Non-Linear Domain Adaptation in Transfer Evolutionary Optimization

Ray Lim[1,3] · Abhishek Gupta[2,3,4] · Yew-Soon Ong[1,2,3] · Liang Feng[5] · Allan N Zhang[3,4]

[1]School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798

[2]Data Science and Artificial Intelligence Research Centre, Nanyang Technological University, Singapore 639798

[3]Agency for Science, Technology and Research, Singapore 138632

[4]Singapore Institute of Manufacturing Technology, Singapore 138634

[5]College of Computer Science, Chongqing University, China 400044

Corresponding author: Yew-Soon Ong (e-mail: asysong@ntu.edu.sg; Tel: +65-6790-6448; Fax: +65-6792-6559)

Abstract

BACKGROUND: The cognitive ability to learn with experience is a hallmark of intelligent systems. The emerging *transfer optimization* paradigm pursues such human-like problem-solving prowess by leveraging useful information from various source tasks to enhance optimization efficiency on a related target task. The occurrence of harmful negative transfer is a key concern in this setting, paving the way for recent probabilistic model-based transfer evolutionary algorithms that curb this phenomenon. However, in applications where the source and target *domains*, i.e., the features of their respective search spaces (e.g., dimensionality) and the distribution of good solutions in those spaces, do not match, narrow focus on curbing negative effects can lead to the conservative cancellation of knowledge transfer.

METHOD: Taking this cue, this paper presents a novel perspective on *domain adaptation* in the context of evolutionary optimization, stimulating positive transfers even in scenarios of source-target domain mismatch. Our first contribution is to establish a probabilistic formulation of domain adaptation, by which source and/or target tasks can be mapped to a common representation space in which their discrepancy is reduced. Secondly, a domain adaptive transfer evolutionary algorithm is proposed, supporting both offline construction and online data-driven learning of *non-linear* mapping functions.

RESULTS: The performance of the algorithm is experimentally verified, demonstrating superior convergence rate in comparison to state-of-the-art baselines on synthetic benchmarks and a practical case study in multi-location inventory planning.

CONCLUSION: Our results thus shed light on a new research direction for optimization algorithms that improve their efficacy by learning from *heterogeneous* experiences.

Keywords: Domain adaptation · Probabilistic model-based search · Transfer evolutionary optimization· Memetic computation

Conflict of Interest: Ray Lim declares that he has no conflict of interest. Abhishek Gupta declares that he has no conflict of interest. Yew-Soon Ong declares that he has no conflict of interest. Liang Feng declares that he has no conflict of interest. Allan N Zhang declares that he has no conflict of interest.

1. Introduction

Evolutionary algorithms (EAs) are population-based search procedures that are known to be effective for tackling complex optimization problems with highly non-linear, non-differentiable, or even black-box objectives and constraint functions. Due to their generality and algorithmic simplicity, EAs have been widely used with much success in a plethora of real-world applications [1-3]. Despite these advantages, the largely problem-independent nature of their search operators implies that EAs may sometimes be slow to converge to optimal (or at least near-optimal) solutions. Thus, there has been increasing interest in the design of EAs that could take advantage of available knowledge relevant to the task at hand to boost search performance [4, 5]. To this end, *memetic computation* represents a family of cognitively-inspired algorithms in which *memes*, embodied as computationally encoded building-blocks of problem-solving knowledge (that reside in a machine's metaphorical *brain*), can be learned from one task and transferred to another [6]. Notably, this modern interpretation of memetic computation has led to a recent upsurge in so-called data-driven *transfer evolutionary optimization* (TrEO) algorithms [6, 7] that are equipped to automatically reuse information from various *source* tasks to improve optimization efficiency in a related *target* task of interest.

Although the development of TrEO is still in its nascent stages, some methodological advances have been made. On one hand, s*equential transfer* methods have catered to the reuse of knowledge acquired from *past* problem-solving experiences [7], with a range of success stories in neuro-evolution [6], combinatorial optimization [8-11], learning classifier systems [12, 13], etc. On the other hand, *evolutionary multitasking* is yet another family of methods that gives rise to the scope for mutual knowledge exchange among multiple optimization problems tackled in unison [14-18]. Research works investigating the real-world applicability of both these approaches are found in diverse areas spanning engineering process design [19, 20], intelligent logistics planning [21], robotics [22], and software engineering [23], to name a few.

We note that early realizations of TrEO were reliant on a practitioner's domain expertise for devising and incorporating a knowledge transfer mechanism into an EA; the typical assumption being that similar problems often give rise to similar solutions [24]. In cases where the transferred solution(s) turned out to be useless for the target task, the onus was placed on evolutionary selection to gradually sieve out the irrelevant information [7]. The extra effort spent in sieving naturally impedes the target search, a phenomenon that has come to be referred to as *negative transfer* in optimization. The need to curb negative transfers for effective TrEO has led to recent methods for adjusting the extent of solution transfer across tasks online by learning inter-task similarities from data generated during the course of the search. A representative example is the probabilistic *adaptive model-based transfer* *evolutionary algorithm* (AMTEA), in which the degree of overlap between the optimized search distributions of the source and target tasks (captured via stacked estimation of the target population density [25]) is seen as an online measure of their similarity – mandating the extent of transfer between them [26].

The probabilistic transfer framework in AMTEA has shown its effectiveness in boosting optimization performance in a range of benchmark examples where, given a set of source tasks, there exists at least one whose optimized search distribution closely matches (overlaps with) that of the target. The salient facet of this approach is that the transfer of solutions only occurs from tasks that are deemed to bear similarity with the target. In cases where the source and target *domains*, i.e., the features of their respective search spaces (e.g., dimensionality) and the distribution of good solutions in those spaces, do not match, little to no transfer would occur between them. However, we claim that the lack of source-target search distribution overlap does not imply the absence of useful information in the source. For instance, consider a hypothetical situation where the fitness functions of two distinct optimization tasks are negatively correlated [27]. Here, although existing methods would preemptively curtail inter-task solution transfer, the presence of negative correlation marks a valuable piece of information that, if successfully gleaned, could be exploited to the benefit of the target search (as shall be illustrated in "Motivating Domain Adaptations in TrEO"). Another scenario where narrow focus on the curbing of negative effects can lead to the conservative cancellation of knowledge transfer is when the source and target tasks have different search space dimensionality; this not only hampers similarity estimation between potentially related tasks, but also precludes the direct transfer of solutions between them.

In order to fully tap the potential of transfer optimization in such scenarios of source-target domain mismatch, this paper presents a novel perspective on *domain adaptation* in the context of TrEO. The basic idea is to transform the input feature spaces (i.e., the search spaces) of the source and target tasks to a common representation space, with the specific objective of inducing greater overlap between their respective optimized search distributions in that space. Once such a spatial transformation is constructed, either in a hand-crafted manner or by learning from data generated online, probabilistic transfers can be launched in the common space to achieve a better balance between the curbing of negative transfer and the active stimulation of positive transfers.

Our twofold contribution in this paper is summarized in what follows.

- To the best of our knowledge, this is the first work to establish a comprehensive probabilistic formulation of domain adaptation in TrEO. We categorize our formulation based on different types of spatial transformations, namely, (i) *source-to-target*, (ii) *target-to-source*, and (iii) *source-cum-target* maps, aimed at reducing the degree of inter-task domain mismatch for inducing effective knowledge transfers. The mathematical descriptions of the three categories are derived, followed by an overarching unification of the proposed concepts.

- Secondly, a *domain adaptive transfer evolutionary algorithm* is proposed, supporting both offline construction and online data-driven learning of *non-linear* source-to-target mapping functions. Unlike existing methods [28-30], the non-linear map (in continuous spaces) copes with arbitrary differences between the source and target search space dimensionality, while simultaneously guarding against the threat of negative transfers. Fig. 1 gives a visual illustration of our proposed domain adaptive transfer method when compared with existing state-of-the-art TrEO approaches, in the form of block diagrams.

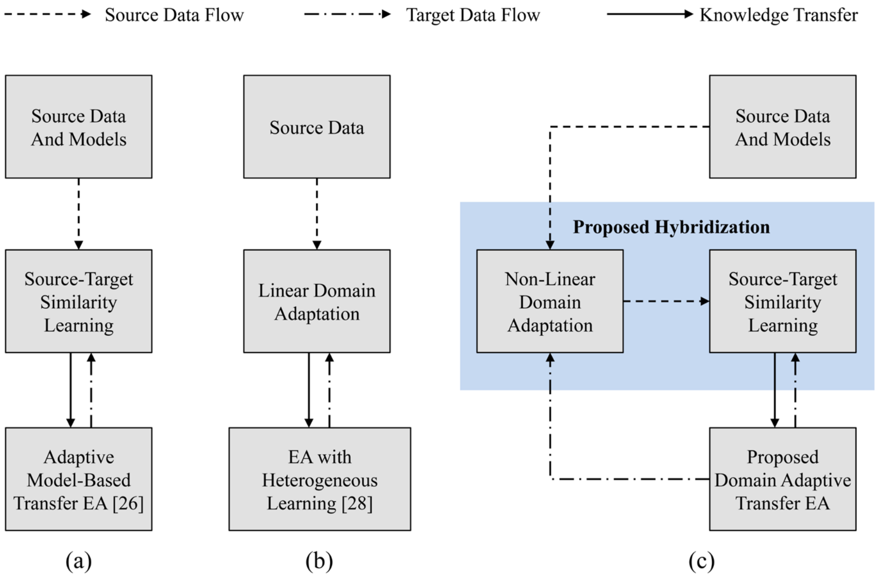

Source Data Flow
Target Data Flow
Knowledge Transfer
Source Data And Models
Source-Target Similarity Learning
Adaptive Model-Based Transfer EA [26]
(a)
Source Data
Linear Domain Adaptation
EA with Heterogeneous Learning [28]
(b)
Source Data And Models
Proposed Hybridization
Non-Linear Domain Adaptation
Source-Target Similarity Learning
Proposed Domain Adaptive Transfer EA
(c)

Fig. 1. Block diagrams that visually summarize the contributions of our proposed domain adaptive transfer method when compared with state-of-the-art TrEO approaches. (a) Adaptive model-based transfer EA [26]. (b) EA with learning across heterogeneous problems [28]. (c) Proposed domain adaptive transfer EA.

The remainder of this paper is organized as follows. “Preliminaries” contains a brief background of TrEO. “Motivating Domain Adaptations in TrEO” and “Categorizing Domain Adaptations in TrEO” are dedicated to exemplifying the importance of domain adaption and establishing its probabilistic formulation, respectively. Our algorithmic contribution is detailed in “The Proposed Domain Adaptive Transfer Evolutionary Algorithm”. Subsequently, “A Study of Hand-Crafted Domain Adaptation in TrEO” presents a pedagogical example of offline construction of source-to-target maps in a class of combinatorial optimization problems, whereas “Empirical

Assessment of Non-Linear Domain Adaptation in TrEO" experimentally verifies the efficacy of online learning of non-linear spatial transformations in continuous search spaces. In the latter, both synthetic functions and a practical example in multi-location inventory planning are considered. The paper is concluded in "Conclusion" with a summary and directions for future research work.

