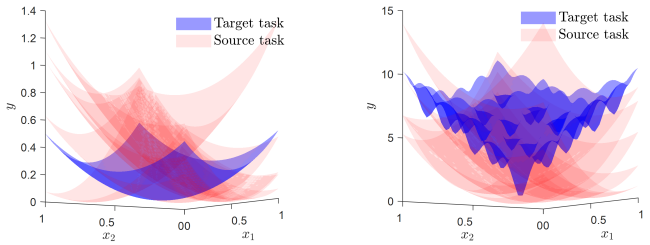


Fig. 6: Similarity distributions of STOPs under different situations: (a) $\Omega = \Omega_2$, $q_{\mathcal{F}} \sim$ U, Realization 1; (b) $\Omega = \Omega_3$, $q_{\mathcal{F}} \sim$ U, Realization 1; (c) $\Omega = \Omega_2$, $q_{\mathcal{F}} \sim$ G, Realization 1; (d) $\Omega = \Omega_2$, $q_{\mathcal{F}} \sim$ U, Realization 2.

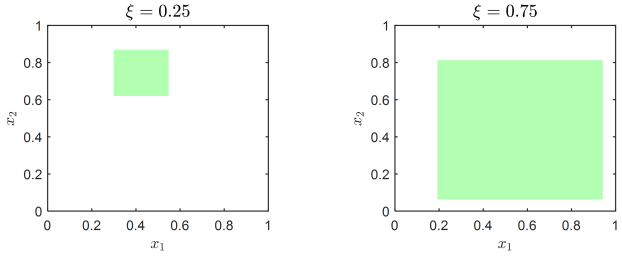
Firstly, we investigate how similarity distribution changes with optimum coverage. The similarity distributions of two STOPs with the same optimum distribution $U[0,1]^2$ but distinct optimum coverages are shown in Fig. 6(a) and Fig. 6(b). It can be seen that similarity distribution is highly susceptible to optimum coverage. As optimum coverage decreases (i.e., from 0.17 under $\Omega = \Omega_2$ to 0.01 under $\Omega = \Omega_3$), the overall similarities of the source tasks to the target task improve a lot. This is because the optimal solutions are closer in the common space when optimum coverage is smaller, thus resulting the higher similarities. Secondly, we investigate the influence of optimum distribution on similarity distribution. Two STOPs with the same optimum coverage under Ω_2 but different optimum distributions are considered, whose similarity distributions are shown in Fig. 6(a) and Fig. 6(c). The optimum distributions are the uniform and Gaussian distributions. It can be seen that the two STOPs with distinct optimum distributions have entirely different similarity distributions. Lastly, we find that the randomness of sampling tasks also greatly impacts similarity distribution. As shown in Fig. 6(a) and Fig. 6(d), the similarity distributions of two STOPs with the same optimum coverage and optimum distribution can differ.

Based on the above analyses, we find that similarity distribution is a highly problem-dependent feature of STOPs. Even for two STOPs from the same task family, their similarity distributions can be entirely different. However, as reviewed in Section II-C, the patterns of similarity relationship in existing test problems are very limited, making them insufficient for evaluating various S-ESTO algorithms. Therefore, to endow



(a) (b)

Fig. 7: Illustrative examples of intra-family and inter-family transfers: (a) intra-family transfer; (b) inter-family transfer.



(a) (b)

Fig. 8: Two box-constrained images of task-optimum mapping in a 2D case: (a) $\xi = 0.25$; (b) $\xi = 0.75$.

test problems with diverse similarity relationships for mimicking complex real-world problems better, one should take this important problem feature (i.e., similarity distribution) into account when designing benchmarks. In part B of this series, we will show that the performance of an S-ESTO algorithm is very sensitive to similarity distribution.

IV. A Problem Generator

In this section, we develop a problem generator of STOP that can flexibly configure similarity distribution. Firstly, we propose a set of guidelines to show how to make similarity distribution of an STOP configurable. After that, a problem generator with detailed algorithmic implementation is proposed. Lastly, we compare the important characteristics of the proposed test problems with those used in previous studies.

A. Design Guidelines

In a problem domain, one always sequentially encounters a variety of optimization tasks with different parameters under the same formulation, which can be modeled as a task family defined in this work. When a certain number of intra-family source tasks are available, we can choose to trigger the intra-family transfer. However, intra-family source tasks may be scarce, especially at the early stage of solving the tasks from a family. In this case, one often employs some out-of-family tasks to serve as source tasks for guiding the target search, which is known as the inter-family transfer.

Given a task family, different elementary tasks normally possess distinct fitness landscapes and thus may have varying optimal solutions. Since optimum is just one of the features of fitness landscape, landscape modality is more resistant to the change of tasks than optimum. In other words, two similar elementary tasks are more likely to have similar landscape modalities with distinct optimal solutions than having close optima under entirely different landscape modalities. For simplicity, intra-family tasks are deemed to possess a common landscape modality with variable optima in this work, which can be realized by a single-objective minimization function with configurable optimum. To illustrate, we present two 2-dimensional examples in Fig. 7, where the source and target tasks are colored red and blue, respectively. In the intra-family transfer, source and target tasks with distinct optimal solutions are drawn from the Sphere family. In the inter-family transfer, the target task is from the Ackley family, while the source tasks are sampled from the Sphere family. In this way, we can configure the functions and optimal solutions of the tasks in an STOP independently. In the intra-family transfer, we can use the distribution of optimal solutions of tasks to model the task distribution directly. In the inter-family transfer, the optimal solution of tasks can be configured in the same way, with the only difference being the outer-family functions for the source tasks. Therefore, the key design guideline of generating an STOP is to configure the optima of its source and target tasks, which can be done via the following two-step procedure:

1. *Range of optima of source and target tasks*: generate the image of task-optimum mapping with adjustable optimum coverage in the common space $\Omega_c = [0, 1]^d$.
2. *Instantiated optima of source and target tasks*: configure the optimal solutions of source and target tasks under a specific similarity distribution within the obtained image.

