Sequential Transfer via Clustering-Based Similarity Measurement for Faster Trajectory Optimization

1st Wu Lin
Department of Computing
The Hong Kong Polytechnic University
Hong Kong SAR, China
linwu.lin@connect.polyu.hk

2nd Qiuzhen Lin
College of Computer Science and Software Engineering
Shenzhen University
Shenzhen, China
qiuzhlin@szu.edu.cn

3rd Xiaoming Xue
Department of Computer Science
City University of Hong Kong
Hong Kong SAR, China
xming.hsueh@my.cityu.edu.hk

4th Kay Chen Tan
Department of Computing
The Hong Kong Polytechnic University
Hong Kong SAR, China
kctan@polyu.edu.hk

***Abstract*—In evolutionary sequential transfer optimization (ESTO), the precision of similarity measurements between source and target tasks is crucial in selecting useful source solutions to accelerate the optimization of the target task. Despite the success of existing methods in optimizing various numerical functions, their capability to accurately measure the similarity among practical trajectory optimization problems (TOPs) remains limited due to the complex nature of real-world scenarios. To alleviate their limitations, this paper proposes a clustering-based similarity measurement method for ESTO, aiming to accurately select promising source solutions to speed up the evolutionary search in solving TOPs. Specifically, all candidate source TOPs that have been optimized are first categorized into different clusters based on their populations via *k*-means clustering. Afterward, the populations of source TOPs in the same cluster are employed to compute the similarity of these source TOPs to the target TOP that is being optimized. In this way, the TOPs with the highest similarity can be identified and their optimized solutions will be injected into the population of the target task for faster convergence. The experimental results demonstrate that our proposed method outperforms existing approaches, thereby confirming its capability to improve the optimization performance and efficiency in solving practical TOPs.**

***Keywords*—Evolutionary sequential transfer optimization, similarity measurement, trajectory optimization problems.**

I. Introduction

Evolutionary algorithms (EAs) are a class of population-based heuristic algorithms that simulate the process of natural evolution [1]-[3]. Due to its simplicity and practicality, EAs have gained widespread applications in a variety of real-world scenarios [4], [5]. In general, one canonical EA usually solves an optimization problem at one execution by starting with an initial population that is randomly sampled from the search space. However, problems do not usually exist in isolation [6]-[9]. As a large number of problems (i.e., source tasks) have been solved, the search experience, including entire algorithm [10], configured parameter [11], and evaluated solution [12], can be collected as available knowledge. After that, these accumulated abundant knowledge can be effectively utilized to speed up the optimization process of a new problem (called the target task). Therefore, evolutionary sequential transfer optimization (ESTO), which integrates knowledge transfer (KT) into EA, is proposed as an emerging search paradigm [13], [14]. By transferring knowledge from source tasks to the target task, ESTO aims to achieve faster evolutionary search.

Due to the ease of implementation and availability, various solution-based KT methods have been developed to transfer solutions from the population of the source task (called the source population) to the population of the target task (called the target population) [15]-[17]. In the literature, existing KT methods can be categorized into two categories: 1) domain adaptation-based approaches and 2) similarity measurement-based approaches. The domain adaptation-based approaches aim to adapt the solutions of the source population via the learned mapping from the source task to the target task [18]-[19]. For example, through using the source population as the input and the target population as the output, the denoising autoencoder (AE) [18] was proposed to learn the mapping by minimizing the reconstruction loss of the corrupted input on the source task. Furthermore, to capture the nonlinearity between source and target populations, kernelized AE (KAE) [19] was developed to learn the mapping in a reproduced kernel Hilbert space via a kernel function. However, learning the mapping will result in extra computational costs, thereby diminishing the effectiveness of knowledge transfer [20].

To perform more effective transfer, some similarity measurement-based approaches have been developed to select promising solutions from source tasks as transferred solutions [14]. In particular, several distance metrics are employed to measure the similarity between source and target tasks by computing the distance of their populations [21]-[24]. For example, the Euclidean distance (ED) [21], the Wasserstein distance (WD) [22], the Kullback-Leibler divergence (KLD) [23], and the maximum mean discrepancy (MMD) [24] have been used for similarity measurements in solving several artificial numerical functions [14], [25]. However, in some practical trajectory optimization problems (TOPs), exactly measuring similarity between them using the above similarity measurements becomes challenging due to the complex nature of real-world scenarios [26]-[30].

To alleviate the limitations of existing methods, this paper proposes a novel clustering-based similarity measurement method (CSM) for ESTO when tackling practical TOPs. Our proposed method first classifies all source tasks (i.e., the TOPs that have been solved) into different clusters by *k*-means clustering [31]. After that, for each source TOP, the cluster to which it belongs is identified and then the populations of all

