Curbing Negative Influences Online for Seamless Transfer Evolutionary Optimization

Bingshui Da, Abhishek Gupta, and Yew-Soon Ong, *Fellow, IEEE*

***Abstract*—This paper draws motivation from the remarkable ability of humans to extract useful building-blocks of knowledge from past experiences and spontaneously reuse them for new and more challenging tasks. It is contended that successfully replicating such capabilities in computational solvers, particularly global black-box optimizers, can lead to significant performance enhancements over the current state-of-the-art. The main challenge to overcome is that in general black-box settings, no problem-specific data may be available prior to the onset of the search, thereby limiting the possibility of offline measurement of the synergy between problems. In light of the above, this paper introduces a novel evolutionary computation framework that enables *online* learning and exploitation of similarities across optimization problems, with the goal of achieving an algorithmic realization of the *transfer optimization* paradigm. One of the salient features of our proposal is that it accounts for latent similarities which while being less apparent on the surface, may be gradually revealed during the course of the evolutionary search. A theoretical analysis of our proposed framework is carried out, substantiating its positive influences on optimization performance. Furthermore, the practical efficacy of an instantiation of an adaptive transfer evolutionary algorithm is demonstrated on a series of numerical examples, spanning discrete, continuous, as well as single- and multi-objective optimization.**

***Index Terms*—Negative transfer, online similarity learning, probabilistic models, transfer evolutionary optimization.**

I. Introduction

In THE global black-box optimization literature, efforts have seldom been made to automate the reuse of knowledge acquired from past problem-solving experiences. This limitation is primarily due to the lack of problem-specific data available prior to the onset of the search, which makes it difficult to ascertain (offline) the relationships across problems. In contrast, the idea of taking advantage of *available data* from various *source* tasks to improve the learning of a related *target* task has achieved significant success in the field of machine learning—under the label of *transfer learning* [1]–[5]. With this in mind, and under the observation that optimization problems of practical interest seldom exist in isolation [6], our goal in this paper is to achieve an algorithmic realization of the novel concept of *transfer optimization* [7]. Our contributions are expected to benefit real-world optimization settings of a time-sensitive nature, *where ignoring prior experience implies the waste of a rich pool of knowledge that can otherwise be exploited to facilitate efficient re-exploration of possibly overlapping search spaces*.

Any practical useful system, especially those in industrial settings, must be expected to tackle many related problems over a lifetime. Many of these problems will either be repetitive or at least share some domain-specific similarities. Thus, given the present-day demands for achieving high-quality solutions within strict time constraints, the need to effectively exploit the knowledge learned from past experiences is well recognized [8]. Indeed, it is the ability to take advantage of acquired domain knowledge that sets apart an expert from a novice. Despite the significant advances made in this direction by the machine learning community, it is noted that related progress in the field of *optimization* has been relatively scarce. Out of the handful of efforts that have emerged over the years, the following have been identified as common approaches for knowledge reuse: 1) storing a pool of high quality solutions and/or learned models from various source problems that are subsequently *injected* to bias the search on the target optimization task [6], [9]–[12] and 2) learn iterative mappings that transform stored solutions from previously solved optimization problems to candidate solutions in the ongoing target problem of interest [13].

Based on an initial study, we find that the aforementioned approaches typically make use of large databases to store past solutions. Thereafter, the *case by case assessment* required to select the most appropriate candidate solutions gradually becomes prohibitively time consuming as the database grows in size; often referred to as the *swamping problem* [14], [15]. Challenges are particularly exacerbated by the fact that little can be said *a priori*, with any degree of certainty, about the *similarity* between black-box optimization problems (due to the lack of target data available before the onset of the search). *As a result, the threat of negative transfer acts as a major impedance when dealing with multiple sources of knowledge, where some sources may be more relevant than others.*

Manuscript received January 12, 2018; revised May 22, 2018; accepted August 2, 2018. Date of publication August 27, 2018; date of current version September 5, 2019. This work was supported in part by the National Research Foundation of Singapore and in part by SAP. This paper was recommended by Associate Editor M. Zhang. *(Corresponding author: Bingshui Da.)*

B. Da is with the School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798 (e-mail: da0002UI@e.ntu.edu.sg).

A. Gupta is with the Data Science and Artificial Intelligence Research Center, School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798 (e-mail: abhishekg@ntu.edu.sg).