1) Image of Task-Optimum Mapping: As mentioned before, the optimum coverage of the image from a task family depends on the intrinsic distribution of optima and the decision space. In an extreme case where the optimal solutions of all the elementary tasks are located in a single point, the optimum coverage is 0. The other extreme case is when the image fills the decision space, leading to the optimum coverage of 1. However, most problems are more likely to fall between these two extreme cases. Thus, an image with adjustable optimum coverage is expected to resemble a wider range of problems. For simplicity, we consider box-constrained image[6] of task-optimum mapping, which is given by:

$$\begin{cases} \hat{x}^i_{lb} = rand * (1 - \xi)\,,\ i = 1, 2, ..., d \\ \hat{x}^i_{ub} = \hat{x}^i_{lb} + \xi,\ i = 1, 2, ..., d \end{cases} \tag{15}$$

where $\hat{x}^i_{lb}$ and $\hat{x}^i_{ub}$ denote the lower and upper bounds of the image with respect to the ith variable, respectively, $rand$ is a uniformly distributed random number from 0 to 1, $0 \leq \xi \leq 1$ is a prespecified parameter for adjusting optimum coverage.

Fig. 8 illustrates two 2-dimensional box-constrained images with different optimum coverages. We can see that the images are randomly generated within the common space, and optimum coverage increases with the parameter ξ. Particularly, the decision space is filled by the image when $\xi = 1$, while the optimal solutions of all the elementary tasks are situated on a single point if $\xi = 0$.

[6]Without loss of generality, images with other shapes can also be used.

2) Optima of Source and Target Tasks: As discussed earlier, similarity distribution of STOPs is highly problem-dependent. Different STOPs are likely to have distinct similarity distributions, resulting in varying relationships between the optimal solutions of source and target tasks. To better simulate the diversity of such similarity relationships, our strategy is to consider STOPs with different representative similarity distributions as many as possible. To configure the optimal solutions of source and target tasks, one can sample the latent optimum distribution shown in Eq. (10), which mimics the process of sequentially encountering the optimization tasks in an STOP. It is noted that the latent optimum distribution also depends on the intrinsic nature of a task family. Without loss of generality, we first examine the uniform distribution, in which the optimal solutions of source and target tasks can be obtained as follows:

$$\begin{cases} \mathbf{o}_{si} = \hat{\mathbf{x}}_{lb} + \mathbf{r} \times (\hat{\mathbf{x}}_{ub} - \hat{\mathbf{x}}_{lb}), \ i = 1, 2, ..., k \\ \mathbf{o}_t = \hat{\mathbf{x}}_{lb} + \mathbf{r} \times (\hat{\mathbf{x}}_{ub} - \hat{\mathbf{x}}_{lb}) \end{cases} \tag{16}$$

where $\mathbf{o}_{si}$ is the optimum of the ith source task, $\mathbf{o}_t$ denotes the optimum of the target task, $\mathbf{r} \in [0, 1]^d$ is a randomly generated real vector from the multivariate uniform distribution.

With Eq. (16), we have the optimal solutions of source and target tasks that are uniformly distributed over the image in the common space. If we consider that two tasks are sufficiently similar when their similarity defined in Eq. (12) is greater than $1 - \epsilon$, where ϵ is a very small positive real number, the probability for a source task that is sufficiently similar to the target task can be estimated as follows:

$$\begin{aligned} p(S > 1 - \epsilon) &= p(D_{Cheb} < \epsilon) \\ &= \prod_{j=1}^{d} p\left(|\mathbf{o}_t^j - \mathbf{o}_{si}^j| < \epsilon\right) \\ &\approx (2\epsilon/\xi)^d \end{aligned} \tag{17}$$

The "sufficient similarity" here implies that the source task possesses a Chebyshev distance with no more than ϵ to the target task, thus ensuring the transferability of the source optimum for accelerating the target search. The expected number of source tasks needed to meet the sufficient similarity can be estimated as follows:

$$\eta(\epsilon, \xi, d) = \frac{1}{p(\mathcal{S} > 1 - \epsilon)} = \left(\frac{\xi}{2\epsilon}\right)^d \tag{18}$$

When $\epsilon = 0.1$, the contour graph of η with respect to ξ and d is illustrated in Fig. 9. From the figure, we can see that sufficient similarity is consistently satisfied by a small number of source tasks when ξ is close to 0 regardless of the problem dimension. However, on problems with large optimum coverages, the number of source tasks needed to satisfy the sufficient similarity increases exponentially. Specifically, when $\xi = 1$ and $d = 10$, η is about 10^{10}. This is impractical in real-world applications since the number of source tasks is always limited. In other words, the transferability of source solutions is difficult to be ensured when the optimal solutions are uniformly distributed over a large image of task-optimum mapping, especially on high-dimensional problems with a limited number of source tasks. To examine the relationship

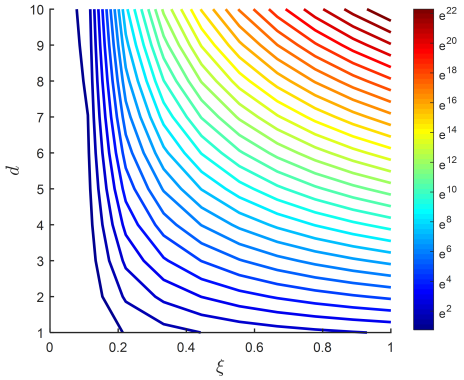


Fig. 9: Contour graph of η w.r.t. ξ and d when $\epsilon = 0.1$.

between the optimal solutions configured using Eq. (16), we generate 10^4 source tasks under different ξ to observe the similarity distribution defined in Eq. (13), as shown in Fig. 11(a), Fig. 11(e), and Fig. 11(i). We can see from Fig. 11(a) that the sufficient similarity is consistently satisfied when ξ is zero regardless of the problem dimension. However, as optimum coverage increases, the cumulative probability of meeting the sufficient similarity decreases a lot, as illustrated in Fig. 11(e) and Fig. 11(i). The results here are consistent with the observations in Fig. 9. It is worth noting that the assumption for optimum distribution here (i.e., the uniform distribution) may not hold on many other task families due to its problem-dependent nature, on which the sufficient similarity may be easier to be satisfied. Since our ultimate goal is to generate STOPs with diverse similarity distributions, we propose to configure similarity distribution directly without the need of sampling the latent optimum distribution. This can be done by introducing a weight parameter $\tau \in [0, 1]$ used for controlling the relationship between the optimal solutions of source and target tasks,

$$\begin{cases} \mathbf{o}_{si}^b = \hat{\mathbf{x}}_{lb} + \mathbf{r} \times (\hat{\mathbf{x}}_{ub} - \hat{\mathbf{x}}_{lb}), \ i = 1, 2, ..., k \\ \mathbf{o}_t = \hat{\mathbf{x}}_{lb} + \mathbf{r} \times (\hat{\mathbf{x}}_{ub} - \hat{\mathbf{x}}_{lb}) \\ \mathbf{o}_{si} = \mathbf{o}_t \times (1 - \tau_i) + \mathbf{o}_{si}^b \times \tau_i, \ i = 1, 2, ..., k \end{cases} \tag{19}$$

where τ_i is the weight parameter imposed on the optimum of the ith source task. We can see that Eq. (19) turns into Eq. (16) when τ is equal to a constant value of 1. By adopting different distributions of τ, one can obtain a number of representative similarity distributions. In this study, we consider three types of distributions of τ, whose probability density functions are presented as follows:

- The uniform distribution: $p_d(\tau) = 1, \tau \in [0, 1]$.
- The increasing distribution: $p_d(\tau) = 2\tau, \tau \in [0, 1]$.
- The decreasing distribution: $p_d(\tau) = 2 - 2\tau, \tau \in [0, 1]$.