979-8-3503-5409-6/24/$31.00 ©2024 IEEE
DOI 10.1109/CAI59869.2024.00229

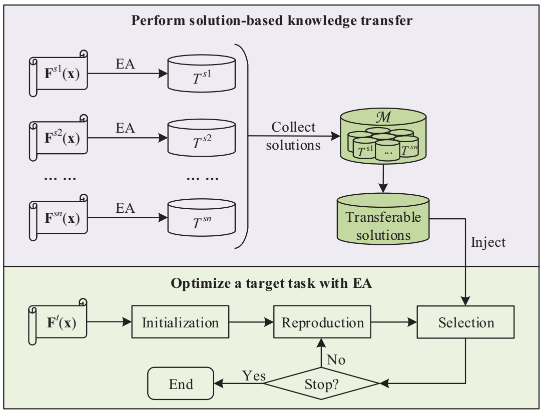

Perform solution-based knowledge transfer
$\mathbf{F}^{s1}(\mathbf{x})$
EA
T^{s1}
$\mathbf{F}^{s2}(\mathbf{x})$
EA
T^{s2}
… …
… …
$\mathbf{F}^{sn}(\mathbf{x})$
EA
T^{sn}
Collect solutions
$\mathcal{M}$
T^{s1} … T^{sn}
Transferable solutions
Inject
Optimize a target task with EA
$\mathbf{F}^{t}(\mathbf{x})$
Initialization
Reproduction
Selection
No
Stop?
Yes
End

Fig. 1. Flowchart of a general paradigm of solution-based ESTO.

source TOPs in this cluster are employed to measure its similarity to the target task (i.e., the TOP that is currently being optimized). In this way, the source TOPs exhibiting the highest similarity are more likely to possess the trajectories that closely approximate the optimal trajectory of the target TOP. Thus, the promising source solutions can be accurately selected and they are injected into the target population to accelerate the optimization of the target TOP. By embedding CSM into ESTO paradigm, a new ESTO algorithm, i.e., ESTOA-CSM, is implemented. The experimental results on ten test problems validate the effectiveness of our proposed method in accelerating the evolutionary search when solving practical TOPs.

The rest of this paper is organized as follows. Section II provides a concise introduction of ESTO and then discusses related work. Section III presents the details of our proposed methodology. The experimental results and brief discussions are provided in Section IV. Finally, Section V concludes this paper and suggests our future work.

II. Preliminaries

A. Evolutionary Sequential Transfer Optimization

ESTO can utilize the search experience of source tasks to accelerate the optimization of the target task [14]. In general, a sequential transfer optimization problem (STOP) can be formulated by

$$\min_{\mathbf{x}\in\Omega} \; [\; \mathbf{F}(\mathbf{x}) \mid \mathcal{M} \;] \tag{1}$$

where $\mathbf{x} = \{x_1, ..., x_d\}$ is the d-dimensional decision variable in the search space Ω, $\mathbf{F}(\mathbf{x}) = f_1(\mathbf{x}), ..., f_m(\mathbf{x})$ is the vector including m objective functions of the target task, and $\mathcal{M}$ is the knowledge base that contains the available information of previously-solved source tasks. In particular, in solution-based ESTO, knowledge transfer is implemented in the form of solutions [14]. Therefore, $\mathcal{M}$ can be formed by collecting the evaluated solutions of n source tasks during their respective evolutionary search processes with the maximal generations $G_{\max}$, which can be represented by

$$\mathcal{M} = \left\{ T^{s1},\, T^{s2}, ...,\, T^{sn} \right\} \tag{2}$$

where $T^{si} = \{\mathbf{P}^{ij}, \mathbf{F}^{ij}, j = 1, ..., G_{\max}\}$ denotes the population data of the ith source task ($i \in \{1,\; ...,\; n\}$), $\mathbf{P}^{ij}$ is the solution set at the jth generation, and $\mathbf{F}^{ij}$ is the set of their objective values.

To achieve faster evolutionary search in solving STOPs, the solution-based ESTO that integrates solution-based KT into EA is developed as an emerging search paradigm. During the evolutionary search processes of source tasks, their solutions can be collected and then saved into the knowledge base. Afterward, these available solutions can be effectively utilized as transferable solutions to accelerate the optimization process of the target task. To be clear, Fig. 1 shows the flowchart of a general paradigm of solution-based ESTO. In particular, a canonical EA including population initialization, offspring reproduction, and environmental selection is employed as the basic optimizer for the target task. After a large number of source tasks have been sequentially solved by EA, the knowledge base $\mathcal{M}$ is formed by collecting their solutions. Consequently, when using EA to optimize one target task, the transferable solutions are selected from $\mathcal{M}$ and injected into the target population, thereby accelerating the optimization process of the target task.

B. Existing Similarity Measurement Methods

In the literature, several similarity measurement-based KT methods have been proposed to select promising solutions from candidate source tasks to the target task [21]-[24]. These methods measure the similarity between the source and target tasks and then select transferable solutions based on their measured values of similarity. Recently, various commonly used distance metrics (e.g., ED, WD, KLD, and MMD) have been employed to measure the similarity between the source and target tasks by computing the distance between their populations. Particularly, ED [21] measures the similarity between the source and target tasks by computing the distance between the mean vectors of their populations (i.e., $\mathbf{P}_s$ and $\mathbf{P}_t$), which is given by

$$\mathrm{ED}(\mathbf{P}_s,\, \mathbf{P}_t) = \left\| \boldsymbol{\mu}_s - \boldsymbol{\mu}_t \right\|_2 \tag{3}$$

where $\boldsymbol{\mu}_s$ and $\boldsymbol{\mu}_t$ represent the mean vectors of $\mathbf{P}_s$ and $\mathbf{P}_t$, respectively. In addition, WD [22] considers both the distances between the mean vectors and standard deviation vectors of the populations of the source and target tasks, which is computed by