Y.-S. Ong is with the Data Science and Artificial Intelligence Research Center, School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798, and also with the School of Computer Science Engineering, University of New South Wales, Sydney, NSW 2052, Australia (e-mail: asysong@ntu.edu.sg).

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

Digital Object Identifier 10.1109/TCYB.2018.2864345

In contrast, we humans can effortlessly draw useful information from a vast pool of knowledge acquired over a lifetime of experiences. Interestingly, even if two distinct tasks appear unrelated on the surface, humans are often able to uncover any latent synergies that may be present.

The observations above serve as the main motivations behind the present paper. With the goal of incorporating human-like problem-solving capabilities into optimization solvers, we propose a novel *transfer evolutionary computation* paradigm capable of *online source–target similarity learning* as a way to curb the risks of negative transfer on the fly. Given the rapid advancements in modern computing platforms such as the cloud and the Internet of Things (IoT), which give rise to large-scale data storage and seamless communication facilities, the practical viability of such a paradigm (from a hardware perspective) is little in doubt. Thus, we focus on addressing the shortcomings that continue to exist in terms of developing suitable algorithms. In particular, this paper focuses on the use of population-based *evolutionary algorithms* (EAs) as they not only provide significant flexibility in dealing with a wide range of discrete and continuous optimization problems, but are also amenable to be hybridized with various learning strategies. In our approach, we capture the population distribution of *elite* solutions from some source optimization task in the form of a probabilistic model, that is then stored for future usage. These probabilistic knowledge building-blocks serve to bias the search on a *related* target task toward solutions that have been shown to be promising. Importantly, for knowledge transfer to occur in this manner, the probabilistic models must be defined in an all-encompassing *universal search space*, which creates a *common (shared) platform* for the exchange of ideas to take place [16]. The synthesis of diverse knowledge building-blocks is then realized by sampling candidate solutions from a *mixture of source + target probabilistic models*, with the mixture coefficients calculated in a manner that provides theoretical performance guarantees (see Section V for details).

At this juncture, it is worthwhile to mention that the techniques proposed in this paper can be implemented within any EA of ones preference, as a way of endowing it with online knowledge transfer capabilities. Attention of the reader is primarily drawn toward our demonstrations of the explicit capture of source–target similarities for a generic transfer evolutionary optimizer, that, in principle, can be applied to any optimization problem. As an aside, we deem our framework to fall within the realm of *memetic computing* [17], with the *memes* (popularly defined as *computationally encoded units of knowledge for improved problem-solving* [18]) herein taking the form of probabilistic models of elite solution distributions. In this regard, as our proposal is based on the transfer of learned models across problems, it is hereafter labeled as *adaptive model-based transfer* (AMT).

To summarize, the main contributions of this paper are as follows.

1) A novel model-based transfer EA that is capable of online learning and exploitation of similarities across black-box optimization problems, in a manner that minimizes the threat of negative transfer.
2) Theoretical analysis of the proposed approach, demonstrating that the knowledge-enhancement scheme guarantees to facilitate global convergence of the EA.
3) Rigorous experimental verification of the algorithm on a diverse test suite.

The remainder of this paper is organized as follows. In Section II, we first present a brief review of related works in the literature. In Section III, the key ingredients that form the crux of the AMT framework are introduced. In Section IV, we present an instantiation of a transfer evolutionary optimization algorithm incorporating our online similarity learning strategy. We label this algorithm as AMT-*enabled* EA (AMTEA). The theoretical foundations and justifications of the framework are then analyzed in Section V. In Section VI, we present the numerical results demonstrating the efficacy of the AMTEA across a range of problems spanning discrete, continuous, single-, and multi-objective optimization (MOO).

II. Background and Related Work

On one side, transfer learning deals with exploiting knowledge from related source tasks to improve the *predictive modeling* of a latent target function [2]. In contrast, transfer optimization is concerned with harnessing past *search experiences* to enhance convergence efficiency toward the global optimum of a possibly black-box target reward/objective function [7].