Fig. 10 shows the distribution of τ corresponding to Eq. (16) and the three customized distributions of τ. In the uniform optimum distribution, we do not enforce the relationship between the optimal solutions of source and target tasks by setting $\tau = 1$, as shown in Fig. 10(a). In addition, we consider three representative similarity distributions, in which the relationship between the optimal solutions of source and target tasks are governed by the customized distributions of τ

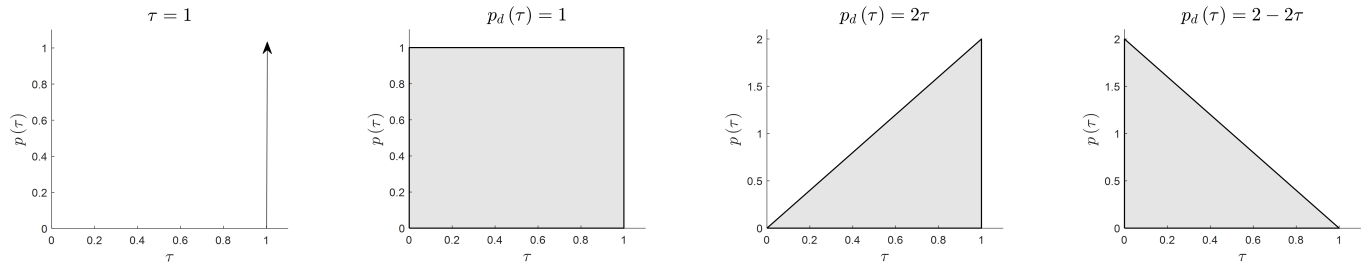


Fig. 10: Four types of probability density functions of τ: (a) $\tau = 1$; (b) $p_d(\tau) = 1$; (c) $p_d(\tau) = 2\tau$; (d) $p_d(\tau) = 2 - 2\tau$.

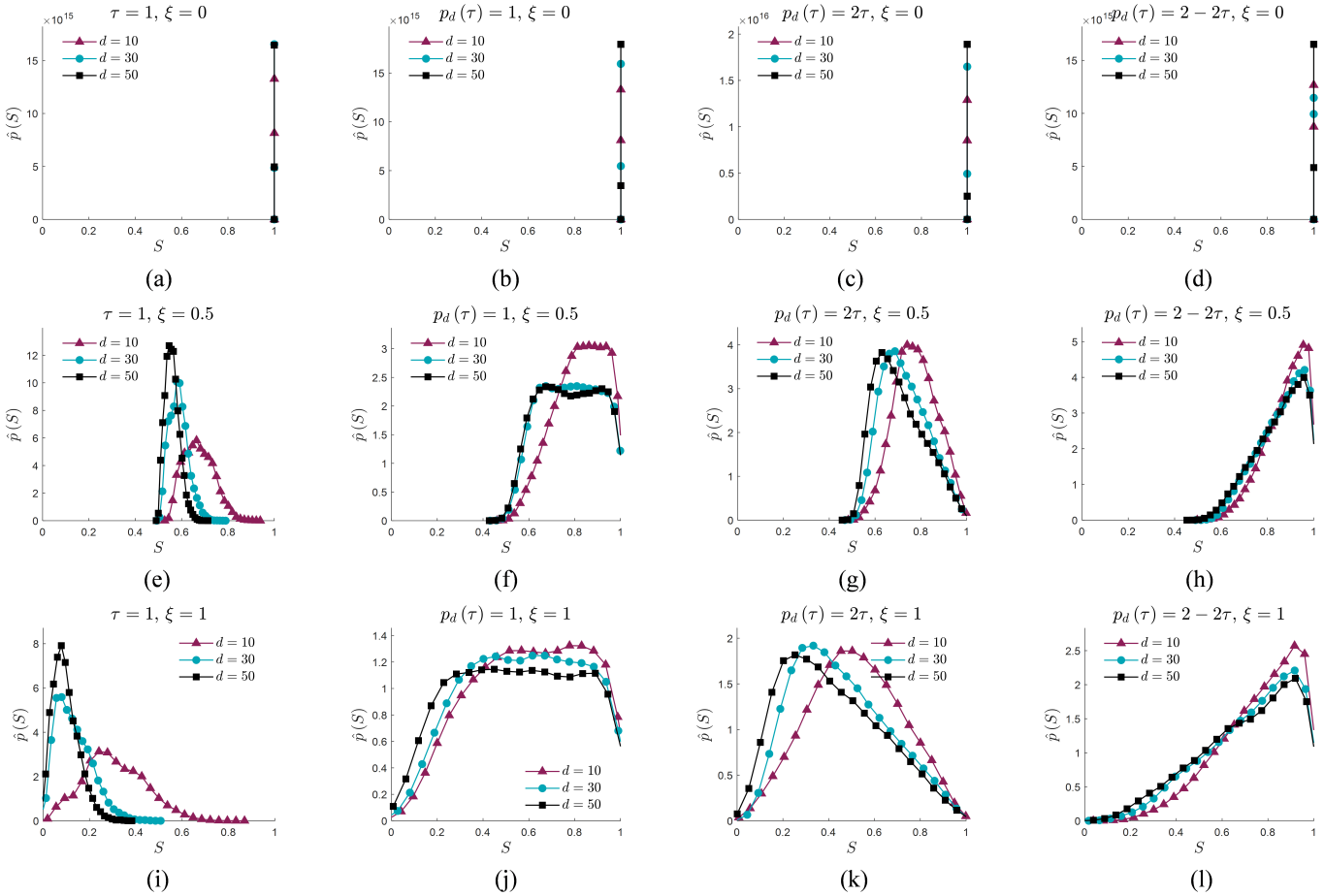


Fig. 11: Four types of similarity distributions under different ξ: (a) $\hat{p}_c$, $\xi = 0$; (b) $\hat{p}_u$, $\xi = 0$; (c) $\hat{p}_i$, $\xi = 0$; (d) $\hat{p}_d$, $\xi = 0$; (e) $\hat{p}_c$, $\xi = 0.5$; (f) $\hat{p}_u$, $\xi = 0.5$; (g) $\hat{p}_i$, $\xi = 0.5$; (h) $\hat{p}_d$, $\xi = 0.5$; (i) $\hat{p}_c$, $\xi = 1$; (j) $\hat{p}_u$, $\xi = 1$; (k) $\hat{p}_i$, $\xi = 1$; (l) $\hat{p}_d$, $\xi = 1$.