$$\mathrm{WD}(\mathbf{P}_s,\, \mathbf{P}_t) = \sqrt{\left\| \boldsymbol{\mu}_s - \boldsymbol{\mu}_t \right\|_2^2 + \left\| \boldsymbol{\sigma}_s - \boldsymbol{\sigma}_t \right\|_2^2} \tag{4}$$

where $\boldsymbol{\sigma}_s$ and $\boldsymbol{\sigma}_t$ are the standard deviation vectors of $\mathbf{P}_s$ and $\mathbf{P}_t$, respectively. Besides, KLD [23] computes the distance of the source and target populations based on their Gaussian representations, which is given by

$$\begin{aligned}\mathrm{KLD}(\mathbf{P}_s,\, \mathbf{P}_t) = \frac{1}{2}\Big(&\mathrm{tr}\left(\boldsymbol{\Sigma}_t^{-1}\boldsymbol{\Sigma}_s\right) + \left(\boldsymbol{\mu}_t - \boldsymbol{\mu}_s\right)^{\mathrm{T}} \Sigma_t^{-1} \left(\boldsymbol{\mu}_t - \boldsymbol{\mu}_s\right) \\ &- d + \ln\left(\det \boldsymbol{\Sigma}_t / \det \boldsymbol{\Sigma}_s\right)\Big)\end{aligned} \tag{5}$$

where $\boldsymbol{\Sigma}_s$ and $\boldsymbol{\Sigma}_t$ are the covariance matrices of $\mathbf{P}_s$ and $\mathbf{P}_t$, d is the smaller dimension of $\mathbf{P}_s$ and $\mathbf{P}_t$, tr(•) is the trace of a matrix, and det(•) is the determinant of a matrix. Moreover, MMD [24] computes the distance between the source and target populations by transforming their solutions into the reproducing kernel Hilbert space, which is given by

$$\begin{aligned}\mathrm{MMD}(\mathbf{P}_s,\, \mathbf{P}_t) = &\frac{1}{n_s^2 - n_s}\sum_{i\neq j}^{n_s} k(\mathbf{x}_i^s,\, \mathbf{x}_j^s) - \frac{2}{n_s n_t}\sum_{i,\, j=1}^{n_s,\, n_t} k(\mathbf{x}_i^s,\, \mathbf{x}_j^t) \\ &+ \frac{1}{n_t^2 - n_t}\sum_{i\neq j}^{n_t} k(\mathbf{x}_i^t,\, \mathbf{x}_j^t)\end{aligned} \tag{6}$$

```
Algorithm 1 Clustering-Based Similarity Measurement (CSM)
Input: P, $\mathcal{M}$, $K$, $g$
Output: $S_1, \ldots, S_{|\mathcal{M}|}$
1  Set V to an empty set
2  for each source task $T^i$ in $\mathcal{M}$
3      $\mathbf{P}^{ig}$ ← Collect the $g$th population of $T^i$
4      $\bar{\mathbf{x}}^i$ ← Calculate the mean vector by (7)
5      $\mathbf{V} = \mathbf{V} \cup \{\bar{\mathbf{x}}^i\}$
6  end
   // divide all source tasks into $K$ clusters
7  $(C_1, \ldots, C_K)$ ← $k$-means clustering $(\mathbf{V}, K)$
8  for $i = 1 : |\mathcal{M}|$
9      $C_a$ ← Find the cluster to which $T^i$ belongs
10     U ← Collect the $g$th populations of all source tasks in $C_a$
11     $S_i$ ← Calculate the similarity of $T^{si}$ to $T^t$ by (8)
12 end
13 return $S_1, \ldots, S_{|\mathcal{M}|}$
```

where n_s and n_t are the numbers of solutions in $\mathbf{P}_s$ and $\mathbf{P}_t$, respectively. In addition, $\mathbf{x}_i^s$ and $\mathbf{x}_i^t$ are the ith solutions in $\mathbf{P}_s$ and $\mathbf{P}_t$, respectively. Here, the Gaussian kernel function with the width parameter $\sigma = 0.5$ is employed in MMD, i.e., $k(\mathbf{x}, \mathbf{x}') = \exp(-\|\mathbf{x}-\mathbf{x}'\|_2^2/(2\sigma^2))$.

III. Methodology

This section presents the details of our proposed method. In particular, the specific details of our proposed CSM are first introduced in subsection II.A. Then, the implementation of embedding CSM into ESTO paradigm is provided in subsection II.B.

A. Clustering-Based Similarity Measurement

The pseudocode of CSM is given in **Algorithm 1** with the inputs: **P** (the population of the target task T_t), $\mathcal{M}$ (the knowledge base including the solutions of source tasks), K (the number of clusters), g (the current generation). In line 1, **V** is set to an empty set, which is employed to collect the mean vectors of all source tasks. In particular, as shown in lines 2-6, for each source task T^i, its population at current generation g, as denoted by $\mathbf{P}^{ig}$, is used to compute the mean vector $\bar{\mathbf{x}}^i = \{\bar{x}_1^i, \ldots, \bar{x}_d^i\}$ by

$$\bar{x}_j^i = \frac{\sum_{n=1}^{N} x_{nj}}{N} \tag{7}$$

where N is the number of solutions in $\mathbf{P}^{ig}$ and x_{nj} is the value of the nth solution on the jth dimension ($j \in \{1, \ldots, d\}$). After visiting all source tasks, their mean vectors are collected into **V**. Then, these source tasks are divided into K clusters (i.e., $C_1, \ldots, C_K$) by using k-means clustering [31] with **V** and K as the inputs in line 7. Next, the similarity of each source task to the target task is computed as shown in lines 8-12. The cluster to which the ith source task T^{si} belongs is first identified, which is denoted by C_a. Then, the gth populations of all source tasks in C_a are collected to form the combined population **U**. After that, the similarity of T^{si} to T^t, which is denoted by S_i, is measured by computing the distance between **P** and **U** as follows:

$$S_i = \mathrm{WD}(\mathbf{P}, \mathbf{U}) = \sqrt{\|\boldsymbol{\mu}_{\mathbf{P}} - \boldsymbol{\mu}_{\mathbf{U}}\|_2^2 + \|\boldsymbol{\sigma}_{\mathbf{P}} - \boldsymbol{\sigma}_{\mathbf{U}}\|_2^2} \tag{8}$$

where $\boldsymbol{\mu}_{\mathbf{P}}$ and $\boldsymbol{\mu}_{\mathbf{U}}$ denote the mean vectors of **P** and **U**, respectively. Besides, $\boldsymbol{\sigma}_{\mathbf{P}}$ and $\boldsymbol{\sigma}_{\mathbf{U}}$ are the standard deviation vectors of **P** and **U**, respectively. Finally, the measured similarity values of all source tasks in $\mathcal{M}$, i.e., $S_1, \ldots, S_{|\mathcal{M}|}$, are returned as the output in line 13.

```
Algorithm 2 The Main Framework
Input: An STOP with $T^t$ and $\mathcal{M}$, $N$, $TG$, $c$, $K$, $G_{max}$.
Output: P
1  initialize P with $N$ solutions
2  set $g = 1$
3  while $g \le G_{\max}$
4      if mod ($g$, $TG$) == 0 // Perform knowledge transfer
5          $(S_1, \ldots, S_{|\mathcal{M}|})$ ← CSM ($\mathbf{P}, \mathcal{M}, K, g$) // Algorithm 1
6          $\boldsymbol{Q}$ ← Find the most promising source tasks by (9)
7          TS ← get $c$ source solution(s) of one source task in $\boldsymbol{Q}$
8      else
9          TS = Ø
10     end
11     O ← Crossover and Mutation on P
12     P ← Environmental Selection on $\mathbf{P} \cup \mathbf{O} \cup \mathbf{TS}$
13     $g = g + 1$
14 end
15 return P
```

B. Main Framework

To be clear, the pseudocode of the main framework of embedding CSM into ESTO paradigm is given in **Algorithm 2** with the inputs: an STOP with T^t (the target task) and $\mathcal{M}$ (the knowledge base including solutions of source tasks), N (the population size), TG (the transfer interval), c (the number of transferable solutions), K (the number of clusters), and $G_{\max}$ (the maximal generations). As shown in lines 1-2, a population **P** is first initialized through randomly sampling N solutions in the search space of T_t, and the count of generation (i.e., g) is set to 1. Next, the main evolutionary search process is run in lines 3-14. In particular, as shown in lines 4-10, at each transfer interval, i.e., mod (g, TG) == 0, **Algorithm 1** is first performed to measure the similarity values of all candidate source tasks to T_t, i.e., $S_1, \ldots, S_{|\mathcal{M}|}$. Next, these promising source tasks with maximal similarity value are collected into $\boldsymbol{Q}$ as follows:

$$\boldsymbol{Q} = \left\{ T^{si} \mid i = \underset{i \in \{1, \ldots, |\mathcal{M}|\}}{\arg\max} \{S_i\} \right\}. \tag{9}$$

Then, the solution set, i.e., **TS**, can be obtained by collecting c optimized solutions of any one promising source task in $\boldsymbol{Q}$. Otherwise, **TS** will be an empty set. Then, in line 11, the crossover and mutation operators are sequentially executed to generate N-|**TS**| offspring, forming the offspring population **O**. Next, in line 12, the environmental selection is performed to select N solutions from the combination population of **P**, **O**, and **TS** based on their objective values, forming the next-generation population **P**. The count of generation, i.e., g, is incremented by 1 in line 13. The abovementioned procedures will be iteratively executed as long as the condition is satisfied (i.e., $g \le G_{\max}$). Otherwise, **P** will be returned as the final population in line 15.

IV. Numerical Experiments

A. Test Problems

Generally, TOPs involve finding an optimal trajectory or path for a system, such as a robot or a vehicle, to navigate within a given environment while satisfying some specific constraints and objectives [26]. Indeed, TOPs are pervasive and widely applicable across various domains, including robotics, aerospace, autonomous vehicles, and motion planning [27]-[30]. In the

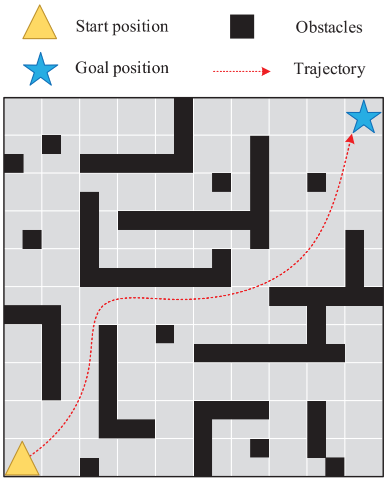


Fig. 2. A toy example of the trajectory optimization problem.