Notably, significant accomplishments using transfer learning have been achieved in the domain of predictive analytics [2], [3], [19]. It has been found to not only accelerate the speed of learning but also to improve the generalization performance of the target predictive model [1], [20]. In contrast, relatively little progress has been seen in the optimization literature, where the reuse of knowledge extracted from source optimization problems is seldom automated. Initial efforts in the context of transfer optimization have, however, explored a direct solution injection scheme. For example, Cunningham and Smyth [9] proposed to directly inject known good schedules into target scheduling problems. Similarly, a family bootstrapping approach was put forward in [21] for neuro-evolutionary robot controller design, where the optimized solutions of a common source task were used to bias the initial population of target tasks. Likewise, Louis and McDonnel [6] showcased the benefits of periodically injecting a certain number of solutions drawn from intermediate populations of related source optimization problems into the target evolutionary search.

In addition to direct *genetic transfer* from source to target evolutionary searches [22], higher-order model-based transfer algorithms have also appeared in the literature. In [10], a heuristic criterion was used within a specific class of combinatorial problems for retrieving related source probabilistic models from a database to bias the target optimization search. In [12] and [23], the model took the form of a positive semidefinite matrix that induces a modified (biased) distance metric for graph-based clustering cum sequencing problems. More recently, Feng *et al.* [13] proposed to learn a mapping from a continuous source domain to the target search space through a single layer denoising autoencoder. In addition to the above, Iqbal *et al.* [24], [25] and Alvarez *et al.* [26] demonstrated that building blocks of knowledge in the form of code fragments (i.e., trees or subtrees of computer programs evolved by genetic programming), that were extracted from small-scale problems, could be reused while solving more complex, large-scale problems.

While there has been growing interest in the general idea of transfer evolutionary optimization, the majority of existing methods tend to rely on the prior assumption that there must exist some form of exploitable synergy between source and target tasks. This gives rise to the danger of negative transfer, since for a range of real-world problems, little can be said beforehand about source–target similarities. In order to prevent harmful knowledge incorporation, there exist a handful of methods that attempt to infer the synergy between problems offline [12], [23]. However, most of these approaches cater to a specific domain, and further, are not equipped to exploit underlying synergies that may not be apparent on the surface. Much like the salient feature of human-like problem-solving, the distinguishing facet of this paper lies in the online learning and exploitation of latent similarities between distinct optimization problems, thereby opening doors to tapping the full potential of automated knowledge transfer in EAs.

In addition to evolutionary computation, it is also worth noting the recent strides taken in Bayesian optimization toward incorporating knowledge transfer. For example, Swersky *et al.* [27] have shown that multitask Gaussian processes [28] can be used in Bayesian optimization to significantly speed up search in comparison to classical approaches that start the search from scratch. Recently, it was also demonstrated that a principled stacking of multiple Gaussian process models, drawn from different optimization exercises [29], could serve to accelerate search in computationally expensive settings. Despite some success stories, a well-known drawback of Bayesian optimization is that it does not scale well to high dimensional problems, primarily due to the large function evaluation data needed to learn sufficiently informative Gaussian process models [30]. On the other hand, evolutionary methods have been shown to scale well with increasing dimensionality [7]. Furthermore, EAs are largely agnostic to the representation scheme of the underlying optimization problem, while in the Bayesian optimization literature, little progress has been reported in this regard due to the difficulties in coping with indefinite kernels under combinatorial representations [31]. Thus, the advantages of the proposed transfer evolutionary optimization paradigm, in comparison to advances in Bayesian optimization, are not hard to comprehend.

III. Basics of Adaptive Model-Based Transfer

We consider a case where there are $K-1$ previously tackled source optimization tasks, labeled as $\mathcal{T}_1, \ldots, \mathcal{T}_{K-1}$, and a target task $\mathcal{T}_K$ of current interest. Each task could either have a single or multiple objectives. In the former case, we denote the objective function of the kth task as f_k. In this paper, we consider the building-blocks of knowledge extracted from source tasks to take the form of probabilistic models of optimized search distributions, that are subsequently used to bias the search on the target. Specifically, a model φ_k drawn from a maximization task $\mathcal{T}_k$ satisfies:

$$\int f_k(\mathbf{x})\varphi_k(\mathbf{x})d\mathbf{x} \geq f_k^* - \epsilon_k \tag{1}$$

where (*) represents the global optimum, and $\epsilon_k(> 0)$ is a small convergence tolerance threshold.