shown in Fig. 10(b) to Fig. 10(d). For brevity, the similarity distributions corresponding to the distributions of τ from Fig. 10(a) to Fig. 10(d) are denoted by $\hat{p}_c$, $\hat{p}_u$, $\hat{p}_i$, and $\hat{p}_d$, respectively. To illustrate, we generate 10^4 source tasks using Eq. (19) under each of the customized distributions of τ to obtain their similarity distributions. As shown in Fig. 11(a) to Fig. 11(d), when optimum coverage is extremely small, all the source tasks from any of the similarity distributions possess incredibly high similarities to the target task. This is because the optimal solutions of source and target tasks are exactly the same when optimum coverage is 0. However, as can be seen from Fig. 11(e) to Fig. 11(l), the difference between the similarity distributions gradually shows up as optimum coverage increases. As discussed earlier, when optimal solutions are uniformly distributed over the image under $\xi = 1$, the transferability of source tasks is difficult to be ensured. By contrast, the three similarity distributions based on the customized distributions of τ can provide a certain number of source tasks similar to the target task, thus ensuring that there are always a few source tasks containing highly transferable solutions for guiding the target search. Moreover, the proportions of source tasks that show different degrees of similarities to the target task can be flexibly adjusted by modifying the distribution $p_d(\tau)$, as demonstrated in Fig. 11(i)

TABLE I: The important characteristics of our problems against those in previous studies.

Publications	The Number of Source Tasks (k)	Transfer Scenario ($\mathcal{T}$)	Similarity Distribution ($\hat{p}$)
[14, 18, 47]	fixed	inter-family	not applicable, fixed
[16]	fixed	inter-family	$\hat{p}_c$, fixed
[5, 34]	adjustable	intra-family	$\hat{p}_c$, fixed
Ours	adjustable	intra-family & inter-family	$\hat{p}_c$, $\hat{p}_u$, $\hat{p}_i$, $\hat{p}_d$, adjustable

Algorithm 1: A Problem Generator of STOP

Input: Υ (candidate families), t (the index of target family), $\mathcal{T}$ (transfer scenario), ξ (the parameter that controls optimum coverage), $p_d(\tau)$ (the distribution that controls similarity distribution), d (dimension), k (the number of source tasks), $\mathbb{O}$ (evolutionary optimizer)

Output: $f^t(\mathbf{x})$ (the target task), $\mathcal{M} = \{\mathcal{P}^{s1}, ..., \mathcal{P}^{sk}\}$ (the knowledge base)

// Generate the source and target tasks

1 $[\hat{\mathbf{x}}_{lb}, \hat{\mathbf{x}}_{ub}] \leftarrow$ Eq. (15) based on ξ;

2 $[\tau_1, ..., \tau_k] \leftarrow$Sample k weight scalars from $p_d(\tau)$;

3 $[\mathbf{o}_t, \mathbf{o}_{s1}, ..., \mathbf{o}_{sk}] \leftarrow$ Eq. (19) using $[\hat{\mathbf{x}}_{lb}, \hat{\mathbf{x}}_{ub}]$ and $[\tau_1, ..., \tau_k]$;

4 $f^t(\mathbf{x}) \leftarrow \Upsilon(t, \mathbf{o}_t)$;

5 **for** $i = 1$ to k **do**

6 | **if** $\mathcal{T} = T_a$ **then** // intra-family transfer

7 | | $s \leftarrow t$;

8 | **else** // inter-family transfer

9 | | $s \leftarrow$Randomly select from $I_\Upsilon \setminus t$;

10 | $f_i^s(\mathbf{x}) \leftarrow \Upsilon(s, \mathbf{o}_{si})$;

// Construct the knowledge base

11 $\mathcal{M} = \emptyset$;

12 **for** $i = 1$ to k **do**

13 | $\mathcal{P}^{si} = \text{EvolutionarySearch}(\mathbb{O}, f_i^s)$;

14 | $\mathcal{M} \leftarrow \mathcal{M} \cup \mathcal{P}^{si}$;

15 **return** $f^t(\mathbf{x})$ and $\mathcal{M}$;

to Fig. 11(l). Here, as can be observed, we are not chasing for a single all-embracing similarity distribution, as it can only represent a specific class of problems. Instead, we consider several representative similarity distributions to mimic the diverse similarity relationships of real-world problems.

B. A Problem Generator

According to the design guidelines discussed above, one can build an STOP by instantiating three aspects: transfer scenario, box-constrained image, and similarity distribution. Algorithm 1 summarizes the detailed implementation of the proposed problem generator. When generating the source and target tasks, we configure their functions and optimal solutions independently. Firstly, the optimal solutions of the source and target tasks are determined based on the settings of ξ and $p_d(\tau)$, as shown in lines 1 to 3. Then, the objective functions of the source and target tasks are determined based on the transfer scenario and the obtained optimal solutions, as shown in lines 4 to 10. For brevity, the intra-family and inter-family transfers are represented by T_a and T_e, respectively. Lastly, we need to optimize the k source tasks using an evolutionary optimizer and store their evaluated solutions into the knowledge base $\mathcal{M}$, to complete the construction of the STOP. The optimizer-independent feature enables any other evolutionary optimizers to be embedded into the generator for solving source tasks. Following the implementation in Algorithm 1, we name an STOP as $\mathcal{F}$-$\mathcal{T}$-ξ-$\mathcal{S}$-d-k, where $\mathcal{F}$ denotes the target family, $\mathcal{T}$ represents the transfer scenario, ξ is the optimum coverage-related parameter, $\mathcal{S}$ represents the distribution $p_d(\tau)$ that governs similarity distribution, d denotes the problem dimension, k is the number of source tasks.