2. Preliminaries

Consider a *series* of T distinct optimization tasks denoted as $\mathcal{T}_1, \mathcal{T}_2, \dots, \mathcal{T}_T$, with objective (fitness) functions $f_1, f_2, \dots, f_T$, respectively. In this section, we make the explicit assumption that all the fitness functions are predefined in the same input feature space (i.e., search space) $\mathcal{X}$. (This assumption shall be relaxed in "Categorizing Domain Adaptations in TrEO".) Any task $\mathcal{T}_s$ can then be stated as,

$$\max_{x \in \mathcal{X}} \; f_s(\boldsymbol{x}), \tag{1}$$

where we consider maximization without loss of generality. (Constraints are ignored for simplicity of exposition.) When carrying out optimization via probabilistic model-based search algorithms, Eq. (1) is effectively reformulated as,

$$\max_{p_s(x)} \; \int_{\mathcal{X}} f_s(\boldsymbol{x}) \cdot p_s(\boldsymbol{x}) \cdot \mathrm{d}\boldsymbol{x}. \tag{2}$$

Here, $p_s(\boldsymbol{x})$ represents the distribution of a population $P_s = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_n\}$ of candidate solutions for $\mathcal{T}_s$. A population P_s^* – with distribution $p_s^*(\boldsymbol{x})$ – of *good* (near-optimal) solutions for $\mathcal{T}_s$ satisfies the following,

$$\int_{\mathcal{X}} f_s(\boldsymbol{x}) \cdot p_s^*(\boldsymbol{x}) \cdot \mathrm{d}\boldsymbol{x} \geq f_s^* - \varepsilon_s, \tag{3}$$

where f_s^* represents the globally optimum fitness value, and $\varepsilon_s (> 0)$ is a small convergence threshold. Notice that letting ε_s be zero could lead to the unsolicited case of a degenerate distribution $p_s^*(\boldsymbol{x})$.

Given the above, in this paper, the domain $\mathcal{D}$ of an optimization task is deemed to consist of two components: its input feature space $\mathcal{X}$, and the probability distribution of good solutions $p^*(\boldsymbol{x})$ in that space; i.e., $\mathcal{D} = \{\mathcal{X}, p^*(\boldsymbol{x})\}$.

This expands on a previous description of domains presented in [7], where the second component had been excluded. However, referring to [31], the optimized search distribution is regarded as a necessary component to fully characterize the domain of an optimization task. A case in point (which shall be revisited in "A Case-Study in Multi-Location Inventory Planning") is the planning of inventory levels at a chain of retailers for different products with different demand patterns. Even if the search space corresponding to each product is the same (i.e., it spans the same retailers), the optimum inventory level will vary significantly across product types – leading to different search distributions and thus reflecting different problem domains.

2.1. Probabilistic Formulation of TrEO

In the sequential transfer optimization setting, it is assumed that when addressing *target* task $\mathcal{T}_T$, we have access to the data (including the optimized search distribution models) drawn from $T-1$ *source* tasks $\mathcal{T}_1, \mathcal{T}_2, \dots, \mathcal{T}_{T-1}$ that have been previously encountered. The crux of the TrEO paradigm lies in harnessing the available knowledge (from the sources) in the form of search biases that could accelerate the target search. The modified formulation of the target from the perspective of probabilistic model-based search then becomes [6],

$$\max_{w_1, w_2, \dots, w_T, p_T(x)} \int_{\mathcal{X}} f_T(\boldsymbol{x}) \cdot \left[\sum_{s=1}^{T-1} w_s \cdot p_s^*(\boldsymbol{x}) + w_T \cdot p_T(\boldsymbol{x})\right] \cdot \mathrm{d}\boldsymbol{x}, \tag{4}$$

where $\sum_{s=1}^{T} w_s = 1$, and $w_s \geq 0 \ \forall s \in \{1,2,\dots,T\}$. The coefficients $w_1, w_2, \dots, w_T$ in the probability mixture model $\left[\sum_{s=1}^{T-1} w_s \cdot p_s^*(\boldsymbol{x}) + w_T \cdot p_T(\boldsymbol{x})\right]$ specify the extent to which the sources influence the target search – and are therefore also referred to as *transfer coefficients*. If there exists at least one source, say $\mathcal{T}_{s'} \in \{\mathcal{T}_1, \mathcal{T}_2, \dots, \mathcal{T}_{T-1}\}$, whose optimized search distribution $p_{s'}^*(\boldsymbol{x})$ significantly overlaps with $p_T^*(\boldsymbol{x})$ – which we denote as $p_{s'}^*(\boldsymbol{x}) \approx p_T^*(\boldsymbol{x})$s – then assigning $w_{s'} \to 1$ and sampling from the resultant mixture will yield solutions that map close to f_T^* with high probability.

In practice, of course, the distribution $p_T^*(\boldsymbol{x})$ is *a priori* unknown, which is what makes it challenging to select the appropriate source probabilistic model for transfer. On one hand, assigning a high transfer coefficient to an inappropriate source model will lead to the sampling of inferior solutions for the target task. On the other hand, not assigning a sufficiently high transfer coefficient to a relevant source model will lead to the loss of useful solutions that could help boost the target search. The challenge faced is even greater in black-box optimization settings where little can be said with any certainty about inter-task similarities. Therefore, in order to effectively resolve Eq. (4), it is important for a transfer optimization algorithm to be able to deduce accurate mixture coefficients online, based on the data generated during the course of the target search. It is to this end that an adaptive probabilistic model-based transfer evolutionary algorithm has recently been proposed in [26].

3. Motivating Domain Adaptations in TrEO

The success of probabilistic model-based transfer has been demonstrated for a range of benchmark examples where there exists at least one source task whose optimized search distribution overlaps with that of the target. In these cases, solutions sampled from the related source model survive in the target population (due to the survival of the fittest principle in EAs), and, in turn, drive up the associated transfer coefficient in Eq. (4). The opposite occurs for source models that are not related to the target, pre-emptively curbing the threat of negative transfer.

In contrast, consider a situation in which there is not a single source whose domain matches that of the target. Even assuming the same predefined search space $\mathcal{X}$, their optimized search distributions do not overlap in that space; i.e., $p_T^*(\boldsymbol{x}) \not\approx p_s^*(\boldsymbol{x})\ \forall s \in \{1,2,\dots,T-1\}$. In such cases, the probability mixture modelling framework will serve to suppress the transfer of solutions from the various sources to the target task. It is at this juncture we note that the lack of significant search distribution overlap does not necessarily imply the absence of useful information in a source. We argue that there may exist inter-task relationships that although remain hidden in the originally defined search spaces, can be revealed under certain spatial transformations. Overlooking the latent relationships would potentially lead to the loss of useful solutions from the source that could help boost the target optimization search.

As a substantiation of our claim, take the example of the synthetic source and target fitness functions (denoted as $f_S(x)$ and $f_T(x)$, respectively) in Fig. 2, where the 1-D search space is box-constrained in $[-1, 1]$. Given the same search space, the source domain $\mathcal{D}_S$ and target domain $\mathcal{D}_T$ are distinguished by their respective optimized search distributions, $p_S^*(x)$ and $p_T^*(x)$. As is clear, the two functions are negatively correlated (one decreases in the direction in which the other increases), which indicates that $p_S^*(x)$ and $p_T^*(x)$ will be focused at opposite ends of the feasible space. In this case, the dissonance in optimized distributions gives rise to a scenario of source-target domain mismatch, i.e., $p_S^*(x) \not\approx p_T^*(x) \Rightarrow \mathcal{D}_S \neq \mathcal{D}_T$. Therefore, directly sampling solutions from $p_S^*(x)$ for the purpose of the target task would lead to negative transfer. But, as also shown in Fig. 2, under a simple source-to-target mapping $M_S(x) = -x$, a strong (positive) correlation is induced between the functions $f_S(M_S^{-1}(x))$ and $f_T(x)$. Sampling from the source distribution $p_{M_S}^*(x)$, obtained by transforming P_S^* to $M_S(P_S^*)$, will then yield near-optimal solutions for the target.The utility of such transformations extends beyond the 1-D case, as highlighted by the following result.

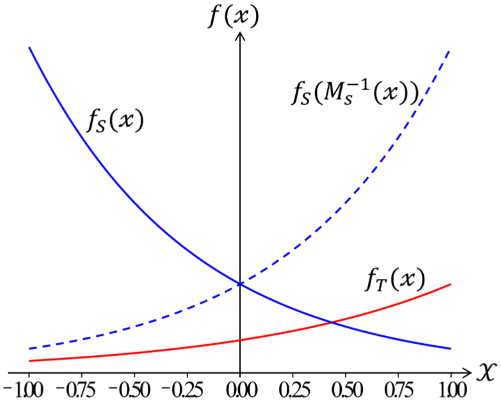


Fig. 2. An example where a mapping $M_S(x) = -x$ applied to the source solutions induces strong (positive) correlation between the transformed source function $f_S(M_S^{-1}(x))$ and the target fitness function $f_T(x)$, resulting in greater overlap between their respective optimized search distributions.

***Proposition 1**: Given a pair of continuous fitness functions $f_a(\boldsymbol{x})$ and $f_b(\boldsymbol{x})$ where $\boldsymbol{x} \in \mathcal{X} \subseteq \mathbb{R}^d$, if M_a is an invertible mapping such that $f_a(M_a^{-1}(\boldsymbol{x}_1)) < f_a(M_a^{-1}(\boldsymbol{x}_2)) \Rightarrow f_b(\boldsymbol{x}_1) < f_b(\boldsymbol{x}_2)$ for any $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, then their respective optimized search distributions satisfy $p_{M_a}^*(\boldsymbol{x}) \approx p_b^*(\boldsymbol{x})$ as $\varepsilon_a, \varepsilon_b \rightarrow 0$.*

These kinds of transformations of the search spaces of the source and/or target tasks to a common representation space, inducing greater overlap between their respective optimized search distributions, motivate domain adaptations in the context of TrEO. The success of domain adaptation however depends on the construction of the mapping function. On one hand, in areas where sufficient domain expertise is available, a hand-crafted (offline) specification of the mapping function is possible. On the other, for general black-box optimization settings with a scarcity of prior knowledge, the data generated during the course of the target search could be utilized for learning appropriate mapping functions online.

4. Categorizing Domain Adaptations in TrEO

Different from what is depicted in Fig. 2, alternate forms of spatial transformations for domain adaption may be conceived. For instance, it is possible to construct a target-to-source map $M_T(\boldsymbol{x})$ that transforms $f_T(\boldsymbol{x})$ to $f_T(M_T^{-1}(\boldsymbol{x}))$, inducing strong correlation with $f_S(\boldsymbol{x})$. Joint transformation of both the source and target search spaces to a common space (labelled $\mathcal{X}_C$) via $M_S^C(\boldsymbol{x})$ and $M_T^C(\boldsymbol{x})$, respectively, is yet another option. In this section, we propose three categories of domain adaptation in TrEO based on these considerations, together with their mathematical formulations from the standpoint of probabilistic model-based search.

We relax our assumption from "Preliminaries" that the tasks are predefined in the same search space. However, we simplify our exposition by limiting to the case of a single source. Scaling to a large number of sources can be carried out as in [32].

4.1. Category 1: Source-to-Target Map

Let the search space of source task $\mathcal{T}_S$ be $\mathcal{X}_S \subseteq \mathbb{R}^{d_S}$ and that of the target task $\mathcal{T}_T$ be $\mathcal{X}_T \subseteq \mathbb{R}^{d_T}$. In general, $\mathcal{X}_S \neq \mathcal{X}_T$ and $d_S \neq d_T$. Modifying Eq. (4), the problem formulation inclusive of a source-to-target map becomes,

$$\max_{w_S, w_T, p_T(x_T)} \int_{\mathcal{X}_T} f_T(\boldsymbol{x}_T) \cdot [w_S \cdot p_{M_S}^*(\boldsymbol{x}_T) + w_T \cdot p_T(\boldsymbol{x}_T)] \cdot \mathrm{d}\boldsymbol{x}_T,$$

$$\text{s.t.}, w_S + w_T = 1 \text{ and } w_S, w_T \geq 0. \tag{5}$$

Here, $\boldsymbol{x}_T \in \mathcal{X}_T$, $M_S: \mathcal{X}_S \to \mathcal{X}_T$, and $p_{M_S}^*$ is the transformation of $p_S^*(\boldsymbol{x}_S)$ under mapping function M_S where $\boldsymbol{x}_S \in \mathcal{X}_S$. The objective behind introducing M_S in Eq. (5) is to stimulate $p_{M_S}^*(\boldsymbol{x}_T) \approx p_T^*(\boldsymbol{x}_T)$, so that sampling the mixture model $\left[w_S \cdot p_{M_S}^*(\boldsymbol{x}_T) + w_T \cdot p_T(\boldsymbol{x}_T)\right]$ yields good solutions for the target task when $w_S > 0$.