A favorable aspect of probabilistic models is their relatively small memory footprint. As an example, consider the case of a source task for which a large number (say N) of solutions have been evolved, each consisting of B binary bits. Naively storing the raw solution data consumes NB bits of memory. In contrast, a factored Bernoulli distribution can represent higher-order knowledge about the underlying distribution of the same population of solutions while consuming only $\mathcal{O}(B \log_2 N)$ bits of memory [32].

For the purpose of facilitating seamless transfer of knowledge building-blocks across problems, two ingredients form the crux of the proposed AMT framework: 1) a *universal search space* and 2) mixture modeling through *stacked density estimation* [33]. These are discussed in the next section.

A. Universal Search Space

Simply stated, a universal search space $\mathcal{X}$ serves as a common (unified) platform bringing together the individual search spaces of distinct optimization tasks. The unification is the key element that enables probabilistic models drawn from different source tasks to be directly brought to bear on the target task $\mathcal{T}_K$. To elaborate, consider the following optimization problem reformulation of $\mathcal{T}_K$ in terms of the search distribution:

$$\mathcal{T}_K : \max_{p(\mathbf{x})} \int_{\mathcal{X}} f_K(\mathbf{x})p(\mathbf{x})d\mathbf{x}. \tag{2}$$

In the above, by approximating the latent distribution $p(\mathbf{x})$ as

$$p(\mathbf{x}) \approx \sum_{k=1}^{K} \alpha_k \varphi_k(\mathbf{x}) \tag{3}$$

where $\sum_{k=1}^{K} \alpha_k = 1$ and $\alpha_k \geq 0$ for $k = 1, \ldots, K$, the source models are activated to influence the target search. Note that the mixture weights (α_k's) are tunable, enabling us to adapt the extent of influence.

With this in mind, the universal search space $\mathcal{X}$ is described so as to *encode* solutions to *all* (source + target) optimization problems, such that all probabilistic models $\varphi_1, \ldots, \varphi_K$ can be built in this space. To avoid abuse of notations, $f_1, \ldots, f_K$ are considered to be defined in the universal space.

While dealing only with continuous optimization problems, a viable unification procedure is to linearly scale each variable to the common range of $[0, 1]$. Further, in [16], [34], and [35], it was shown that by using an associated *random-key encoding scheme* [36], it becomes possible to unify discrete and continuous optimization problems as well, with provisions for handling search spaces of differing dimensionality. As an illustration, consider K distinct optimization problems $\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_K$ with search space dimensionality

$d_1, d_2, \ldots, d_K$, respectively. In such a scenario, a unified space $\mathcal{X}$ of dimensionality $d_{\text{unified}} = \max\{d_1, d_2, \ldots, d_K\}$ is defined, so that candidate solutions with respect to all optimization problems can be encoded in $\mathcal{X}$. Thereafter, while addressing the kth optimization problem, a subset of d_k variables are extracted from a candidate solution vector in $\mathcal{X}$, and *decoded* (inverse of the encoding step) into a task-specific solution representation. Importantly, the cost involved in random-key encoding and decoding is practically negligible, which implies that there is little computational overhead compared to the core optimization steps.

In the context of AMT, once a probabilistic model capturing the search distribution corresponding to any source task is built (in the universal space), it can be transferred to a target optimization task of interest. Note that the possible difference in search space dimensionality of source and target optimization tasks can be addressed via a simple heuristic strategy. *If the dimensionality of the source optimization problem is greater than that of the target, the transferred probabilistic model is simply restricted to the active subset of variables by marginalizing over the distribution of all inactive variables. On the other hand, if the dimensionality of the source optimization problem is smaller, extra variables are padded to the source probabilistic model—with each new variable assigned an independent uniform distribution.*

B. Background on Mixture Modeling for Source–Target Similarity Capture

Given the premise of (2) and (3), herein, we present a brief overview of *finite mixture modeling* for probability density estimation. The goal of finite mixture modeling can be described as the linear combination of probabilistic models for estimating an unobservable latent probability density function based on observed data. Within the context of optimization problem-solving, consider $D_t = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ to represent a dataset of N solutions in a universal search space $\mathcal{X}$ at the tth generation of an EA tackling target task $\mathcal{T}_K$. Notice that since $D_0 = \emptyset$, offline source target similarity measurement is precluded. As introduced in (2), $p(\mathbf{x}|t)$ is the *true* latent probability density function describing the target population dataset D_t.