C. Comparison with Existing Test Problems

Table I provides the important characteristics of our problems against those used in previous studies. In [14, 18, 47], the problems are composited by the commonly used benchmarks in multiobjective optimization. The source and target tasks in the derived STOPs have distinct objective functions but possess the same optima, which can be seen as a specific class of problems produced by the generator under $\mathcal{T} = T_e$ and $\xi = 0$. The similarity distribution of these problems is exactly the one shown in Fig. 11(a), which lacks the desired diversity of similarity relationship. To mimic the heterogeneity of problems in terms of optimum, the authors in [16] use the randomly generated optima to configure their STOPs. Since the optimal solutions are uniformly sampled in the common space, the resultant similarity distribution can be seen as $\hat{p}_c$. In this way, the test problems in [16] can be treated as a special class of STOPs with the specific similarity distribution $\hat{p}_c$. Moreover, the number of source tasks in all of the above STOPs is fixed. In [5, 34], an arbitrary number of source tasks are synthesized based on the target task. However, the diversity of the generated tasks is greatly constrained to ensure the similarities between the source and target tasks, which thus leads to a similarity distribution with incredibly high similarity values. In this sense, the STOPs in [5, 34] belong to a special type of problems from the proposed generator under $\mathcal{T} = T_a$ and small ξ.

In summary, the highlights of our proposed problem generator are as follows. (1) The number of source tasks is adjustable. (2) Both the intra- and inter-family transfer scenarios are available. (3) Unlike the fixed patterns of similarity distribution in existing problems, similarity distribution of our test problems can be systematically configured by modifying the distribution $p_d(\tau)$, leading to a closer resemblance of the diversity of similarity relationship in real-world problems.

V. A Benchmark Suite

In this section, we design a benchmark suite containing 12 STOPs with varying properties. Firstly, the available realizations of the parameters for building STOPs are provided.

Then, we develop a benchmark suite with detailed problem specifications. Lastly, the evaluation criterion for comparing S-ESTO algorithms on the developed benchmark suite is provided.

A. Available Realizations of the Parameters

According to Algorithm 1, we can see that there are six necessary parameters for building STOPs: task family, transfer scenario, optimum coverage of the box-constrained image, similarity distribution, problem dimension, and the number of source tasks. Here, we provide the available realizations of these six parameters, which will then be used for designing benchmarks.

1) Task Family: We choose eight single-objective functions with a configurable optimum to serve as candidate families due to their widespread use in the area of continuous optimization.

– Sphere family:

$$\min f_1(\mathbf{x}) = \sum_{i=1}^{d} (x_i - o_i)^2, \mathbf{x} \in [-100, 100]$$

– Ellipsoid family:

$$\min f_2(\mathbf{x}) = \sum_{i=1}^{d} (d - i + 1)(x_i - o_i)^2, \mathbf{x} \in [-50, 50]$$

– Schwefel 2.2 family:

$$\min f_3(\mathbf{x}) = \sum_{i=1}^{d} |x_i - o_i| + \prod_{i=1}^{d} |x_i - o_i|, \mathbf{x} \in [-30, 30]$$

– Quartic family with noise:

$$\min f_4(\mathbf{x}) = \varepsilon + \sum_{i=1}^{d} i \times (x_i - o_i)^4, \mathbf{x} \in [-5, 5]$$

– Ackley family:

$$\begin{aligned}\min f_5(\mathbf{x}) = &-20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} z_i^2}\right) - \exp\left(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi z_i)\right) \\ &+ 20 + e,\ z_i = x_i - o_i,\ \mathbf{x} \in [-32, 32]\end{aligned}$$

– Rastrigin family:

$$\begin{aligned}\min f_6(\mathbf{x}) = &\sum_{i=1}^{d} \left\{ (x_i - o_i)^2 + 10\cos\left[2\pi (x_i - o_i)\right] \right\} \\ &+ 10d,\ \mathbf{x} \in [-10, 10]\end{aligned}$$

– Griewank family:

$$\begin{aligned}\min f_7(\mathbf{x}) = &1 + \frac{1}{4000}\sum_{i=1}^{d} (x_i - o_i)^2 \\ &- \prod_{i=1}^{d} \cos\left(\frac{x_i - o_i}{\sqrt{i}}\right), \mathbf{x} \in [-200, 200]\end{aligned}$$

– Levy family:

$$\begin{aligned}\min f_8(\mathbf{x}) = &\sin^2(\pi\omega_1) + \sum_{i=1}^{d-1} (\omega_i - 1)^2 \left[1 + 10\sin^2(\pi\omega_i + 1)\right] + \\ &(\omega_d - 1)^2 \left[1 + \sin^2(2\pi\omega_d)\right],\ \omega_i = 1 + \frac{z_i}{4},\ \mathbf{x} \in [-20, 20]\end{aligned}$$

where d is the problem dimension, x_i denotes the ith decision variable, and o_i represents the optimal solution of the ith variable, $\varepsilon \sim U(0, 1)$ denotes the random noise. The eight functions here constitute the set of candidate families (i.e., Υ) in Algorithm 1. It should be noted that the design here is not limited to the eight selected functions and can be generalized to any set of functions with a configurable optimum.

2) Transfer Scenario: In this work, we consider the intra- and inter-family transfers. In the intra-family transfer, source and target tasks are from the same family. By contrast, source tasks in the inter-family transfer are from the families that differ from the target one, as shown in line 9 of Algorithm 1.

3) Optimum Coverage of the Image: By adjusting the parameter $\xi \in [0, 1]$ in Eq. (15), we can obtain a box-constrained image with an arbitrary value of optimum coverage, ranging from 0 to 1. Moreover, by introducing a parameterized manifold to describe the image, one can construct a nonlinear image with adjustable optimum coverage to meet their own needs.

4) Similarity Distribution: Instead of modeling optimum distribution explicitly, we propose to use a parameterized distribution $p_d(\tau)$ to manipulate the similarity relationship between the optimal solutions of the source and target tasks in STOPs, thus enabling us to generate different similarity distributions directly. Four customized similarity distribution, i.e., $\hat{p}_c$, $\hat{p}_u$, $\hat{p}_i$, and $\hat{p}_d$, are generated to mimic the diversity of similarity relationships in various STOPs. The design here also allows one to construct STOPs with customized similarity distributions to meet their own needs.

5) Problem Dimension: Following the from-simple-to-complex way of research [62], we suggest the problem dimensions from 25 to 50 to avoid the curse of dimensionality. Notably, different STOPs can be configured with different dimensions, but the source and target tasks in a single STOP have the same dimensions.

6) The Number of Source Tasks: The possible values of the number of source tasks are positive integers ranging from 1 to ∞. Both the single-source and multi-source scenarios can be set up using the proposed problem generator. Table II summarizes the available realizations of the six parameters. With these realizations, we formulate a benchmark suite with 12 individual STOPs in the next subsection.