TABLE I

PARAMETER SETTINGS FOR STOP1 TO STOP10

Problem	Number of obstacles in target task (m_t)	Number of source tasks (n)	Number of obstacles in source task (m_s)
STOP1	$m_t = 10$	$n = 100$	$m_s \in [10,\ 30]$
STOP2	$m_t = 15$		
STOP3	$m_t = 20$		
STOP4	$m_t = 25$		
STOP5	$m_t = 30$		
STOP6	$m_t = 10$	$n = 200$	
STOP7	$m_t = 15$		
STOP8	$m_t = 20$		
STOP9	$m_t = 25$		
STOP10	$m_t = 30$		

context of robot navigation and obstacle avoidance, TOPs aim to find the optimal trajectory from a start point to a goal point in a complex environment while effectively avoiding obstacles [32]-[34]. To provide a more intuitive understanding, Fig. 2 presents a toy example that visually represents a typical TOP. As shown in Fig. 2, the environment is denoted by the 2-D map where the squares marked by blank color are the obstacles while a red dashed line with an arrow denotes one candidate trajectory from the start position to the goal position. To establish the mathematical formulation of the typical TOP, BSpline is employed to fit the trajectory by using d sample points (i.e., $\mathbf{p}_1, \ldots, \mathbf{p}_d$) in a 2-D map that is normalized in $[0, 1]^2$. For each sample point, i.e., $\mathbf{p}_i = \{x_i, y_i\}$, x_i is set to i/d, which can make all sample points have equal intervals on the x-axis. With these definitions, the typical TOP can be formulated as a d-dimensional optimization problem, which is given by

$$\arg\min_{\{y_1,\ldots,y_d\}} \{c_d(\mathbf{p}_1,\ \ldots,\ \mathbf{p}_d) + c_c(\mathbf{p}_1,\ \ldots,\ \mathbf{p}_d) + \lambda(\| \mathbf{p}_1 - \mathbf{s} \|_2 + \| \mathbf{p}_d - \mathbf{g} \|_2) + b\} \tag{10}$$

where $\{y_1, \ldots, y_d\}$ is the d-dimensional decision variable in the search space $[0, 1]^d$. As shown in (10), $c_d(\mathbf{p}_1, \ldots, \mathbf{p}_d)$ and $c_c(\mathbf{p}_1, \ldots, \mathbf{p}_d)$ are the distance cost and collision cost of the fitted trajectory with $\mathbf{p}_1, \ldots, \mathbf{p}_d$, respectively. Besides, the distance costs of the fitted trajectory to the start position and goal position (i.e., $\mathbf{s} = \{0,\ 0\}$ and $\mathbf{g} = \{1,\ 1\}$) are computed by $\| \mathbf{p}_1 - \mathbf{s} \|_2$ and $\| \mathbf{p}_d - \mathbf{g} \|_2$. As suggested in [34], the size of each obstacle is 0.5×0.5, the penalty cost of every collision is set to 20, and λ and b are set to 10 and 5, respectively.

In practical scenarios, the environments frequently exhibit variations, which result in changes in the optimal trajectory for different TOPs. With a large number of previously-solved TOPs, the collected search experience can be effectively utilized to accelerate the optimization process when solving a new TOP. Therefore, ten representative STOPs are constructed through using practical TOPs as the source and target tasks. In each STOP, one TOP is used as the target task while n other TOPs are regarded as source tasks. Note that one canonical EA is used to solve these source tasks and all evaluated solutions are collected to form the knowledge base. Both the numbers of obstacles for source and target tasks, i.e., m_s and m_t, are integers that are limited in the range [10, 30]. All TOPs are generated by randomly placing some obstacles on their 2-D maps. In this way, the total number of obstacles and the position of each obstacle vary in the 2-D map of each TOP, thereby contributing to the diversity of generated source tasks. The detailed parameter settings of ten test problems (i.e., STOP1 to STOP10) are provided in Table I.

B. Compared Algorithms and Parameter Settings

In the experiments, a canonical EA and four ESTOAs equipped with distance metrics (i.e., ESTOA-ED, ESTOA-WD, ESTOA-KLD, and ESTOA-MMD) are used as the compared algorithms. In the four ESTOAs, the corresponding distance metrics, i.e., ED, WD, KLD, and MMD, are used to measure the similarity of the target task to each source task based on their populations, respectively. To have fair performance comparison, c available best solutions of the most similar source task are selected as transferable solutions at each transferable generation.

The common parameters of all compared algorithms are kept consistent. The population size (N) and the maximum number of generations ($G_{\max}$) are set to 50 and 100, respectively. In addition, the simulated binary crossover (SBX) [35] with the probability $p_c = 1$ and the distribution index $\eta_c = 15$ and the polynomial-based mutation (PM) [36] with the probability $p_m = 1/d$ and the distribution index $\eta_m = 15$ are employed as the crossover and mutation operators to generate offspring population. For all ESTOAs, the transfer generation interval (TG) and the number of transferred solutions (c) are set to 1 and 1, respectively. For ESTOA-CSM, the number of clusters (K) is set to 10. Note that each algorithm is independently executed 50 times on each test problem and the numerical results are collected for performance comparison.