An illustration of the source-to-target map is provided in Fig. 3(a). In recent work, M_S has been realized in the form of a *linearized* transformation function from $\mathcal{X}_S$ to $\mathcal{X}_T$ [28]; i.e.,

$$M_S^L(\boldsymbol{x}_S) = A \cdot \boldsymbol{x}_S + \boldsymbol{b}, \tag{6}$$

where matrix $A \in \mathbb{R}^{d_T \times d_S}$, and $\boldsymbol{b} \in \mathbb{R}^{d_T}$ is a bias vector. While a linear map is workable when $d_T \leq d_S$, it does not suffice when $d_T > d_S$. This is because the distribution $p_{M_S^L}^*$ becomes degenerate when $d_T > d_S$; although $p_{M_S^L}^*$ is defined in the d_T-dimensional space $\mathcal{X}_T$, it is supported only on a lower d_S-dimensional subspace. In [26], an ad-hoc workaround of padding the source model with extra variables was proposed to compensate for the difference between d_T and d_S. *A more principled approach however – one that can cope with both $d_T \leq d_S$ as well as $d_T > d_S$ without resulting in a degenerate distribution – is to let M_S be non-linear.* Another idea is to consider a target-to-source map instead, which brings us to the second category.

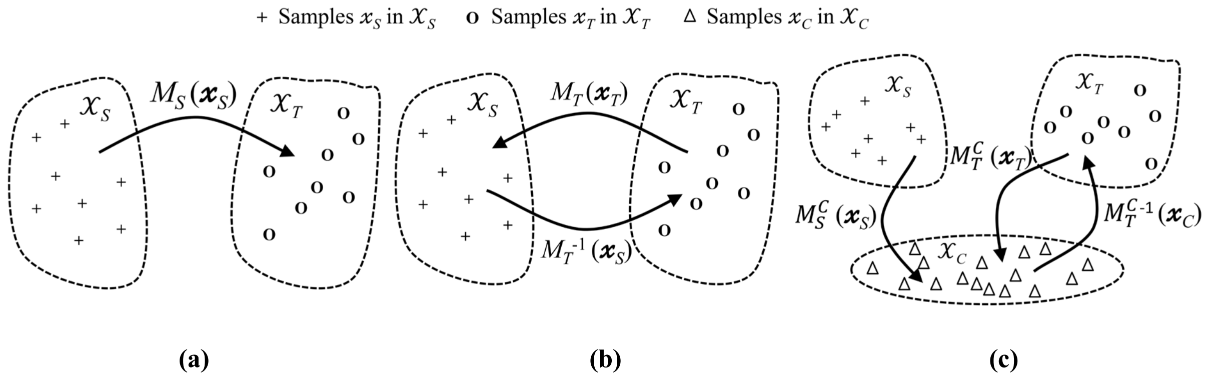

+ Samples x_S in $\mathcal{X}_S$
o Samples x_T in $\mathcal{X}_T$
△ Samples x_C in $\mathcal{X}_C$
$\mathcal{X}_S$
$\mathcal{X}_T$
$M_S(\boldsymbol{x}_S)$
$M_T(\boldsymbol{x}_T)$
$M_T^{-1}(\boldsymbol{x}_S)$
$\mathcal{X}_C$
$M_S^C(\boldsymbol{x}_S)$
$M_T^C(\boldsymbol{x}_T)$
$M_T^{C-1}(\boldsymbol{x}_C)$
(a)
(b)
(c)

Fig. 3. Illustration of the proposed categories of search space transformations for domain adaptation in TrEO. (a) Source-to-Target Map. (b) Target-to-Source Map. (c) Source-cum-Target Maps.

4.2.Category 2: Target-to-Source Map

In accordance with Eq. (4), the probabilistic formulation of TrEO inclusive of a target-to-source map can be written as,

$$\max_{w_S, w_T, p_T(x_T)} \int_{\mathcal{X}_S} f_T(M_T^{-1}(\boldsymbol{x}_S)) \cdot \left[w_S \cdot p_S^*(\boldsymbol{x}_S) + w_T \cdot p_{M_T}(\boldsymbol{x}_S)\right] \cdot \mathrm{d}\boldsymbol{x}_S,$$

$$\text{s.t.}, w_S + w_T = 1 \text{ and } w_S, w_T \geq 0. \tag{7}$$

Above, $M_T: \mathcal{X}_T \to \mathcal{X}_S$ and p_{M_T} is obtained by transforming the argument of $p_T(\boldsymbol{x}_T)$ via M_T. For applications where problem sizes increase over time (i.e., $d_T > d_S$), it is feasible for the mapping function to be linear under the formulation of Eq. (7), as p_{M_T} will not be degenerate in $\mathcal{X}_S$; thus, enabling building and sampling of the mixture model $\left[w_S \cdot p_S^*(\boldsymbol{x}_S) + w_T \cdot p_{M_T}(\boldsymbol{x}_S)\right]$.

The purpose of introducing M_T in Eq. (7) is to stimulate $p_S^*(\boldsymbol{x}_S) \approx p_{M_T}^*(\boldsymbol{x}_S)$, where$p_{M_T}^*$ is the transform of the *a priori* unknown $p_T^*(\boldsymbol{x}_T)$. Given such a transformation, samples drawn from the resultant mixture, when mapped back to $\mathcal{X}_T$ via M_T^{-1} as depicted in Fig. 3(b), will yield near-optimal target solutions. According to Proposition 1, one way to attain the desired outcome is to learn M_T and M_T^{-1} so that a strong ordinal correlation is induced between $f_S(\boldsymbol{x}_S)$ and $f_T(M_T^{-1}(\boldsymbol{x}_S))$. Here, M_T^{-1} marks a slight abuse of notation as M_T may not be precisely invertible. Nevertheless, we assume that in learning M_T and M_T^{-1}, the minimization of the reconstruction error of the composition $(M_T^{-1} \circ M_T)(\boldsymbol{x}_T)$ is accounted for. This can be achieved by optimizing against the following objective [33],

$$\min_{M_T, M_T^{-1}} \ \lambda_1 \cdot \sum\nolimits_{\forall \boldsymbol{x}_T \in P_T} \|\boldsymbol{x}_T - (M_T^{-1} \circ M_T)(\boldsymbol{x}_T)\|_2^2 - \lambda_2 \cdot \Gamma\left(f_S(\boldsymbol{x}_S); f_T\left(M_T^{-1}(\boldsymbol{x}_S)\right)\right), \tag{8}$$

where P_T is the target population of solutions, $\Gamma(f_S\,; f_T)$ is an estimate of the correlation between the two fitness functions [27], while λ_1 and λ_2 are weighting terms.

An important distinction between the first and second categories is in the need to explicitly define an inverse map M_T^{-1} in the latter, which would increase the complexity of its learning procedure compared to the former.

4.3. Category 3: Source-cum-Target Maps

Imagine a special case where the behaviours of the source and target fitness functions are correlated with respect to a small subset of features in their overall search spaces. An illustration is depicted in Fig. 3(c), where $\mathcal{X}_C \subseteq \mathbb{R}^{d_C}$ represents the common low-dimensional subspace. The probabilistic formulation of Eq. (4) can be modified to such situations involving the transformation of both source and target search spaces as,

$$\max_{w_S, w_T, p_T(x_T)} \int_{\mathcal{X}_C} f_T({M_T^C}^{-1}(\boldsymbol{x}_C)) \cdot \left[w_S \cdot p^*_{M_S^C}(\boldsymbol{x}_C) + w_T \cdot p_{M_T^C}(\boldsymbol{x}_C)\right] \cdot \mathrm{d}\boldsymbol{x}_C,$$

$$\text{s.t.}, w_S + w_T = 1 \text{ and } w_S, w_T \geq 0. \tag{9}$$

Here, $M_S^C: \mathcal{X}_S \rightarrow \mathcal{X}_C$, $M_T^C: \mathcal{X}_T \rightarrow \mathcal{X}_C$ and $p^*_{M_S^C}$, $p_{M_T^C}$ are the respective transformed distributions in $\mathcal{X}_C$. We assume that M_T^C is *approximately* invertible, which is achievable by minimizing the reconstruction error.

M_S^C and M_T^C could also take the form of *feature selection* operations expressible as $M_S^C(\boldsymbol{x}_S) = \boldsymbol{x}_S \odot \boldsymbol{\beta}_S^C$ and $M_T^C(\boldsymbol{x}_S) = \boldsymbol{x}_T \odot \boldsymbol{\beta}_T^C$, where $\boldsymbol{x} \odot \boldsymbol{\beta}$ is the Hadamard product (element-wise multiplication) between $\boldsymbol{x}$ and $\boldsymbol{\beta}$, while $\boldsymbol{\beta}_S^C \in \{0,1\}^{d_S}$ and $\boldsymbol{\beta}_T^C \in \{0,1\}^{d_T}$ are binary indicator vectors whose entries are 1 for the selected subset of features and 0 otherwise [34]. Since both $\boldsymbol{\beta}_S^C$ and $\boldsymbol{\beta}_T^C$ map solutions to the common space $\mathcal{X}_C$ of dimensionality d_C, they must satisfy $||\boldsymbol{\beta}_S^C||_0 = ||\boldsymbol{\beta}_T^C||_0 = d_C$.

4.4. A Summary and Unified Definition

The singular theme of the three categories of transformations presented heretofore has been to map the source and target tasks, originally defined in different search spaces (i.e., $\mathcal{X}_S \neq \mathcal{X}_T$), to a common representation space in which positive transfers can be stimulated. In *Category 1*, $\mathcal{X}_T$ serves as the common space into which the source is mapped; in *Category 2*, $\mathcal{X}_S$ is the common space into which the target is mapped; and in *Category 3*, the source and target are jointly transformed to a third space $\mathcal{X}_C$.

Based on the above, we now give a general and unified definition of domain adaptation in TrEO.

Definition 1 *(Domain Adaptation in TrEO): Given source optimization task $\mathcal{T}_S$ with domain $\mathcal{D}_S = \{\mathcal{X}_S, p_S^*(\boldsymbol{x}_S)\}$, and target task $\mathcal{T}_T$ with domain $\mathcal{D}_T = \{\mathcal{X}_T, p_T^*(\boldsymbol{x}_T)\}$, where $\mathcal{X}_S \neq \mathcal{X}_T$ or $p_S^*(\boldsymbol{x}_S) \not\approx p_T^*(\boldsymbol{x}_T)$, domain adaptation* *aims to help boost the optimization of $\mathcal{T}_T$ through search space transformations that stimulate positive knowledge transfers across tasks in the transformed space.*

It is worth reiterating that in optimization problems, the target search distribution $p_T^*(\boldsymbol{x}_T)$ would not be known in advance, and what is more, no task-specific data may be available prior to the onset of the search. This marks a key distinguishing factor between the notion of domain adaptation unveiled in this paper and that prevalent in the machine learning literature (where a plethora of methods have been proposed under the assumption of source and target data availability [35]). The algorithm put forth in the next section thus realizes a novel perspective on domain adaptation, in the context of TrEO.

5. The Proposed Domain Adaptive Transfer Evolutionary Algorithm

At the core of our proposed approach is the probabilistic *adaptive model-based transfer evolutionary algorithm* (AMTEA) [26]. In this section, we first provide a brief overview of canonical probabilistic model-based search as well as the AMTEA, and subsequently enhance the algorithm with non-linear domain adaptation. The incorporation of non-linear maps is chosen herein due to its flexibility in coping with heterogeneous scenarios of TrEO where $d_T \leq d_S$ or $d_T > d_S$ without leading to degenerate distributions.

5.1. Canonical Probabilistic Model-Based Search

Probabilistic model-based search algorithms (also known as estimation of distribution algorithms [36]) are derivative-free optimizers that guide a population of candidate solutions towards the global optimum f^* of a fitness function $f(\boldsymbol{x})$ by iterating over the steps below.

Step 1: Define an initial search distribution model $p(\boldsymbol{x}|t=0)$ over the search space $\mathcal{X}$.

Step 2: At iteration t, sample and evaluate a set $P'(t)$ of N candidate solutions (referred to as *offspring*) from the current model $p(\boldsymbol{x}|t-1)$.