TABLE II: The available realizations of the six parameters for building STOPs.

Aspect	Configuration
Task Family (Υ)	$\{f_1, f_2, f_3, f_4, f_5, f_6, f_7, f_8\}$
Transfer Scenario ($\mathcal{T}$)	$\{T_a, T_e\}$
Optimum Coverage (γ)	$\xi \in [0, 1]$
Similarity Distribution ($\hat{p}(S)$)	$\{\hat{p}_c, \hat{p}_u, \hat{p}_i, \hat{p}_d\}$
Problem Dimension (d)	$N+$
The Number of Source Tasks (k)	$N+$

B. A Benchmark Suite

One of key rationales behind designing benchmark problems is to provide a set of representative test problems with diverse properties, which can create a good resemblance to a wide range of real-world problems [63]. Thus, the configurations

TABLE III: A benchmark suite of STOPs.

Similarity Relationship	Problem Specification ($\mathcal{F}$-$\mathcal{T}$-ξ-$\mathcal{G}$-d-k)	Problem ID
High Similarity (HS)	Sphere-T_a-0-$\hat{p}_c$-50-k	STOP 1
	Ellipsoid-T_e-0-$\hat{p}_u$-25-k	STOP 2
	Schwefel-T_a-0-$\hat{p}_i$-30-k	STOP 3
	Quartic-T_e-0-$\hat{p}_d$-50-k	STOP 4
Mixed Similarity (MS)	Ackley-T_a-1-$\hat{p}_i$-25-k	STOP 5
	Rastrigin-T_e-1-$\hat{p}_u$-50-k	STOP 6
	Griewank-T_a-0.7-$\hat{p}_i$-25-k	STOP 7
	Levy-T_e-0.7-$\hat{p}_d$-30-k	STOP 8
Low Similarity (LS)	Sphere-T_a-1-$\hat{p}_c$-25-k	STOP 9
	Rastrigin-T_e-0.7-$\hat{p}_c$-30-k	STOP 10
	Ackley-T_a-0.7-$\hat{p}_c$-50-k	STOP 11
	Ellipsoid-T_e-1-$\hat{p}_c$-50-k	STOP 12

in Table II should be adequately considered when designing an STOP benchmark suite. In this study, we develop three categories of problems with different types of similarity relationships between source and target tasks: 1) STOPs full of source tasks that are highly similar to the target task in terms of optimum; 2) STOPs containing source tasks with mixed similarities to the target task in terms of optimum; 3) STOPs full of source tasks with low similarities to the target task in terms of optimum. These three categories of STOPs are termed HS, MS, and LS problems for short, respectively, each of which contains four individual STOPs. Thus, a benchmark suite with a total number of 12 STOPs is built here. The eight task families and two transfer scenarios are alternately configured across the 12 problems. The number of source tasks is a user-defined parameter, which can be configured to cater to the needs of studies on STOPs with different amount of optimization experience.

Table III lists the twelve problems in the benchmark suite, which can be readily produced by the proposed generator (i.e., Algorithm 1) according to the specifications. For the first category of problems from STOP 1 to STOP 4, their optimum coverages are zero so that the source and target tasks in the STOPs share the same optimal solution, making the optimal solutions of all the source tasks are highly transferable for accelerating the target search. For the second category of problems from STOP 5 to STOP 8, three customized distributions of τ (i.e., $\hat{p}_u$, $\hat{p}_i$, and $\hat{p}_d$) are used to adjust the relationship between the optimal solutions of source and target tasks based on Eq. (19), making the proportion of source tasks with different degrees of similarities to the target task flexibly adjustable, as shown from Fig. 11(j) to Fig. 11(l). Therefore, STOP 5 to STOP 8 contain source tasks with mixed similarities to the target task. As for the last category of problems from STOP 9 to STOP 12, the optimal solutions of source and target tasks are configured using the similarity distribution $\hat{p}_c$, often making the optimal solutions of all the source tasks dissimilar to the target optimum, as shown in Fig. 11(i). Consequently, the optimal solutions of all the source tasks in STOP 9 to STOP 12 are unhelpful for accelerating the target search. It is worth mentioning that similarity here indicates that whether the unadapted optimal solution of a source task is similar to the target one. If the solution can be adapted properly, its similarity to the target optimum can be improved a lot.

C. Evaluation

The minimum value of all the target tasks in the benchmark suite is zero. When comparing multiple algorithms using the proposed benchmarks, the computational budget in terms of function evaluation allocated to each algorithm should be the same. Meanwhile, the usage of machine learning models that may require high computational costs should be carefully examined despite the promising results reported since our benchmarks are not considered computationally expensive. To examine the significance of the results, we recommend the methods of using statistical significance tests in [64].

VI. Conclusion

In this part of the series, we have first proposed a number of fire-new concepts to characterize tasks in an STOP, including task family, task-optimum mapping, optimum coverage, and similarity distribution. Particularly, similarity distribution is an important problem feature of STOPs ignored by previous studies. The empirical results in part B reveal that the performance of an S-ESTO algorithm is highly susceptible to similarity distribution. Therefore, this important problem feature should be adequately taken into account when designing benchmark STOPs. Since similarity distribution of STOPs is highly problem-dependent, our strategy is to make it flexibly configurable to mimic the diversity of similarity relationship in real-world problems. To this end, we propose general design guidelines to generate an STOP whose similarity distribution can be flexibly adjusted by modifying a parameterized distribution $p_d(\tau)$. Based on the design guidelines, we have proposed a problem generator with superior extendability and scalability, allowing one to generate STOPs to cater to their own studies' needs. Lastly, to build an arena for analyzing S-ESTO algorithms, we have developed a benchmark suite with 12 individual STOPs using the proposed problem generator. To the best of our knowledge, this is the first study on how to generate STOPs systematically.

In the next part of this series, we will empirically revisit many knowledge transfer methods in the context of S-ESTO on the benchmark problems developed in this part. Five key research questions related to knowledge transfer techniques are addressed to help practitioners and researchers gain a deeper understanding of how to better exploit the optimization experience using S-ESTO approaches. Moreover, we present a few potential challenges and promising future directions to facilitate the development of S-ESTO algorithms.

References

[1] A. Gupta, Y.-S. Ong, and L. Feng, "Insights on transfer optimization: Because experience is the best teacher," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 1, pp. 51–64, 2017.

[2] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.

[3] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A comprehensive survey on transfer learning," *Proceedings of the IEEE*, vol. 109, no. 1, pp. 43–76, 2020.