C. Comparison Results

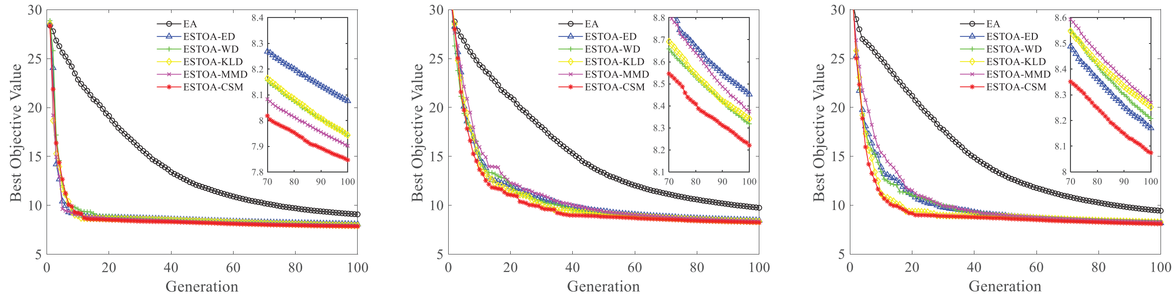
Table II presents the detailed numerical results of all compared algorithms, while the summarized comparison results on all test problems are provided in the last row. It can be observed from Table II, ESTOA-CSM can achieve better performance on all test problems when compared to EA. The comparison results demonstrate the effectiveness of ESTO in accelerating the optimization process of practical TOPs. In

TABLE II

Mean Objective Values and Standard Deviations Obtained by All Compared Algorithms

Problem		EA	ESTOA-ED	ESTOA-WD	ESTOA-KLD	ESTOA-MMD	ESTOA-CSM
STOP1	mean	9.08e+00(-)	8.08e+00(-)	7.94e+00(-)	7.94e+00(-)	7.90e+00(-)	**7.85e+00**
	std	6.16e-01	4.32e-01	3.61e-01	3.47e-01	2.87e-01	**3.07e-01**
STOP2	mean	9.02e+00(-)	8.03e+00(-)	**7.80e+00(+)**	7.85e+00(-)	7.84e+00(-)	7.81e+00
	std	5.88e-01	3.13e-01	**2.76e-01**	3.01e-01	2.55e-01	2.92e-01
STOP3	mean	9.74e+00(-)	8.45e+00(-)	8.32e+00(-)	8.34e+00(-)	8.37e+00(-)	**8.22e+00**
	std	6.56e-01	5.16e-01	5.30e-01	4.29e-01	3.38e-01	**3.17e-01**
STOP4	mean	9.27e+00(-)	8.31e+00(-)	8.23e+00(-)	8.06e+00(-)	8.15e+00(-)	**8.04e+00**
	std	5.92e-01	4.87e-01	5.05e-01	5.21e-01	4.43e-01	**4.22e-01**
STOP5	mean	9.92e+00(-)	8.30e+00(-)	8.33e+00(-)	**8.11e+00(+)**	8.23e+00(-)	8.22e+00
	std	6.81e-01	4.10e-01	3.83e-01	**3.01e-01**	4.55e-01	3.40e-01
STOP6	mean	8.85e+00(-)	8.02e+00(-)	8.01e+00(-)	7.99e+00(-)	**7.88e+00(+)**	7.91e+00
	std	5.18e-01	4.24e-01	5.35e-01	3.86e-01	**4.18e-01**	3.06e-01
STOP7	mean	9.11e+00(-)	7.99e+00(-)	7.99e+00(-)	8.09e+00(-)	7.99e+00(-)	**7.97e+00**
	std	4.83e-01	3.23e-01	2.65e-01	3.26e-01	3.42e-01	**2.68e-01**
STOP8	mean	9.42e+00(-)	8.17e+00(-)	8.21e+00(-)	8.25e+00(-)	8.27e+00(-)	**8.07e+00**
	std	5.80e-01	3.99e-01	3.52e-01	2.53e-01	3.66e-01	**2.57e-01**
STOP9	mean	9.72e+00(-)	7.84e+00(+)	**7.74e+00(+)**	7.86e+00(-)	7.75e+00(+)	7.85e+00
	std	7.22e-01	3.94e-01	**2.91e-01**	2.90e-01	4.11e-01	2.49e-01
STOP10	mean	1.05e+01(-)	8.59e+00(-)	8.59e+00(-)	8.58e+00(-)	**8.42e+00(+)**	8.45e+00
	std	6.84e-01	8.61e-01	9.12e-01	6.14e-01	**7.38e-01**	5.90e-01
+/-/=		0/10/0	1/9/0	2/8/0	1/9/0	3/7/0	\

"+", "-", and "=" indicate that the results of the compared algorithm on test problems are better than, worse than, and similar to that of ESTOA-CSM, respectively. The best result on each test problem is highlighted in bold.



(a) (b) (c)

Fig. 3. Convergence curves of all compared algorithms on (a) STOP1, (b) STOP3, and (c) STOP8.

addition, ESTOA-CSM outperforms ESTOA-ED, ESTOA-WD, ESTOA-KLD, and ESTOA-MMD in solving most test problems (i.e., 9, 8, 9, and 7 out of 10 cases, respectively). The above comparisons highlight the performance superiority of ESTOA-CSM when compared with other existing ESTOAs, thereby validating its effectiveness in accurately measuring the similarity of source and target tasks when handling practical TOPs.