Step 3: Based on the evaluated fitness, select a set of $n(< N)$ *parent solutions* from $P'(t)$; denoted as $P(t)$.

Step 4: Build probabilistic model $p(\boldsymbol{x}|t)$ of the parent population $P(t)$.

Step 5: Increment the iteration count and return to Step 2 until a termination condition is met.

A well-known result and proof supporting the asymptotic global convergence of the above steps is presented in [37].

Theorem 1 *(Zhang and Muhlenbein [37]): Let $p(\boldsymbol{x}|t=0)$ be positive and continuous everywhere in $\mathcal{X}$. Assuming $n \to \infty$, if the probabilistic model $p(\boldsymbol{x}|t)$ – from which offspring $P'(t+1)$ are sampled – captures the true underlying distribution of the parent population $P(t)$, then,*

$$\lim_{t\to\infty} \int_{\mathcal{X}} f(\boldsymbol{x}) \cdot p(\boldsymbol{x}|t) \cdot d\boldsymbol{x} \; = f^*. \tag{10}$$

The main takeaway from Theorem 1 is that global convergence results from the model $p(\boldsymbol{x}|t)$ capturing the true distribution of the parent population $P(t)$. It is this insight that is utilized in the AMTEA to deduce accurate transfer coefficients in Eq. (4).

5.2. Basic Learning of the Transfer Coefficients

The AMTEA makes the simplifying assumption that the various source and target tasks have been predefined in a unified space $\mathcal{X}$, so that a probability mixture model can be built in that space. The algorithm does not include the construction of the space itself.

As in Step 4 of probabilistic model-based search, at any iteration t, a probability mixture model of the target task's parent population $P_T(t)$ is built. To this end, the *stacked density estimation* procedure [25] is used to learn the transfer coefficients $w_1, w_2, \dots, w_T$ in Eq. (4) by maximizing the following log-likelihood;

$$\max_{w_1,w_2,\dots,w_T} \sum\nolimits_{\forall \boldsymbol{x} \in P_T(t)} \log\Big(\sum\nolimits_{s=1}^{T-1} w_s \cdot p_s^*(\boldsymbol{x}) + w_T \cdot p_T(\boldsymbol{x}|t)\Big). \tag{11}$$

Given a database of acquired source probabilistic models $\{p_1^*(\boldsymbol{x}), p_2^*(\boldsymbol{x}), \dots, p_{T-1}^*(\boldsymbol{x})\}$ and an intermediate target distribution $p_T(\boldsymbol{x}|t)$, the mathematical program in Eq. (11) can be solved to optimality by the classical *expectation-maximization* (EM) algorithm. The procedure is not reproduced herein for the sake of brevity; for a detailed description readers are referred to [6].

5.3. Domain Adaptive Learning and Optimization

Domain adaptation generalizes TrEO to cases of source-target heterogeneity where the assumption of a predefined unified space is relinquished. In contrast, for any source task $\mathcal{T}_s$, we consider $\mathcal{D}_s \neq \mathcal{D}_T$ and even $d_s \neq d_T$. Referring to the result in Proposition 1, a source-to-target mapping function $M_s: \mathcal{X}_s \to \mathcal{X}_T$ can be sought such that $p_{M_s}^*(\boldsymbol{x}_T) \approx p_T^*(\boldsymbol{x}_T)$; see details in the next sub-section. The program in Eq. (11) is thus modified to,

$$\max_{w_1,w_2,\ldots,w_T} \sum\nolimits_{\forall x_T \in P_T(t)} \log\left(\sum\nolimits_{s=1}^{T-1} w_s \cdot p^*_{M_s}(\boldsymbol{x}_T) + w_T \cdot p_T(\boldsymbol{x}_T|t)\right). \tag{12}$$

The maximum log-likelihood statement of Eq. (12) can also be solved to optimality via expectation-maximization to obtain the transfer coefficients under domain adaptation. A pseudocode of the resultant domain adaptive transfer evolutionary algorithm, which we refer to hereafter as the AMTEA+M_S, is provided in Algorithm 1.

Algorithm 1. Pseudocode of the proposed AMTEA+M_S

```
Input: Source database; $f_T$; $\mathcal{X}_T$; transfer interval $\Delta$
Output: Data and optimized search distribution for target $\mathcal{T}_T$
1:  Set $t = 0$
2:  Sample $N$ initial solutions uniformly at random from $\mathcal{X}_T$ to form $P'_T(t)$
3:  While stopping condition not met do
4:      Evaluate fitness of individuals in $P'_T(t)$ w.r.t. $f_T$
5:      Select parent solutions $P_T(t)$ from $P'_T(t)$
6:      If $mod\ (t, \Delta) \neq 0$ then
          /* Canonical EA search iteration */
7:        Generate $P'_T(t+1)$ from $P_T(t)$ applying standard crossover and/or mutation operators
8:      Else
          /* Sequential transfer iteration */
9:        Update $M_s$ $\forall s \in \{1,2,\ldots T-1\}$ based on current parent population $P_T(t)$
10:       Optimize $w_1, w_2, \ldots, w_T$ in Eq. (12) via EM algorithm
11:       Sample $N$ offspring from mixture model $\left[\sum_{s=1}^{T-1} w_s \cdot p^*_{M_s}(\boldsymbol{x}_T) + w_T \cdot p_T(\boldsymbol{x}_T|t)\right]$ to get $P'_T(t+1)$
12:     End if
13:     Set $t = t + 1$
14: End while
```

The difference between the AMTEA+M_S and the original AMTEA lies in steps 9 to 11 of Algorithm 1 where the constructed mapping functions enable the desired transformation of the source models to the target search space. As shall be experimentally shown, this enhancement is vital to achieving a better balance between the curbing of negative transfers and the active stimulation of positive transfers in the AMTEA+M_S.

An important facet of the AMTEA+M_S is that it intertwines probabilistic model-based search (at user-defined transfer interval Δ) with conventional crossover and/or mutation based evolutionary updates. This, as suggested in [38], helps to reduce the (albeit small) computational cost associated with repeated building of the probability mixture models.

Notably, the workflow of Algorithm 1 is not limited to any specific construction / update of source-to-target maps M_s in step 9. In areas where sufficient domain expertise is available, a hand-crafted (offline) specification of M_s is possible. However, in the general setting of black-box optimization tasks with scarcity of prior knowledge, the data generated during the course of the target optimization search could be utilized for learning appropriate mapping functions online. To this end, an online construction of non-linear spatial transformations for continuous search spaces is presented next.

5.4. Online Construction of Non-linear Source-to-Target Maps

As discussed in "Category 1: Source-to-Target Map", a non-linear construction of M_s, which we denote hereafter as $M_s^{NL}: \mathcal{X}_s \to \mathcal{X}_T$, offers greater flexibility than its linear counterpart in dealing with heterogeneous examples of TrEO where either $d_T \leq d_s$ or $d_T > d_s$. Moreover, returning to Eq. (5), we see that the first category of domain adaptation (i.e., source-to-target maps) does not require construction of an inverse mapping function. Thus, the learning complexity of source-to-target maps is expected to be lower than the other two categories.

Taking a cue from [28], we assume that while optimizing a source task $\mathcal{T}_s$, the parent population $P_s(t) = \{\boldsymbol{x}_{s,1}, \boldsymbol{x}_{s,2}, \dots, \boldsymbol{x}_{s,n}\}$ at every iteration t of an EA had been cached. In the ongoing target task $\mathcal{T}_T$, let its parent solutions at the current (tth) iteration be $P_T(t) = \{\boldsymbol{x}_{T,1}, \boldsymbol{x}_{T,2}, \dots, \boldsymbol{x}_{T,n}\}$. The solutions in $P_s(t)$ and $P_T(t)$ are *sorted* in decreasing order of their fitness evaluations, resulting in a monotonic relationship between the two sets; i.e., $f_s(\boldsymbol{x}_{s,i}) > f_s(\boldsymbol{x}_{s,j}) \implies f_T(\boldsymbol{x}_{T,i}) \geq f_T(\boldsymbol{x}_{T,j})$, and vice versa. Consequently, for large n, building a mapping $M_s^{NL}(\boldsymbol{x}_s)$ from $P_s(t)$ to $P_T(t)$ will locally, within the support of distribution $p_T(\boldsymbol{x}_T|t)$, induce strong (positive) ordinal correlation between $f_s({M_s^{NL}}^{-1}(\boldsymbol{x}_T))$ and $f_T(\boldsymbol{x}_T)$. (${M_s^{NL}}^{-1}$ is a hypothetical inversion that need not be explicitly derived in practice.) Assuming the global optimums $\boldsymbol{x}_s^*$ and $\boldsymbol{x}_T^*$ to be contained within the support of $p_s(\boldsymbol{x}_s|t)$ and $p_T(\boldsymbol{x}_T|t)$, respectively, we can then invoke Proposition 1 to approach $p_{M_s}^*(\boldsymbol{x}_T) \approx p_T^*(\boldsymbol{x}_T)$ in $\mathcal{X}_T$.

For any source-target pair, we propose M_s^{NL} to take the form of a two-layer feedforward neural network. The parameters $\boldsymbol{\theta}$ of this network are learned for mapping $P_s(t)$ to $P_T(t)$ by optimizing against the objective,

$$\min_{\boldsymbol{\theta}} \sum_{j=1}^{n} \left\| \boldsymbol{x}_{T,j} - M_s^{NL}(\boldsymbol{x}_{s,j}; \boldsymbol{\theta}) \right\|_2^2 + \lambda \|\boldsymbol{\theta}\|_2^2, \tag{13}$$

where the second term is an L_2 regularization of the parameters. Note that as the construction of M_s^{NL} is to be nested in step 9 of Algorithm 1, it is important for the learning to be fast in order to keep the computational overhead to a minimum.

With that in mind, consider the two-layer neural network M_s^{NL} to be formed by the composition,

$$M_s^{NL}(\boldsymbol{x}_s; \boldsymbol{\theta}) = \boldsymbol{\psi}_2\big(\boldsymbol{\psi}_1(\boldsymbol{x}_s)\big), \tag{14}$$

in which,

$$\boldsymbol{\psi}_1(\boldsymbol{x}_s) = \sigma(A_1 \cdot \boldsymbol{x}_s + \boldsymbol{b}), \tag{15}$$

and,

$$\boldsymbol{\psi}_2\big(\boldsymbol{\psi}_1(\boldsymbol{x}_s)\big) = A_2 \cdot \boldsymbol{\psi}_1(\boldsymbol{x}_s). \tag{16}$$

Let $\boldsymbol{\theta} = [\boldsymbol{\theta}_1, \boldsymbol{\theta}_2]$ in Eq. (14), such that $\boldsymbol{\theta}_1$ serves as the entries of matrix $A_1 \in \mathbb{R}^{d_h \times d_s}$ and the bias vector $\boldsymbol{b} \in \mathbb{R}^{d_h}$ in Eq. (15). Likewise, in Eq. (16), $\boldsymbol{\theta}_2$ serves as the entries of matrix $A_2 \in \mathbb{R}^{d_T \times d_h}$; d_h is the number of neurons in the hidden layer of the neural network, and σ symbolizes the non-linear sigmoidal activation function. In the spirit of [39], the elements of $\boldsymbol{\theta}_1$ are randomly assigned (by sampling from the standard normal distribution) and then fixed. Thereafter, entries of A_2 (i.e., $\boldsymbol{\theta}_2$) can be obtained using the closed-form expression,

$$A_2 = P_T \cdot [\boldsymbol{\psi}_1(P_s)]^{\text{tr}} \cdot \big(\boldsymbol{\psi}_1(P_s) \cdot [\boldsymbol{\psi}_1(P_s)]^{\text{tr}} + \lambda \cdot I_{d_h}\big)^{-1}, \tag{17}$$

where $[\cdot]^{\text{tr}}$ indicates a matrix transpose and I_{d_h} is an identity matrix of size d_h. Given the closed-form expression of Eq. (17), the computational complexity of learning M_s^{NL} is comparable to its linear counterpart.