[4] P. Cunningham and B. Smyth, "Case-based reasoning in scheduling: reusing solution components," *International Journal of Production Research*, vol. 35, no. 11, pp. 2947–2962, 1997.

[5] S. J. Louis and J. McDonnell, "Learning with case-injected genetic algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 4, pp. 316–328, 2004.

[6] A. T. W. Min, A. Gupta, and Y.-S. Ong, "Generalizing transfer bayesian optimization to source-target heterogeneity," *IEEE Transactions on Automation Science and Engineering*, vol. 18, no. 4, pp. 1754–1765, 2020.

[7] K. C. Tan, L. Feng, and M. Jiang, "Evolutionary transfer optimization-a new frontier in evolutionary computation research," *IEEE Computational Intelligence Magazine*, vol. 16, no. 1, pp. 22–33, 2021.

[8] A. Gupta, Y.-S. Ong, and L. Feng, "Multifactorial evolution: toward evolutionary multitasking," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 3, pp. 343–357, 2015.

[9] A. Gupta, Y.-S. Ong, L. Feng, and K. C. Tan, "Multiobjective multifactorial optimization in evolutionary multitasking," *IEEE transactions on cybernetics*, vol. 47, no. 7, pp. 1652–1665, 2016.

[10] L. Feng, L. Zhou, J. Zhong, A. Gupta, Y.-S. Ong, K.-C. Tan, and A. K. Qin, "Evolutionary multitasking via explicit autoencoding," *IEEE transactions on cybernetics*, vol. 49, no. 9, pp. 3457–3470, 2018.

[11] L. Zhang, Y. Xie, J. Chen, L. Feng, C. Chen, and K. Liu, "A study on multiform multi-objective evolutionary optimization," *Memetic Computing*, vol. 13, pp. 307–318, 2021.

[12] Y. Feng, L. Feng, S. Kwong, and K. C. Tan, "A multi-variation multifactorial evolutionary algorithm for large-scale multi-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 26, no. 2, pp. 248–262, 2022.

[13] F. Yin, X. Xue, C. Zhang, K. Zhang, J. Han, B. Liu, J. Wang, and J. Yao, "Multifidelity genetic transfer: an efficient framework for production optimization," *SPE Journal*, vol. 26, no. 04, pp. 1614–1635, 2021.

[14] L. Feng, Y. Ong, S. Jiang, and A. Gupta, "Autoencoding evolutionary search with learning across heterogeneous problems," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 5, pp. 760–772, 2017.

[15] M. Shakeri, E. Miahi, A. Gupta, and Y.-S. Ong, "Scalable transfer evolutionary optimization: Coping with big task instances," *IEEE Transactions on Cybernetics*, 2022, accepted, DOI: 10.1109/TCYB.2022.3164399.

[16] X. Xue, C. Yang, Y. Hu, K. Zhang, Y.-m. Cheung, L. Song, and K. C. Tan, "Evolutionary sequential transfer optimization for objective-heterogeneous problems," *IEEE Transactions on Evolutionary Computation*, vol. 26, no. 5, pp. 1424–1438, 2022.

[17] H. H. Hoos, "Automated algorithm configuration and parameter tuning," in *Autonomous search*. Springer, 2011, pp. 37–71.

[18] B. Da, A. Gupta, and Y.-S. Ong, "Curbing negative influences online for seamless transfer evolutionary optimization," *IEEE Transactions on Cybernetics*, vol. 49, no. 12, pp. 4365–4378, 2018.

[19] F. Hutter, H. H. Hoos, K. Leyton-Brown, and T. Stützle, "ParamILS: an automatic algorithm configuration framework," *Journal of Artificial Intelligence Research*, vol. 36, pp. 267–306, 2009.

[20] C. Huang, Y. Li, and X. Yao, "A survey of automatic parameter tuning methods for metaheuristics," *IEEE transactions on evolutionary computation*, vol. 24, no. 2, pp. 201–216, 2019.

[21] J. R. Rice, "The algorithm selection problem," in *Advances in computers*. Elsevier, 1976, vol. 15, pp. 65–118.

[22] K. A. Smith-Miles, "Cross-disciplinary perspectives on meta-learning for algorithm selection," *ACM Computing Surveys (CSUR)*, vol. 41, no. 1, pp. 1–25, 2009.

[23] L. Xu, H. Hoos, and K. Leyton-Brown, "Hydra: Automatically configuring algorithms for portfolio-based selection," in *Twenty-Fourth AAAI Conference on Artificial Intelligence*, 2010.

[24] K. Tang, F. Peng, G. Chen, and X. Yao, "Population-based algorithm portfolios with automated constituent algorithms selection," *Information Sciences*, vol. 279, pp. 94–104, 2014.

[25] K. Tang, S. Liu, P. Yang, and X. Yao, "Few-shots parallel algorithm portfolio construction via co-evolution," *IEEE Transactions on Evolutionary Computation*, vol. 25, no. 3, pp. 595–607, 2021.

[26] S. Liu, K. Tang, and X. Yao, "Experience-based optimization: A coevolutionary approach," *arXiv preprint arXiv:1703.09865*, 2017.

[27] M. A. Muñoz, Y. Sun, M. Kirley, and S. K. Halgamuge, "Algorithm selection for black-box continuous optimization problems: A survey on methods and challenges," *Information Sciences*, vol. 317, pp. 224–245, 2015.

[28] M. W. Hauschild, M. Pelikan, K. Sastry, and D. E. Goldberg, "Using previous models to bias structural learning in the hierarchical boa," *Evolutionary Computation*, vol. 20, no. 1, pp. 135–160, 2012.

[29] M. Pelikan and M. W. Hauschild, "Learn from the past: Improving model-directed optimization by transfer learning based on distance-based bias," *Missouri Estimation of Distribution Algorithms Laboratory, University of Missouri in St. Louis, MO, United States, Tech. Rep*, vol. 2012007, 2012.

[30] S. Friess, P. Tiňo, S. Menzel, B. Sendhoff, and X. Yao, "Improving sampling in evolution strategies through mixture-based distributions built from past problem instances," in *International Conference on Parallel Problem Solving from Nature*. Springer, 2020, pp. 583–596.

[31] A. Gupta and Y.-S. Ong, *Memetic computation: the mainspring of knowledge transfer in a data-driven optimization era*. Springer, 2018, vol. 21.

[32] R. Lim, L. Zhou, A. Gupta, Y.-S. Ong, and A. N. Zhang, "Solution representation learning in multi-objective transfer evolutionary optimization," *IEEE Access*, vol. 9, pp. 41 844–41 860, 2021.