With this, the finite mixture model is defined as the following stacking (aggregation) of probabilistic models:

$$q(\mathbf{x}|t) = \sum_{k=1}^{K} \alpha_k \varphi_k(\mathbf{x}) \tag{4}$$

where $q(\mathbf{x}|t)$ is the desired *approximation* of the latent probability density function $p(\mathbf{x}|t)$, and components φ_k, $k = 1, \ldots, K$, are individual probabilistic models. In the present case, these K models include those that are drawn from the $K-1$ source optimization problems, as well as a *preliminary* model φ_K built for the target task $\mathcal{T}_K$. The coefficients in the mixture model (α_k's) are seen as capturing the *similarity* between the kth source and the target. If the probabilistic model φ_k has little relevance to the target, then the corresponding *transfer* coefficient α_k will take a value close to zero once the mixture is optimized (thereby reducing the influence of the corresponding source). On the other hand, if φ_k is useful for improving the approximation of the latent probability density function $p(\mathbf{x}|t)$, then α_k will take a relatively high value (closer to one). To this end, the objective of the mixture learning algorithm (detailed in Section III-C) is to deduce α_k's such that the probability of observing *out-of-sample* target data is maximized. Given an out-of-sample (test) dataset D_{test}, the mathematical program is accordingly formulated as the maximization of the following log-likelihood function:

$$\log L = \sum_{\mathbf{x}_i \in D_{\text{test}}} \log q(\mathbf{x}_i|t). \tag{5}$$

C. Adaptive Model-Based Transfer With Online Source–Target Similarity Learning

We approach the formulation in (2) by successively estimating the distribution $p(\mathbf{x}|t)$ of a population evolving toward the optimum of $\mathcal{T}_K$. In this regard, a distinguishing feature of the proposed stacked density estimation procedure, as opposed to prior work [33], is that $K-1$ *pretrained* probabilistic models originate from source tasks that are distinct from the ongoing target task of interest. Further, a source model φ_k may itself be a finite mixture model. Nevertheless, while solving $\mathcal{T}_K$, φ_k is always viewed as a single component encapsulating the knowledge acquired from the past k^{th} optimization experience.

With the learned mixture of probabilistic models capturing source–target similarities, the so-called adaptive transfer of knowledge is realized by iteratively sampling target candidate solutions from the mixture distribution. The theoretical rationale behind doing so, with regard to guiding the evolutionary search, shall be substantiated in Section V.

Next, we enumerate the steps followed for learning the optimal finite mixture model $q(\mathbf{x}|t)$ in practice.

Step 1: Randomly partition the target population D_t into v-folds according to the standard cross-validation procedure. For each fold, a target probabilistic model is learned from the *training part* of the partition of D_t. Thereafter, the likelihood of each data point in the *test partition* of D_t (which constitutes D_{test}) is evaluated corresponding to the $K-1$ pretrained source models and the learned target probabilistic model.

Step 2: By repeating step 1 for each of the v-folds, construct an $N \times K$ matrix comprising K density estimates for each of the N data points in D_t. The (i, k)th entry of the matrix is $\varphi_k(\mathbf{x}_i)$, representing the out-of-sample likelihood of the kth model on the ith data point in D_t.

Step 3: Using the matrix constructed from pretrained models in step 2, the α_k's of the mixture model are learned by maximizing the following equivalent form of (5):

$$\log L = \sum_{i=1}^{N} \log \sum_{k=1}^{K} \alpha_k \varphi_k(\mathbf{x}_i). \tag{6}$$

This can be (easily) solved by applying the classical expectation-maximization (EM) algorithm [37]. For the sake of brevity, details of the algorithm are not reproduced herein. Readers are referred to [38] for an intuitive description.

Step 4: To complete the learning process, consider the target probabilistic model trained on the entire training dataset D_t (without partitioning) to give φ_K. The final mixture model $q(\mathbf{x}|t)$ is thus the linear combination of the stored $K-1$ source probabilistic models $\varphi_1, \ldots, \varphi_{K-1}$ and the fully trained target model φ_K, with the combination given by the learned transfer coefficients (α_k's).