A_2 is updated (online) in step 9 of Algorithm 1 based on population datasets $P_s(t)$ and $P_T(t)$ at iteration t. The AMTEA+M_S when equipped with such non-linear source-to-target maps is labelled as AMTEA+M_s^{NL}. The AMTEA+M_s^{NL} is operable for continuous search spaces where $\mathcal{X}_s \subseteq \mathbb{R}^{d_s}$ and $\mathcal{X}_T \subseteq \mathbb{R}^{d_T}$, facilitating the transfer of probabilistic models and solutions across tasks when $d_s \neq d_T$. An illustrative flowchart of the overall non-linear domain adaptive transfer evolutionary algorithm is provided in Fig. 4. Once a non-linear mapping M_s^{NL} is learned, the transformation of a source model $p_s^*(\boldsymbol{x}_s)$ to $p_{M_s^{NL}}^*(\boldsymbol{x}_T)$ is achieved in our implementation via simple Monte Carlo approximation; i.e., we first transform solution samples P_s^* to $M_s^{NL}(P_s^*)$, and then build a probabilistic model of $M_s^{NL}(P_s^*)$ in $\mathcal{X}_T$. Given continuous search spaces, probabilistic models $p_s^*(\boldsymbol{x}_s)$, $p_{M_s^{NL}}^*(\boldsymbol{x}_T)$, and $p_T(\boldsymbol{x}_T|t)$ are realizable in the form of multi-variate Gaussian distributions. For differentiable mapping functions, as is the case with neural networks, alternative analytical procedures for the non-linear transformation of Gaussians, offering greater computational speedup, are discussed in [40].

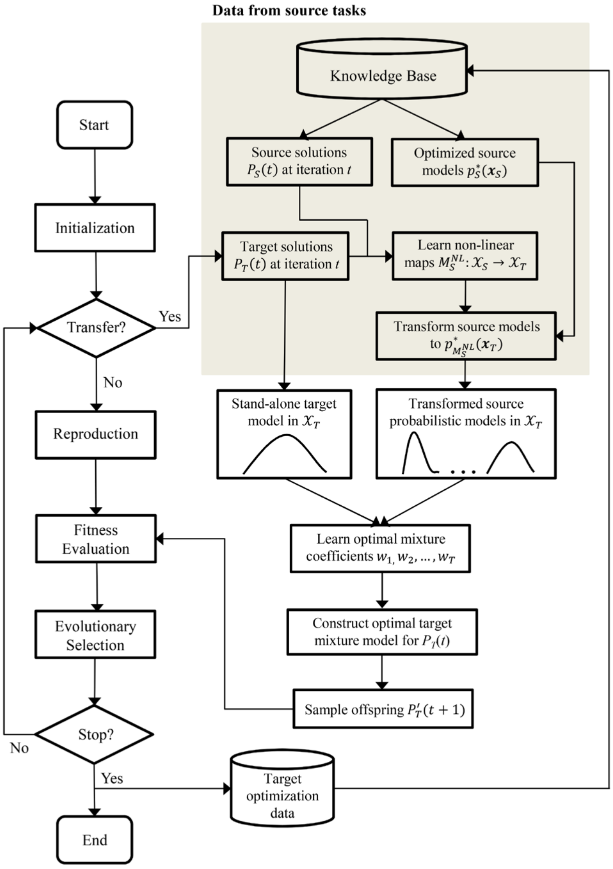

Data from source tasks
Knowledge Base
Start
Source solutions $P_S(t)$ at iteration t
Optimized source models $p_S^*(\boldsymbol{x}_S)$
Initialization
Target solutions $P_T(t)$ at iteration t
Learn non-linear maps $M_S^{NL}: \mathcal{X}_S \rightarrow \mathcal{X}_T$
Transfer?
Yes
Transform source models to $p^*_{M_S^{NL}}(\boldsymbol{x}_T)$
No
Stand-alone target model in $\mathcal{X}_T$
Transformed source probabilistic models in $\mathcal{X}_T$
Reproduction
Fitness Evaluation
Learn optimal mixture coefficients $w_1, w_2, \dots, w_T$
Evolutionary Selection
Construct optimal target mixture model for $P_T(t)$
Sample offspring $P_T'(t+1)$
Stop?
No
Yes
Target optimization data
End

Fig. 4. Illustrative flowchart of the non-linear domain adaptive AMTEA+M_S^{NL}. The key enhancement of the algorithm has been highlighted.

6. A Study of Hand-Crafted Domain Adaptation in TrEO

Before empirical verification of the AMTEA+M_S^{NL}, in this section, we use the 0-1 knapsack problem (KP) [41] as an illustrative toy example to showcase the efficacy of domain adaptation when hand-crafted offline. Due to its ubiquity in various practical applications, the KP has been widely studied over the years in computer science and operations research, with a number of heuristics and meta-heuristic optimization procedures proposed; the most prominent among them being *Dantzig's greedy approximation algorithm* [42].

Let there be a set of m items each of which has a value v_i and a weight q_i, $i \in \{1, \dots, m\}$, and a knapsack of capacity Q. Then, the KP asks to select a combination of items so that the total profit of the selected items is maximized without violating the knapsack's capacity constraint. The overall mathematical program is stated as,

$$\max_{x \in \{0,1\}^m} \ \sum_{i=1}^{m} x_i \cdot v_i,$$

$$\text{s.t.}, \sum_{i=1}^{m} q_i \cdot x_i \leq Q \text{ and } x_i \in \{0,1\}\ \forall i. \tag{18}$$

6.1. Prior Knowledge Guided Domain Adaptation Strategy

According to Dantzig's greedy approximation algorithm, given a KP instance, the first step is to compute the *efficiency ratio* of each of its items as,

$$r_i = \frac{v_i}{q_i}\ \forall i \in \{1,2,\dots,m\}. \tag{19}$$

The items are then inserted into the knapsack in *decreasing order* of efficiency, until the capacity constraint is violated. Although this algorithm does not guarantee optimality, it is able to provide good solutions in practice.

Dantzig's algorithm offers us the general insight that even across distinct KP instances (say, KP_S and KP_T), items with high efficiency ratio are similarly treated. Specifically, the item with the highest efficiency among all remaining items in a KP instance is preferentially considered for insertion into the knapsack. This insight allows the construction of a linear source-to-target domain adaptation M_S^L defined as,

$$M_S^L(\boldsymbol{x}_S) = A_{\mathcal{P}} \cdot \boldsymbol{x}_S, \tag{20}$$

where $A_{\mathcal{P}}$ is a *permutation matrix* for reordering items in KP_S such that their efficiency ratios (ordinally) correlate with those of the items in KP_T. M_S^L can be crafted offline, and is subsequently fixed throughout the target search.

6.2. Experimental Setting

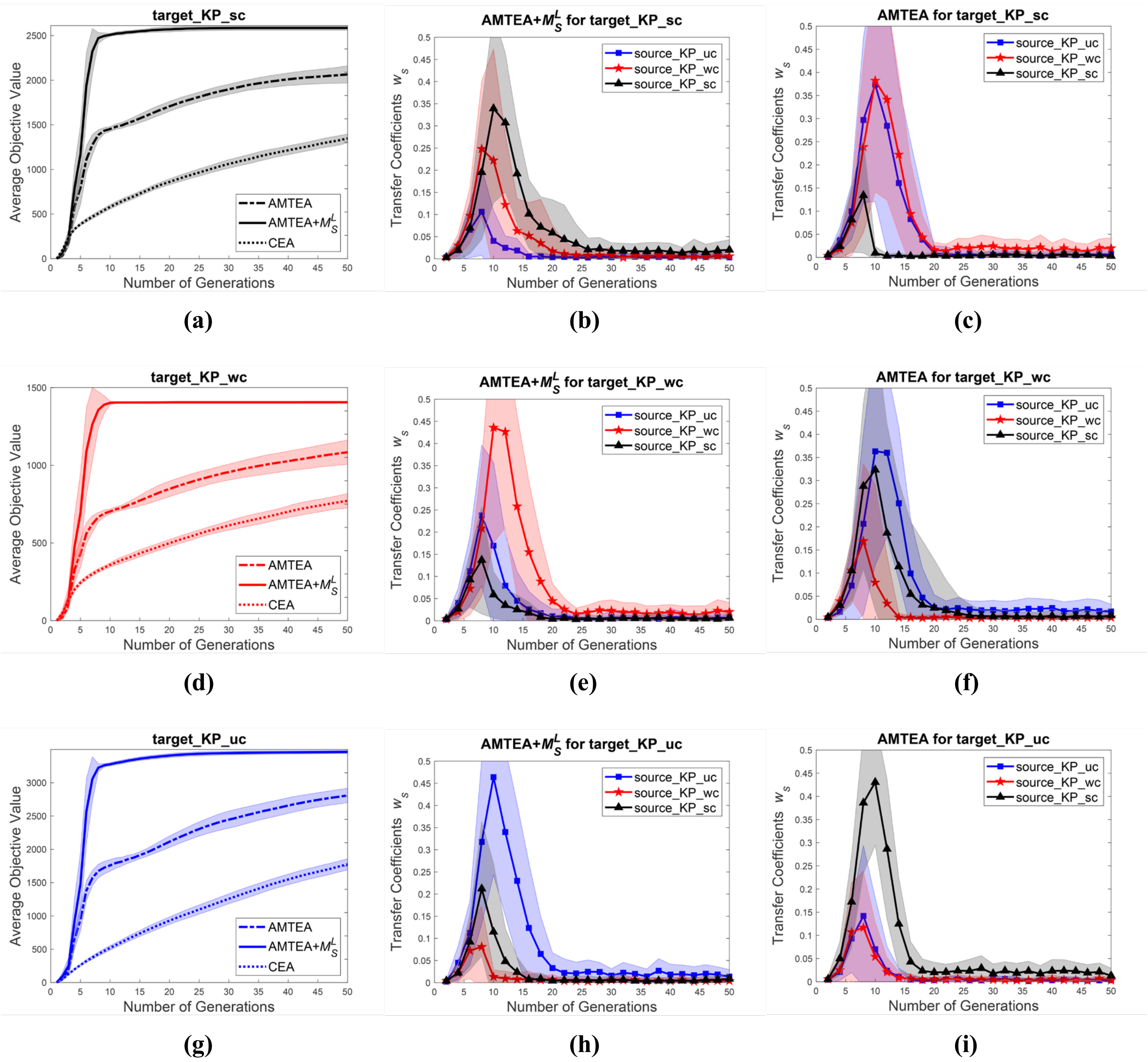
Three optimizers are considered in this comparative study. These are: (i) a *canonical evolutionary algorithm* (CEA) without transfer, (ii) the baseline AMTEA without domain adaptation, and (iii) the AMTEA+M_S^L. The parameter configurations of the three optimizers are kept consistent for fairness. They are all binary-coded, employ the uniform crossover operator (with crossover probability of 1), bit-flip mutation (with probability $1/m$), and a population size of $N = 50$. Furthermore, the same greedy solution repair procedure from [43] has been incorporated in the CEA, AMTEA and AMTEA+M_S^L to ensure capacity constraint satisfaction. We set the transfer interval $\Delta = 2$ (see Algorithm 1) in both AMTEA and AMTEA+M_S^L, at which stacked density estimation is launched to learn the optimal mixture of source and target probabilistic models. Each component of the mixture model takes the form of a univariate marginal distribution [44].

In this paper, all the source and target KP instances are generated independently at random as proposed in [43]. A KP instance can be classified either as *strongly correlated* (sc), *weakly correlated* (wc) or *uncorrelated* (uc), depending on the relationship between its items' values and weights. Our experiments are designed such that we have access to optimized search distribution models drawn from multiple source tasks, viz., “source_KP_sc”, “source_KP_wc” and “source_KP_uc”, while solving the target tasks “target_KP_sc”, “target_KP_wc” and “target_KP_uc”. The capacity of the knapsack is *restrictive* in all the instances, which is to say that very few items can be inserted into it.