[33] R. Lim, A. Gupta, Y.-S. Ong, L. Feng, and A. N. Zhang, "Non-linear domain adaptation in transfer evolutionary optimization," *Cognitive Computation*, vol. 13, no. 2, pp. 290–307, 2021.

[34] J. Zhang, W. Zhou, X. Chen, W. Yao, and L. Cao, "Multisource selective transfer framework in multiobjective optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 3, pp. 424–438, 2020.

[35] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE transactions on evolutionary computation*, vol. 1, no. 1, pp. 67–82, 1997.

[36] T. Wei, S. Wang, J. Zhong, D. Liu, and J. Zhang, "A review on evolutionary multi-task optimization: Trends and challenges," *IEEE Transactions on Evolutionary Computation*, vol. 26, no. 5, pp. 941–960, 2022.

[37] E. K. Burke, B. MacCarthy, S. Petrovic, and R. Qu, "Structured cases in case-based reasoning—re-using and adapting cases for time-tabling problems," *Knowledge-Based Systems*, vol. 13, no. 2-3, pp. 159–165, 2000.

[38] W.-J. Yin, M. Liu, and C. Wu, "A genetic learning approach with case-based memory for job-shop scheduling problems," in *Proceedings. International Conference on Machine Learning and Cybernetics*, vol. 3. IEEE, 2002, pp. 1683–1687.

[39] S. J. Louis and Y. Zhang, "A sequential similarity metric for case injected genetic algorithms applied to TSPs," in *Proceedings of the Genetic and Evolutionary Computation Conference*, vol. 1, 1999, pp. 377–384.

[40] L. Feng, Y.-S. Ong, I. W.-H. Tsang, and A.-H. Tan, "An evolutionary search paradigm that learns with past experiences," in *2012 IEEE Congress on Evolutionary Computation*. IEEE, 2012, pp. 1–8.

[41] L. Feng, Y. Huang, I. W. Tsang, A. Gupta, K. Tang, K. C. Tan, and Y.-S. Ong, "Towards faster vehicle routing by transferring knowledge from customer representation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 2, pp. 952–965, 2022.

[42] M. Omidvar, X. Li, and X. Yao, "A review of population-based metaheuristics for large-scale black-box global optimization–Part I," *IEEE Transactions on Evolutionary Computation*, vol. 26, no. 5, pp. 802–822, 2022.

[43] D. R. Hains, L. D. Whitley, and A. E. Howe, "Revisiting the big valley search space structure in the TSP," *Journal of the Operational Research Society*, vol. 62, no. 2, pp. 305–312, 2011.

[44] S. García, D. Molina, M. Lozano, and F. Herrera, "A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the CEC'2005 special session on real parameter optimization," *Journal of Heuristics*, vol. 15, no. 6, pp. 617–644, 2009.

[45] R. Poli, J. Kennedy, and T. Blackwell, "Particle swarm optimization," *Swarm intelligence*, vol. 1, no. 1, pp. 33–57, 2007.

[46] S. J. Louis and C. Miles, "Playing to learn: case-injected genetic algorithms for learning to play computer games," *IEEE Transactions on Evolutionary Computation*, vol. 9, no. 6, pp. 669–681, 2005.

[47] L. Zhou, L. Feng, A. Gupta, and Y.-S. Ong, "Learnable evolutionary search across heterogeneous problems via kernelized autoencoding," *IEEE Transactions on Evolutionary Computation*, vol. 25, no. 3, pp. 567–581, 2021.

[48] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: Empirical results," *Evolutionary computation*, vol. 8, no. 2, pp. 173–195, 2000.

[49] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable test problems for evolutionary multiobjective optimization," in *Evolutionary multiobjective optimization*. Springer, 2005, pp. 105–145.

[50] H. R. Maier, S. Razavi, Z. Kapelan, L. S. Matott, J. Kasprzyk, and B. A.

Tolson, "Introductory overview: Optimization using evolutionary algorithms and other metaheuristics," *Environmental modelling & software*, vol. 114, pp. 195–213, 2019.
[51] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.
[52] Y. Peng, B. Choi, and J. Xu, "Graph learning for combinatorial optimization: A survey of state-of-the-art," *Data Science and Engineering*, vol. 6, no. 2, pp. 119–141, 2021.
[53] Y. Shen, Y. Shi, J. Zhang, and K. B. Letaief, "LORM: Learning to optimize for resource management in wireless networks with few training samples," *IEEE Transactions on Wireless Communications*, vol. 19, no. 1, pp. 665–679, 2019.
[54] B. Li, G. Wu, Y. He, M. Fan, and W. Pedrycz, "An overview and experimental study of learning-based optimization algorithms for the vehicle routing problem," *IEEE/CAA Journal of Automatica Sinica*, vol. 9, no. 7, pp. 1115–1138, 2022.
[55] Y. Bengio, A. Lodi, and A. Prouvost, "Machine learning for combinatorial optimization: a methodological tour d'horizon," *European Journal of Operational Research*, vol. 290, no. 2, pp. 405–421, 2021.
[56] X. Liv, *Combining genetic algorithm and case-based reasoning for structure design*. University of Nevada, Reno, 1996.
[57] E. K. Burke, S. Petrovic, and R. Qu, "Case-based heuristic selection for timetabling problems," *Journal of Scheduling*, vol. 9, no. 2, pp. 115–132, 2006.
[58] Y. Yamamoto, T. Kawabe, Y. Kobayashi, S. Tsuruta, Y. Sakurai, and R. Knauf, "A refined case based genetic algorithm for intelligent route optimization," in *2015 11th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*. IEEE, 2015, pp. 698–704.
[59] S. Oman and P. Cunningham, "Using case retrieval to seed genetic algorithms," *International Journal of Computational Intelligence and Applications*, vol. 1, no. 01, pp. 71–82, 2001.
[60] M. P. Deisenroth, A. A. Faisal, and C. S. Ong, *Mathematics for machine learning*. Cambridge University Press, 2020.
[61] A. W. Bowman and A. Azzalini, *Applied smoothing techniques for data analysis: the kernel approach with S-Plus illustrations*. OUP Oxford, 1997, vol. 18.
[62] R. Descartes and É. Gilson, *Discours de la méthode*. Vrin, 1987.
[63] M. N. Omidvar, X. Li, and K. Tang, "Designing benchmark problems for large-scale continuous optimization," *Information Sciences*, vol. 316, pp. 419–436, 2015.
[64] J. Derrac, S. García, D. Molina, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 3–18, 2011.