Additionally, the convergence curves of all compared algorithms on STOP1, STOP3, and STOP8 are provided in Figs. 3(a), (b), and (c), respectively. It can be observed in Fig. 3 that all ESTOAs are capable of significantly accelerating the evolutionary search processes in solving STOP1, STOP3, and STOP8 when compared to EA. Furthermore, ESTOA-CSM significantly surpasses ESTOA-ED, ESTOA-WD, ESTOA-KLD, and ESTOA-MMD on the three STOPs in terms of both convergence speed and the quality of the final population. The above observations demonstrate the effectiveness of ESTOA-CSM in measuring the similarity of the source and target tasks, thereby further improving the optimization efficiency and performance when solving practical TOPs. In summary, the above comparison results reveal the capability of our proposed ESTOA-CSM in achieving faster evolutionary search when handling practical TOPs.

V. Conclusion

This paper has proposed a novel clustering-based similarity measurement method for ESTO, which is capable of accurately assessing the similarity of the source and target tasks in practical TOPs. Compared with existing methods, our proposed method employs *k*-means clustering to classify source tasks into different clusters. The similar source tasks are more likely to be in the same cluster. Instead of employing the population of each source task to estimate its similarity to the target task, our proposed method measures the similarity based on the populations of the source tasks in the same cluster, thereby further improving the similarity measurement precision. The experimental results on ten test problems have validated the effectiveness of our method in achieving faster evolutionary search in solving practical TOPs.

In our future work, we will extend our proposed method to solve various practical optimization problems with more complex characteristics. Furthermore, we will continue to refine our similarity measurement method, which aims to effectively select promising source solutions to accelerate the evolutionary search in solving real-world problems.

Acknowledgments

This work was supported by the Research Grants Council of the Hong Kong SAR (Grant No. PolyU11211521, PolyU15218622, PolyU15215623, PolyU25216423), The Hong Kong Polytechnic University (Project IDs: P0039734, P0035379, P0043563, and P0046094), and the National Natural Science Foundation of China (Grant No. U21A20512, 62306259, and 62173236).

References

[1] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*. New York, NY, USA: Wiley, 2001.

[2] W. Lin, Q. Lin, Z. Zhu, J. Li, J. Chen, and Z. Ming, "Evolutionary Search with Multiple Utopian Reference Points in Decomposition-Based Multiobjective Optimization," *Complexity*, vol. 2019, Article ID 7436712, 22 pages, 2019. https://doi.org/10.1155/2019/7436712.

[3] S. Liu, Q. Lin, J. Li, and K. C. Tan, "A Survey on Learnable Evolutionary Algorithms for Scalable Multiobjective Optimization," *IEEE Trans. Evol. Comput.*, vol. 27, no. 6, pp. 1941-1961, Dec. 2023.

[4] X. Qiu, J. -X. Xu, Y. Xu, and K. C. Tan, "A New Differential Evolution Algorithm for Minimax Optimization in Robust Design," *IEEE Trans. Cybern.*, vol. 48, no. 5, pp. 1355-1368, May 2018.

[5] A. Slowik and H. Kwasnicka, "Evolutionary Algorithms and Their Applications to Engineering Problems," *Neural. Comput. Appl.*, vol. 32, no. 16, pp. 12363–12379, Aug. 2020.

[6] K. C. Tan, L. Feng, and M. Jiang, "Evolutionary Transfer Optimization - A New Frontier in Evolutionary Computation Research," *IEEE Comput. Intell. Mag.*, vol. 16, no. 1, pp. 22-33, Feb. 2021.

[7] L. Bai, W. Lin, A. Gupta, and Y. -S. Ong, "From Multitask Gradient Descent to Gradient-Free Evolutionary Multitasking: A Proof of Faster Convergence," *IEEE Trans. Cybern.*, vol. 52, no. 8, pp. 8561-8573, Aug. 2022.

[8] Y. Huang, L. Feng, M. Li, Y. Wang, Z. Zhu, and K. C. Tan, "Fast Vehicle Routing via Knowledge Transfer in a Reproducing Kernel Hilbert Space," *IEEE Trans. Syst., Man, Cybern.: Syst.*, vol. 53, no. 9, pp. 5404-5416, Sept. 2023.

[9] Z. Wang, L. Cao, L. Feng, M. Jiang, and K. C. Tan, "Evolutionary Multitask Optimization With Lower Confidence Bound-Based Solution Selection Strategy," *IEEE Trans. Evol. Comput.*, Jan. 2024. doi: 10.1109/TEVC.2023.3349250.

[10] 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," *Inf. Sci.*, vol. 317, pp. 224–245, Oct. 2015.

[11] H. H. Hoos, "Automated Algorithm Configuration and Parameter Tuning," *Auton. Sear.*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 37–71.

[12] P. Cunningham and B. Smyth, "Case-based Reasoning in Scheduling: Reusing Solution Components," *Int. J. Prod. Res.*, vol. 35, no. 11, pp. 2947–2962, Nov. 1997.

[13] 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 Trans. Evol. Comput.*, vol. 26, no. 6, pp. 1424–1438, Dec. 2022.

[14] X. Xue, C. Yang, L. Feng, K. Zhang, L. Song, and K. C. Tan, "Solution Transfer in Evolutionary Optimization: An Empirical Study on Sequential Transfer," *IEEE Trans. Evol. Comput.*, Dec., 2023. doi: 10.1109/ TEVC.2023.3339506.

[15] Gupta, Y.-S. Ong, and L. Feng, "Insights on Transfer Optimization: Because Experience is the Best Teacher," *IEEE Trans. Emerg. Top. Comput. Intell.*, vol. 2, no. 1, pp. 51–64, Feb. 2018.