6.3.Results

Fig. 5 depicts averaged convergence trends as well as the transfer coefficients learned by the AMTEA+M_S^L and AMTEA. Results in the leftmost column clearly show that AMTEA+M_S^L is able to converge much faster than the baseline AMTEA, which in turn performs significantly better than the CEA. In order to explain this outcome, we draw attention to the transfer coefficients learned (these are plotted in the central and rightmost columns of Fig. 5 for the AMTEA+M_S^L and AMTEA, respectively). Even though all the KP instances are generated independently at random, our prior intuition suggests that source and target instances that belong to the same class of KP instances (i.e., sc, wc, or uc) are likely to share the most in common.

Our intuition is closely borne out by the transfer coefficients of the AMTEA+M_S^L. As can be seen in the central column of Fig. 5 (i.e., in Fig. 5(b), (e), (h)), the AMTEA+M_S^L correctly assigns higher transfer coefficient values to source models that belong to the same class as the target task. As discussed in “Motivating Domain Adaptations in TrEO”, the observed outcome is achieved by the transformed source representation of Eq. (20) which successfully uncovers stronger inter-task relationships that remain hidden in the problem’s originally defined input feature space. This manifests in inducing greater source-target domain overlap, and hence, higher transfer coefficient values being learned for the most relevant sources. As a result, AMTEA+M_S^L’s convergence rates are enhanced by positive transfers from the most relevant source tasks. In contrast, as seen in the rightmost column of Fig. 5 (i.e., in Fig. 5(c), (f), (i)), AMTEA (without domain adaptation) tends to wrongly assign higher transfer coefficient values to source tasks that belong to a class different from that of the target; e.g., in the case of the *uncorrelated* “target_KP_uc”, the transfer coefficient assigned to the *strongly correlated* “source_KP_sc” is highest in the AMTEA. As a consequence, even though the AMTEA is able to receive some positive transfer from the various sources (which explains its performance in comparison to CEA), it is unable to tap the full potential of TrEO. The AMTEA+M_S^L successfully overcomes this limitation by virtue of domain adaptation, stimulating increased positive transfers.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Fig. 5. Convergence trends and learned transfer coefficients for target task “target_KP_sc” (top row), “target_KP_wc” (central row), and “target_KP_uc” (bottom row). The central column depicts the transfer coefficients learned by the AMTEA+M_S^L, whereas the rightmost column depicts the transfer coefficients learned by the baseline AMTEA. All results are averaged over 30 independent runs of each optimizer. The shaded region spans one standard deviation on either side of the mean.

7. Empirical Assessment of Non-Linear Domain Adaptation in TrEO

Here, we experimentally verify the effectiveness of the online learning AMTEA+M_S^{NL} on continuous benchmark functions as well as a practical case study in multi-location inventory planning. Five optimizers are considered for comparison, including state-of-the-art TrEO as well as non-TrEO methods. These are: (i) the *exponential natural* *evolution strategies* (xNES) which is a powerful continuous optimization algorithm for small-to-medium scale problems [45], (ii) CEA without transfer, (iii) CEA+M_S^{NL} which is a non-linear extension of the recently put forward transfer evolutionary algorithm in [28], (iv) the baseline AMTEA without domain adaptation, and (v) our proposed AMTEA+M_S^{NL}. All optimizers are real-coded with a population size of $N = 50$. Apart from the xNES, all other algorithms employ uniform crossovers, Gaussian mutation (with probability 0.04) and tournament selection. We set the transfer interval to $\Delta = 2$, at which stacked density models with Gaussian components are learned in the AMTEA and the AMTEA+M_S^{NL}, whereas transformed source solutions are directly injected into CEA+M_S^{NL}. For xNES, its default parameter settings are used, with the population initialization chosen so as to facilitate fair comparison with its counterparts.

The same non-linear source-to-target mapping, as described in "Online Construction of Non-Linear Source-to-Target Maps", is used in both CEA+M_S^{NL} and AMTEA+M_S^{NL}. In the two-layer feedforward neural network that constitutes M_S^{NL}, we set the number of neurons in the hidden layer as $d_h \longleftarrow 2d_T$. The value of d_h should be set sufficiently large as the randomization of the first layer of the neural mapping function can lead to several redundant features. Any adverse impact of the redundant features is mitigated by the regularization term in Eq. (13). In our implementation, the regularization coefficient is randomized in the range [1, 1000].

7.1. Synthetic Benchmark Example

We choose two synthetic test functions to demonstrate the efficacy of the non-linear domain adaptive AMTEA+M_S^{NL}. The Griewank function [46] is used as a source task, whereas the Sphere function serves as the target task. Both of these are *minimization* problems, and their mathematical expressions are given in Eq. (21) and Eq. (22), respectively. Notice that the first term in the Griewank function is the same as the Sphere, which lends the two functions similar global character (thus giving rise to the scope of knowledge transfer between them). However, while the Griewank function has multiple local optima (due to the second term in Eq. (21)), the Sphere function is unimodal. What is more, in our formulation of Eq. (22), the global optimum of the Sphere function has been shifted to $\{10\}^{\boldsymbol{d_T}}$, while that of the Griewank function is at $\{0\}^{\boldsymbol{d_S}}$. The shifted distribution of good solutions implies $\mathcal{D}_S \neq \mathcal{D}_T$.

$$f_S(\boldsymbol{x}_S) = \frac{1}{4000}\sum_{i=1}^{d_S} x_{S,i}^2 - \prod_{i=1}^{d_S} \cos\left(\frac{x_{S,i}}{\sqrt{i}}\right) + 1, \tag{21}$$

$$f_T(\boldsymbol{x}_T) = \sum_{i=1}^{d_T} \left(x_{T,i} - 10\right)^2. \tag{22}$$

We setup three different single source TrEO examples, namely, (i) $d_T > d_S$, (ii) $d_T = d_S$, and (iii) $d_T < d_S$, using the test functions above. In all three examples we set $d_T = 20$, whereas $d_S = 10, 20$ and 30, respectively. Recall that when $d_T > d_S$, a linear source-to-target mapping does not directly apply due to the occurrence of a degenerate distribution. Even if $d_T = d_S$ (or $\mathcal{X}_S = \mathcal{X}_T$), we have $p_S^*(\boldsymbol{x}_S) \not\approx p_T^*(\boldsymbol{x}_T)$ due to the shifted optimum of the Sphere function, which renders the baseline AMTEA less effective.

A summary of the results returned by the five optimizers (averaged over 30 runs) at the end of 5000 function evaluations is provided in Table 1. In all the cases, the TrEO algorithms (which execute canonical evolutionary operators during non-transfer iterations) are able to surpass the standard CEA by harnessing solutions transferred from the source. The source optimization data was collected by running the CEA on the respective source tasks.

The highlight of Table 1 is that the AMTEA+M_S^{NL} performs significantly better than all the other algorithms. This is attributed to the fact that the AMTEA+M_S^{NL} accesses the complementary benefits of probabilistic model-based transfer and non-linear domain adaptation through M_S^{NL}. The CEA+M_S^{NL} on the other hand does not explicitly guard against negative transfers, which tends to slow down its convergence rate in the presence of inaccurately learned source-to-target transformations. This is evident when CEA+M_S^{NL} is consistently outperformed by xNES, AMTEA and AMTEA+M_S^{NL} in all three examples. Even though xNES performs significantly better than CEA, its solution quality is surpassed by both AMTEA and AMTEA+M_S^{NL}.Comparing the baseline AMTEA with the CEA in Table 1, the former clearly benefits from a boost in performance as a consequence of the source induced search bias. However, the final objective values reached by the AMTEA are far from optimal. In contrast, the scope for non-linear source-to-target mapping in the AMTEA+M_S^{NL} allows it to adapt to source-target domain mismatches, approaching near-optimal solutions more consistently; it's superior convergence trends for the case $d_T > d_S$ are also depicted in Fig. 6.

Table 1.

Experimental results for the shifted Sphere function, at the end of 5000 function evaluations. In bold are the best results with 95% confidence level based on the Wilcoxon signed-rank test. Results are reported based on 30 independent runs of all optimizers.

Algorithms	Target Objective Values

Problem Configuration		Average	Best	Worst	Q_{better} [47]
$d_T > d_S$	xNES	96.22	55.62	140.43	1.0000
	CEA	550.18	122.90	2104.94	1.0000
	CEA+M_S^{NL}	121.36	53.38	231.07	1.0000
	AMTEA	39.60	0.50	215.56	0.9822
	AMTEA+M_S^{NL}	**0.63**	0.01	10.11	-
$d_T = d_S$	xNES	96.22	55.62	140.43	1.0000
	CEA	550.18	122.90	2104.94	1.0000
	CEA+M_S^{NL}	117.44	76.35	167.99	1.0000
	AMTEA	23.31	0.03	72.79	0.7944
	AMTEA+M_S^{NL}	**2.30**	0.01	24.28	-
$d_T < d_S$	xNES	96.22	55.62	140.43	1.0000
	CEA	550.18	122.90	2104.94	1.0000
	CEA+M_S^{NL}	110.70	53.41	182.63	1.0000
	AMTEA	8.94	0.02	68.76	0.8011
	AMTEA+M_S^{NL}	**0.96**	0.00	9.45	-

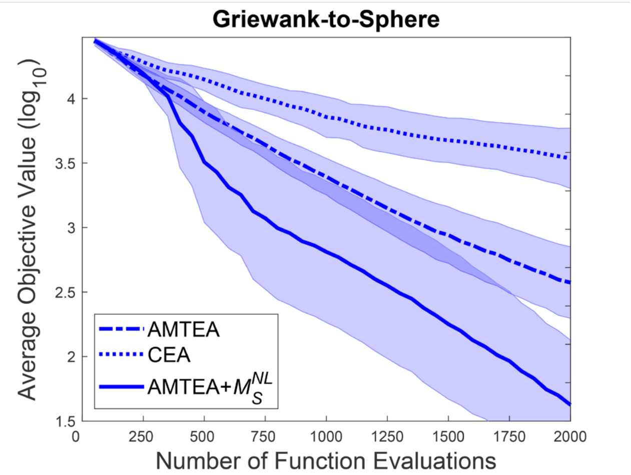


Fig. 6. Convergence trends for the shifted Sphere function when $d_T > d_S$. Source data comes from the multimodal Griewank function.

In addition to the above, we conduct an additional quantitative analysis using a new probabilistic performance comparison metric, denoted herein as Q_{better}; readers are referred to [47] for a detailed description of this metric. Roughly speaking, Q_{better} measures the empirical probability that AMTEA+M_S^{NL} achieves better target objective values relative to a baseline algorithm. The closer Q_{better} is to 1.0, the greater the confidence we have in the superior performance of our proposed approach.

Based on the Q_{better} values in Table 1, we can claim that AMTEA+M_S^{NL} performs better than xNES, CEA and CEA+M_S^{NL} with high confidence across the 30 independent runs. When considering the Q_{better} values relative to AMTEA, the $d_T > d_S$ example gives the largest probability of 98.2% that AMTEA is outperformed by our algorithm. Furthermore, Table 1 shows that AMTEA+M_S^{NL}'s best and worst objective values are consistently smaller than those of the other algorithms. These outcomes strongly support the utility of the non-linear source-to-target mappings learned in AMTEA+M_S^{NL} for reducing the degree of source-target domain mismatch. Our results also suggest that the domain adaptation strategy in AMTEA+M_S^{NL} is able to uncover hidden inter-task relationships, which leads to enhanced target convergence rate in accordance with the discussions in "Motivating Domain Adaptations in TrEO".

7.2. A Case-Study in Multi-Location Inventory Planning

Next, we consider a practical application of our proposed algorithm for multi-location inventory planning under demand uncertainty. The source-target heterogeneity in this domain stems from the fact that the number of locations, as well as demand uncertainty patterns, could frequently change over time.

In complex supply chains, inventory redistribution has long been studied to cope with the imbalance between realized demands and inventory levels. In this study, we account for instantaneous lateral transshipments for inventory redistribution in a multi-location inventory system consisting of a single distribution centre and multiple retailers. At every decision-making step, the optimal order-up-to replenishment level at each retailer is to be determined so that the total expected cost (which includes the inventory holding cost c_h, shortage / backlogging cost c_s, and lateral transshipment cost c_{lt}) is minimized. Lateral transshipments occur when the shortage of inventory at one retailer can be counterbalanced by a surplus of inventory at another retailer. A detailed description and mathematical model of the problem can be found in [48]. Briefly, the cost minimization program can be stated as,

$$\min_{\boldsymbol{x}} \ \mathrm{E}_{D \sim p(D)}[(c_h + c_s + c_{lt}) | \boldsymbol{x}, D], \tag{25}$$

where $\boldsymbol{x} \in \mathbb{R}^d_{\geq 0}$ is the order-up-to replenishment level at each of d retailers, and $p(D)$ is the uncertain demand distribution with mean $\boldsymbol{\mu}$ and variance $\sum$.

The multi-location inventory planning problem gives rise to a range of possible scenarios in which the utility of transfer optimization algorithms can be tested. In the experiments that follow, we consider three contrasting scenarios. (i) In the simplest case, the number of retailers in the source and the target optimization tasks are the same (i.e., $d_T = d_S$) and the expected demands at the retailers do not change over time (we denote this as $\boldsymbol{\mu}_T = \boldsymbol{\mu}_S$). (ii) In the second case, while the number of retailers remains the same (i.e., $d_T = d_S$), the expected demands at the retailers differ in the source and target optimization tasks (i.e., $\boldsymbol{\mu}_T \neq \boldsymbol{\mu}_S$); this could reflect a practical scenario where different items sold by the same retailers face different demand patterns. (iii) Finally, we consider a case where the number of retailers have increased over time (i.e., $d_T > d_S$) which is accompanied by a change in the expected demands at the retailers (i.e., $\boldsymbol{\mu}_T \neq \boldsymbol{\mu}_S$). Recall that when $d_T > d_S$, a linear source-to-target mapping does not directly apply in conjunction with probabilistic formulations of TrEO.

The three scenarios described above can accordingly be summarized as, *Case 1:* $d_T = d_S$ and $\boldsymbol{\mu}_T = \boldsymbol{\mu}_S$, *Case 2:* $d_T = d_S$ but $\boldsymbol{\mu}_T \neq \boldsymbol{\mu}_S$, and *Case 3:* $d_T > d_S$ and $\boldsymbol{\mu}_T \neq \boldsymbol{\mu}_S$. In all cases we set $d_T = 20$, whereas $d_S = 10$ (when $d_T > d_S$) or 20 (when $d_T = d_S$).

7.2.1. Experimental Specifications

In our implementation of the multi-location inventory planner, given any candidate solution $\boldsymbol{x}_T \in \mathbb{R}_{\geq \mathbf{0}}^{d_T}$ in an EA, nested linear programs (LPs) must be solved (using an off the shelf LP solver) to obtain corresponding optimal lateral transshipments under various demands realizations sampled from $p(D_T)$ [48]. Thus the fitness evaluation of every candidate solution is quite expensive, to a degree that the marginal computational overhead in learning the non-linear source-to-target map M_S^{NL} and subsequent stacked density estimation in the AMTEA+M_S^{NL} is negligible.

7.2.2. Results

A summary of the results returned by the five optimizers (averaged over 30 runs) at the end of 3000 function evaluations is provided in Table 2. As in previous sections, we find that the various TrEO algorithms are able to outperform the standard CEA by exploiting the search bias obtained from the source optimization data. The source data used for this purpose was collected by running the CEA on the respective source tasks. In two out of the three cases, the AMTEA+M_S^{NL} was found to be significantly better than all its counterparts (with 95% confidence based on the Wilcoxon singed-rank test). Here although xNES achieves a better target objective value than CEA, its solution quality is outperformed by all TrEO algorithms.

Table 2.

Experimental results for the target multi-location inventory planning task, at the end of 3000 function evalutations. In bold are the best results with 95% confidence level based on the Wilcoxon signed-rank test. Results are reported based on 30 independent runs of all optimizers.

Problem Configuration	Algorithms	Target Objective Values			Q_{better} [47]
		Average	Best	Worst	
$d_T = d_S$ $(\boldsymbol{\mu}_T = \boldsymbol{\mu}_S)$	xNES	6.10 e3	4.93 e3	6.95 e3	0.9978
	CEA	6.96 e3	4.88 e3	10.35 e3	0.9978
	CEA+M_S^{NL}	4.18 e3	3.49 e3	4.99 e3	0.8544
	AMTEA	3.45 e3	2.38 e3	4.68 e3	0.5467
	AMTEA+M_S^{NL}	3.37 e3	2.51 e3	5.15 e3	-
$d_T = d_S$ $(\boldsymbol{\mu}_T \neq \boldsymbol{\mu}_S)$	xNES	6.10 e3	4.93 e3	6.95 e3	1.0000
	CEA	6.96 e3	4.88 e3	10.35 e3	1.0000
	CEA+M_S^{NL}	4.15 e3	3.44 e3	5.31 e3	0.8222
	AMTEA	4.19 e3	3.69 e3	5.05 e3	0.8533
	AMTEA+M_S^{NL}	**3.53 e3**	2.66 e3	4.81 e3	-
$d_T > d_S$ $(\boldsymbol{\mu}_T \neq \boldsymbol{\mu}_S)$	xNES	6.10 e3	4.93 e3	6.95 e3	0.9978
	CEA	6.96 e3	4.88 e3	10.35 e3	0.9967
	CEA+M_S^{NL}	4.08 e3	3.18 e3	6.20 e3	0.8833
	AMTEA	4.06 e3	3.55 e3	5.29 e3	0.8511
	AMTEA+M_S^{NL}	**3.45 e3**	2.57 e3	5.32 e3	-

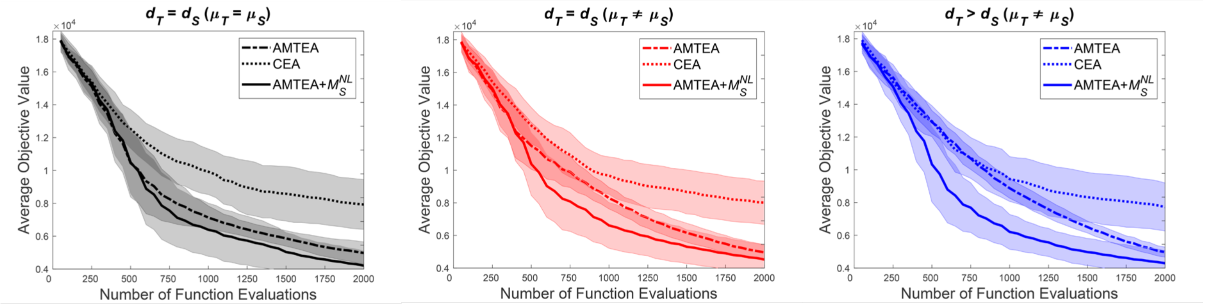
For *Case 1*, where $d_T = d_S$ and $\boldsymbol{\mu}_T = \boldsymbol{\mu}_S$, AMTEA performs competitively with the AMTEA+M_S^{NL} as there is little difference between the source and target domains; thus, there is little to be gained by the additional non-linear domain adaptation step. However, as the source and target domain mismatch gradually increases from *Case 2* (where$d_T = d_S$ but $\boldsymbol{\mu}_T \neq \boldsymbol{\mu}_S$) to *Case 3* (where$d_T > d_S$ and $\boldsymbol{\mu}_T \neq \boldsymbol{\mu}_S$), the performance boost achieved by the AMTEA+M_S^{NL} becomes increasingly apparent; see Fig. 7 and Fig. 8.

Comparisons of convergence trends for the target task are presented in Fig. 7. The corresponding performance gap between AMTEA+M_S^{NL} and AMTEA is depicted in Fig. 8. One of the key highlights of Fig. 7 is the transfer coefficients learned by the two algorithms. In Fig. 7(d) for *Case 1*, the coefficient values are somewhat similar. In contrast, in Fig. 7(e), (f) for *Case 2* and *Case 3*, respectively, AMTEA+M_S^{NL} assigns significantly higher transfer coefficient values to the source probabilistic model. This observation clearly demonstrates the utility of domain adaptations (under increasing source-target domain mismatch) for stimulating positive transfers. As elaborated in “Motivating Domain Adaptations in TrEO”, the observed outcome is achieved by the transformed source representation of Eq. (14) which successfully uncovers stronger inter-task relationships that remain hidden in the problem’s originally defined input feature space. Accordingly, greater source-target domain overlap is induced, and thereby, allowing larger transfer coefficient values being learned in AMTEA+M_S^{NL}. On the other hand, narrow focus on the curbing of negative transfers in the baseline AMTEA causes the conservative attenuation of knowledge transfers – as is reflected by the much lower transfer coefficient values that the algorithm learns.

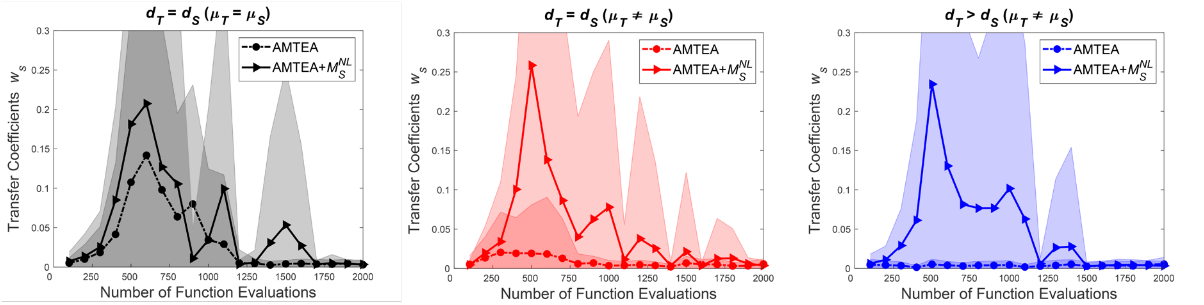
8. Conclusion

This paper is devoted to unveiling the concept of and establishing a probabilistic formulation for *domain adaptation* in the context of transfer evolutionary optimization (TrEO). Leveraging on the cognitive underpinnings of memetic computation, TrEO marks an emerging paradigm in computational intelligence that ventures to achieve human-like problem-solving prowess by autonomously transferring and reusing knowledge from experiential (*source*) tasks to improve optimization efficiency in a related *target* task of interest. In particular, domain adaptation is seen as an approach to promote effective knowledge transfers even in situations of source-target domain mismatch by offering a favorable balance between the curbing of negative transfers and the active stimulation of positive transfers. This is achieved by transforming the search spaces of the source and target tasks (of possibly differing dimensionality) to a common representation space, with the objective of inducing greater overlap between their respective optimized search distributions in that space. Depending on the type of spatial transformation, three categories of domain adaptation have been derived, namely, (i) *source-to-target*, (ii) *target-to-source*, and (iii) *source-cum-target* maps.

In the mathematical descriptions of the aforementioned categories, it was revealed that non-linear source-to-target maps offer a more generalizable form of domain adaptation (as opposed to their linear counterpart) that can cope with arbitrary discrepancies between source and target search space dimensionality. Accordingly, a novel domain adaptive transfer evolutionary algorithm referred to as the AMTEA+M_S was proposed, with its non-linear variant labelled as the AMTEA+M_S^{NL}. Here, M_S and M_S^{NL} represent the source-to-target mapping function. Experimental studies for hand-crafted (offline) as well as online data-driven domain adaptations in TrEO were carried out, spanning the 0-1 knapsack problem, synthetic test functions, and examples in multi-location inventory planning. The results confirmed the efficacy of our proposal in terms of boosting optimization performance, even relative to existing state-of-the-art TrEO algorithms.