[16] S. Liu, Q. Lin, L. Feng, K.-C. Wong, and K. C. Tan, "Evolutionary Multitasking for Large-Scale Multiobjective Optimization," *IEEE Trans. Evol. Comput.*, vol. 27, no. 4, pp. 863-877, Aug. 2023.

[17] Y. Huang, W. Zhou, Y. Wang, M. Li, L. Feng, and K. C. Tan, "Evolutionary Multitasking With Centralized Learning for Large-Scale Combinatorial Multi-Objective Optimization," *IEEE Trans. Evol. Comput.*, doi:10.1109/TEVC.2023.3323877.

[18] 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.

[19] L. Zhou, L. Feng, A. Gupta, and Y.-S. Ong, "Learnable Evolutionary Search across Heterogeneous Problems via Kernelized Autoencoding," *IEEE Trans. Evol. Comput.*, vol. 25, no. 3, pp. 567–581, Jun. 2021.

[20] W. Lin, Q. Lin, L. Feng, and K. C. Tan, "Ensemble of Domain Adaptation-Based Knowledge Transfer for Evolutionary Multitasking," *IEEE Trans. Evol. Comput.*, vol. 28, no. 2, pp. 388–402, Apr. 2024.

[21] L. Zhou, L. Feng, J. Zhong, Z. Zhu, B. Da, and Z. Wu, "A Study of Similarity Measure between Tasks for Multifactorial Evolutionary Algorithm," in *Proc. of Gene. Evol. Comput. Confer. Compan.*, Jul. 2018, pp. 229–230.

[22] J. Zhang, W. Zhou, X. Chen, W. Yao, and L. Cao, "Multi-Source Selective Transfer Framework in Multi-Objective Optimization Problems," *IEEE Trans. Evol. Comput.*, vol. 24, no. 3, pp. 424–438, Jun. 2020.

[23] Y. Chen, J. Zhong, L. Feng, and J. Zhang, "An Adaptive Archive-Based Evolutionary Framework for Many-Task Optimization," *IEEE Trans. Emerg. Top. Comput. Intell.*, vol. 4, no. 3, pp. 369–384, Jun. 2020.

[24] Z. Liang, X. Xu, L. Liu, Y. Tu, and Z. Zhu, "Evolutionary Many-Task Optimization Based on Multisource Knowledge Transfer," *IEEE Trans. Evol. Comput.*, vol. 26, no. 2, pp. 319–333, Apr. 2022.

[25] X. Xue, C. Yang, L. Feng, K. Zhang, L. Song, and K. C. Tan, "A Scalable Problem Generator for Sequential Transfer Optimization," arXiv preprint. Oct. 2023. https://doi.org/10.48550/arXiv.2304.08

[26] A. V. Rao, "Trajectory Optimization: A Survey," *Optimiz. Optim. Control in Autom. Syst.*, pp. 3-21, 2014.

[27] T. T. Mac, C. Copot, D. T. Tran, and R. De Keyser, "Heuristic approaches in robot path planning: A survey," *Rob. Auton. Syst.*, vol. 86, pp. 13–28, Dec. 2016.

[28] D. Malyuta, Y. Yu, P. Elango, and B. Açıkmeşe, "Advances in Trajectory Optimization for Space Vehicle Control," *Annu. Rev. Control.*, vol. 52, pp. 282-315, 2021.

[29] G. Huang, Y. Lu, and Y. Nan, "A Survey of Numerical Algorithms for Trajectory Optimization of Flight Vehicles," *Sci. China Technol. Sci.*, vol. 55, pp. 2538-2560, 2012.

[30] M. Kalakrishnan, S. Chitta, E. Theodorou, P. Pastor, and S. Schaal, "STOMP: Stochastic Trajectory Optimization for Motion Planning," in *Int. Confer. Robot. Autom.*, Shanghai, China, 2011, pp. 4569-4574.

[31] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, "An Efficient *K*-means Clustering Algorithm: Analysis and Implementation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 881-892, July 2002.

[32] Z. Wang, C. Gehring, P. Kohli, and S. Jegelka, "Batched Large-scale Bayesian Optimization in High-dimensional Spaces", in *Int. Confer. Artif. Intell. Stati.*, Mar. 2018. pp. 745-754.

[33] H. Yang, J. Qi, Y. Miao, H. Sun, and J. Li, "A New Robot Navigation Algorithm Based on a Double-Layer Ant Algorithm and Trajectory Optimization," *IEEE Trans. Ind. Electron.*, vol. 66, no. 11, pp. 8557–8566, Nov. 2019.

[34] J.-Y. Li, Z.-H. Zhan, K. C. Tan, and J. Zhang, "A Meta-Knowledge Transfer-Based Differential Evolution for Multitask Optimization," *IEEE Trans. Evol. Comput.*, vol. 26, no. 4, pp. 719–734, Aug. 2022.

[35] K. Deb and R. Agrawal, "Simulated Binary Crossover for Continuous Search Space," *Complex System*, vol. 9, no. 2, pp. 115-148, 1995.

[36] K. Deb and D. Deb, "Analyzing Mutation Schemes for Real-Parameter Genetic Algorithms," *Int. J. Artif. Intell. Soft Comput.*, vol. 4, no. 1, pp. 1-28, 2014.