(a) **(b)** **(c)**



(d) (e) (f)

Fig. 7. Convergence trends for the target multi-location inventory planning task and the transfer coefficients learned by the AMTEA and the AMTEA+M_S^{NL} in *Case 1:* $d_T = d_S$ and $\mu_T = \mu_S$ (left column), *Case 2:* $d_T = d_S$ but $\mu_T \neq \mu_S$ (central column), and *Case 3:* $d_T > d_S$ and $\mu_T \neq \mu_S$ (rightmost column). The shaded region spans one standard deviation on either side of the mean (based on 30 independent runs).

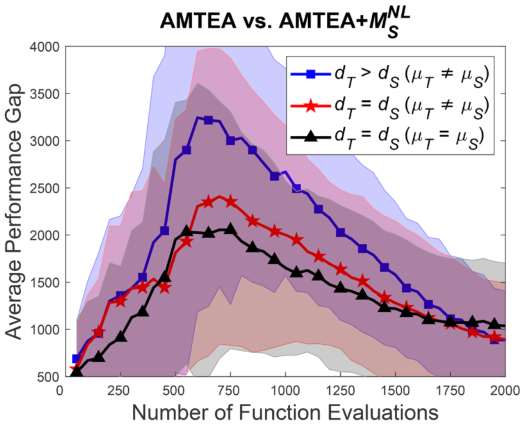


Fig. 8. Performance gap between the AMTEA+M_S^{NL} and the AMTEA on the multi-location inventory cost minimization problem. The gap is measured by subtracting the objective values achieved by AMTEA+M_S^{NL} from those achieved by AMTEA in Fig. 7. The plots show that the impact of the AMTEA+M_S^{NL} is more significant as source-target domain mismatch increases.

In the future, we intend to expand the applicability of our methods to a wider array of problems. In particular, we shall delve deeper into combinatorial optimization problems that are characterized by a rich diversity of feature spaces and representations. While the discrete 0-1 knapsack problem has been studied in this paper, a plethora of other problems remain to be explored (potentially demanding new domain adaptation techniques). Of significant interest are problems characterized by computationally expensive (simulation-based) function evaluations, with recent advances in transfer Bayesian optimization algorithms [49] providing some initial pathways to tackle source-target heterogeneity in such cases as well. We shall also investigate the effectiveness of the AMTEA+M_S^{NL} when applied to multi-objective optimization settings.

Compliance with Ethical Standards

Funding Information: This work is funded by the Agency for Science, Technology and Research (A*STAR) of Singapore, under the Singapore Institute of Manufacturing Technology-Nayang Technological University Collaborative Research Programme in Complex Systems.

Ethical Approval: This article does not contain any studies with human participants or animals performed by any of the authors.

References

[1] Gupta A, Ong YS. Back to the roots: Multi-x evolutionary computation. Cogn Comput. 2019;11(1):1–17.

[2] Mahale RA, Chavan SD. A survey: evolutionary and swarm based bio-inspired optimization algorithms. International Journal of Scientific and Research Publications. 2012;2(12):1–6.

[3] Xu W, Xu JX, He D, Tan KC. An Evolutionary Constraint-Handling Technique for Parametric Optimization of a Cancer Immunotherapy Model. IEEE Trans Emerg Topics Comput Intell. 2019;3(2):151-62.

[4] Jin Y, Wang H, Chugh T, Guo D, Miettinen K. Data-driven evolutionary optimization: an overview and case studies. IEEE Trans Evol Comput. 2019;23(3):442–458.

[5] Eremeev AV, Kovalenko YV. A memetic algorithm with optimal recombination for the asymmetric travelling salesman problem. Memetic Computing. 2020;12(1):23-36.

[6] Gupta A, Ong YS. Memetic Computation: The Mainspring of Knowledge Transfer in a Data-Driven Optimization Era. Vol 21. Springer; 2018.

[7] Gupta A, Ong YS, Feng L. Insights on transfer optimization: Because experience is the best teacher. IEEE Trans Emerg Topics Comput Intell. 2018;2(1):51–64.

[8] Kaedi M, Ghasem-Aghaee N. Biasing Bayesian optimization algorithm using case based reasoning. Knowl Based Syst. 2011;24(8):1245–1253.

[9] Pelikan M, Hauschild MW, Lanzi PL. Transfer learning, soft distance-based bias, and the hierarchical boa. In Proceedings of the International Conference on Parallel Problem Solving in Nature 2012 Sep 1 (pp. 173–183). ACM.

[10] Feng L, Ong YS, Tan AH, Tsang IW. Memes as building blocks: A case study on evolutionary optimization+

transfer learning for routing problems. Memet Comput. 2015;7(3):159–180.

[11] Feng L, Ong YS, Lim MH, Tsang IW. Memetic search with inter-domain learning: A realization between CVRP and CARP. IEEE Trans Evol Comput. 2015;19(5):644–658.

[12] Iqbal M, Browne WN, Zhang M. Extracting and using building blocks of knowledge in learning classifier systems. In Proceedings of the Genetetic and Evolutionary Computation Conference 2012 Jul 7 (pp. 863–870). ACM

[13] Iqbal M, Browne WN, Zhang M. Reusing building blocks of extracted knowledge to solve complex, large-scale Boolean problems. IEEE Trans Evol Comput. 2014;18(4):465–580.

[14] Gupta A, Ong YS, Feng L. Multifactorial evolution: toward evolutionary multitasking. IEEE Trans Evol Comput. 2016;20(3):343–357.

[15] Ong YS, Gupta A. Evolutionary multitasking: A computer science view of cognitive multitasking. Cogn Comput. 2016;8(2):125–142.

[16] Liaw RT, Ting CK. Evolutionary manytasking optimization based on symbiosis in biocoenosis. In Proceedings of the AAAI Conference on Artificial Intelligence 2019 Jan 27 (pp. 4295–4303). ACM.

[17] Tang J, Chen Y, Deng Z, Xiang Y, Joy CP. A Group-based Approach to Improve Multifactorial Evolutionary Algorithm. In the International Joint Conferences on Artificial Intelligence IJCAI 2018 Jul 13 (pp. 3870–3876). ACM.

[18] Zheng X, Qin AK, Gong M, Zhou D. Self-regulated evolutionary multi-task optimization. IEEE Trans Evol Comput. 2020;24(1):16–28.

[19] Min ATW, Ong YS, Gupta A, Goh CK. Multiproblem surrogates: transfer evolutionary multiobjective optimization of computationally expensive problems. IEEE Trans Evol Comput. 2019;23(1):15–28.

[20] Yang C, Ding J, Jin Y, Wang C, Chai T. Multitasking multiobjective evolutionary operational indices optimization of beneficiation processes. IEEE Trans Autom Sci Eng. 2019;16(3):1046–1057.

[21] Ardeh MA, Mei Y, Zhang M. A Novel Genetic Programming Algorithm with Knowledge Transfer for Uncertain Capacitated Arc Routing Problem. In Pacific Rim International Conference on Artificial Intelligence 2019 Aug 26 (pp. 196–200). Springer.

[22] Moshaiov A, Tal A. Family bootstrapping: A genetic transfer learning approach for onsetting the evolution for a set of related robotic tasks. In IEEE Congress on Evolutionary Computation 2014 July 6 (pp. 2801–2808). IEEE.

[23] Sagarna R, Ong YS. Concurrently searching branches in software tests generation through multitask

evolution.In IEEE Symposium Series on Computational Intelligence (SSCI) 2016 Dec 2016 (pp. 1–8). IEEE.

[24] Louis SJ, McDonnell J. Learning with case-injected genetic algorithms. IEEE Trans Evol Comput. 2004;8(4):316–328.

[25] Smyth P, Wolpert D. Stacked density estimation. In Advances in Neural Information Processing Systems (NIPS) 1998 (pp. 668–674). ACM.

[26] Da B, Gupta A, Ong YS. Curbing Negative Influences Online for Seamless Transfer Evolutionary Optimization. IEEE Trans Cybern. 2018;49(12):4365–4378.

[27] Zhou L, Feng L, Zhong J, Zhu Z, Da B, Wu. A study of similarity measure between tasks for multifactorial evolutionary algorithm. In Proceedings of the GECCO Companion 2018 Jul 15 (pp. 229–230). ACM.

[28] Feng L, Ong YS, Jiang S, Gupta A. Autoencoding evolutionary search with learning across heterogeneous problems. IEEE Trans Evol Comput. 2017;21(5):760–772.

[29] Bali KK, Gupta A, Feng L, Ong YS, Tan PS. Linearized domain adaptation in evolutionary multitasking. In IEEE Congress on Evolutionary Computation 2017 Jun 5 (pp. 1295–1302). IEEE.

[30] Feng L, Zhou L, Zhong J, Gupta A, Ong YS, Tan KC, et al. Evolutionary multitasking via explicit autoencoding. IEEE Trans Cybern. 2019;49(9):3457–3470.

[31] Pan SJ, Yang Q. A survey on transfer learning. IEEE Trans Knowl Data Eng. 2010;22(10):1345–1359.

[32] Shakeri M, Gupta A, Ong YS, Xu C, Zhang AN. Coping with Big Data in Transfer Optimization. In IEEE International Conference on Big Data 2019 Dec 9 (pp. 3925–3932). IEEE.

[33] Kodirov E, Xiang T, Gong S. Semantic autoencoder for zero-shot learning. In IEEE Conference on Computer Vision and Pattern Recognition 2017 Jul 21 (pp. 3174–3183). IEEE.

[34] Deng WY, Lendasse A, Ong YS, Tsang IW, Chen L, Zheng QH. Domain adaption via feature selection on explicit feature map. IEEE Trans Neural Netw Learn Syst. 2019;30(4):1180–1190.

[35] Sun S, Shi H, Wu Y. A survey of multi-source domain adaptation. Information Fusion. 2015;24:84–92.

[36] Krejca MS, Witt C. Theory of estimation-of-distribution algorithms. Theory of Evolutionary Computation. Cham, Switzerland: Springer; 2020. pp. 405–442.

[37] Zhang Q, Muhlenbein H. On the convergence of a class of estimation of distribution algorithms. IEEE Trans Evol Comput. 2004;8(2):127–136.

[38] Baluja S, Davies S. Fast probabilistic modeling for combinatorial optimization. In Proceedings of the AAAI Conference on Artificial Intelligence 1998 Jul 26 (pp. 469–476). ACM.

[39] Liang NY, Huang GB, Saratchandran P, Sundararajan N. A fast and accurate online sequential learning

algorithm for feedforward networks. IEEE Trans Neural Netw. 2006;17(6):1411–1423.

[40] Hendeby G, Gustafsson F. On nonlinear transformations of Gaussian distributions. Technical Report from Automatic Control, Linkoping University, Sweden, 2007.

[41] Feng L, Gupta A, Ong YS. Compressed representation for higher-level meme space evolution: a case study on big knapsack problems. Memet Comput. 2019;11(1):3–17.

[42] Dantzig GB. Discrete-Variable Extremum Problems. Oper Res. 1957;5(2):266–288.

[43] Michalewicz Z, Arabas J. Genetic algorithms for the 0/1 knapsack problem. In Proceedings of the International Symposium on Methodologies for Intelligent Systems 1994 Oct 16 (pp. 134–143). Springer.

[44] Mühlenbein H. The equation for response to selection and its use for prediction. Evol Comput. 1997;5(3):303–346.

[45] Wierstra D, Schaul T, Glasmachers T, Sun Y, Peters J, Schmidhuber J. Natural evolution strategies. The Journal of Machine Learning Research. 2014;15(1):949–980.

[46] Locatelli M. A note on the Griewank test function. J Glob Optim. 2003;25(2):169–174.

[47] Gomes WJ, Beck AT, Lopez RH, Miguel LF. A probabilistic metric for comparing metaheuristic optimization algorithms. Structural Safety. 2018;70:59–70.

[48] Chartniyom S, Lee MK, Luong L, Marian R. Multi-location inventory system with lateral transshipments and emergency orders. In IEEE International Conference on Industrial Engineering and Engineering Management 2007 Dec 2 (pp. 1594–1598). IEEE.

[49] Min ATW, Gupta A, Ong YS. Generalizing Transfer Bayesian Optimization to Source-Target Heterogeneity. IEEE Transactions on Automation Science and Engineering. 2020. In Press.