With regard to ascertaining the computational viability of the proposed AMT procedure, it is observed that the EM algorithm typically converges fast, i.e., within the first few iterations. In the case a *leave-one-out* cross-validation procedure is used in step 1, the complexity of repeatedly building, for example, a factored Bernoulli distribution model, is only $\mathcal{O}(N * d_K)$, as the model for each fold can be directly retrieved from the target model φ_K trained on the whole dataset. In other words, it is reasonable to incorporate the entire learning algorithm as a nested module within any external EA, at little computational overhead. Details of an instantiation of a generic transfer evolutionary optimization algorithm containing the aforesaid ideas are presented in the next section.

IV. Framework for AMT-Enabled Evolutionary Algorithm

The main motivation behind incorporating the notion of transfer in optimization is to effectively exploit the potentially rich pool of knowledge that may be found in previous problem-solving experiences. To this end, a transfer interval (denoted by Δ generations) is first introduced in the EA, which determines the frequency at which the AMT procedure (of Section III-C) is launched. Notice that since the AMT procedure is repeated periodically during the course of the evolutionary search, latent synergies, those that are less apparent at the start of the search, may in fact be gradually revealed. Overall, the frequency of transfer Δ controls the rate at which the target optimization search is subjected to the influence of the source tasks. In turn, it serves to manage the computational resources allocated to the stacked density estimation procedure; even though the computational cost associated with the learning module is small when using simplistic probabilistic models φ. To summarize, the AMT framework can be implemented as a nested subroutine within any canonical or state-of-the-art EA. A conceptual illustration of the basic structure of an AMTEA is shown in Fig. 1, and a general pseudocode enumerating the steps of our proposal is presented in Algorithm 1.

When introduced with a new target optimization problem $\mathcal{T}_K$, with dimensionality d_K, the initial population of the AMTEA algorithm is randomly generated. The iterative fitness-based parent selection and offspring creation process is then commenced, incrementing the generation count (t) by one at each iteration. While $\bmod(t, \Delta) \neq 0$, the AMTEA progresses in exactly the same manner as a standard EA, applying genetic operators such as crossover and/or mutation on the parent population P^s to produce the next generation of offspring individuals P^c.

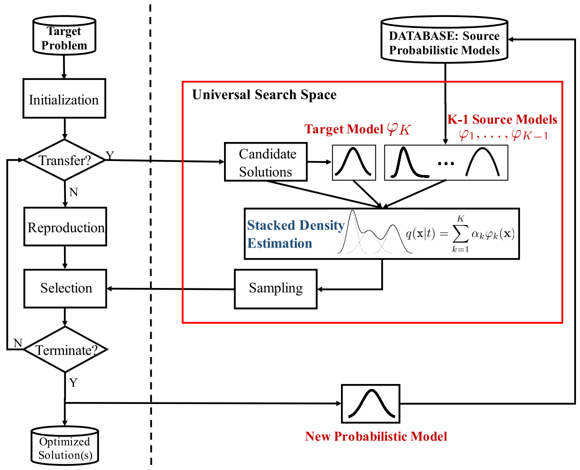


Fig. 1. Conceptual illustration of the proposed AMTEA.

Algorithm 1 Pseudocode of AMTEA

```
Input: Pre-trained source probabilistic models; target
    optimization problem; transfer interval $\Delta$
Output: Optimized solution(s) to the target optimization
    problem
Set $t = 1$
Randomly generate $N$ initial solutions: $P(t)$
Evaluate target objective values of individuals in $P(t)$
Marginalize or pad all pre-trained source models so as to
    match the dimensionality of the target problem
while stopping condition not satisfied do
    Sample parent population $P^s(t)$ from $P(t)$
    if $\bmod(t, \Delta) == 0$ then
        Encode $P^s(t)$ in the universal search space $\mathcal{X}$, to form
            the target dataset $D_t$
        Learn the mixture model $q(\mathbf{x}|t)$ describing $D_t$: fol-
            lowing Steps 1 to 4 in Section III-C
        Sample $q(\mathbf{x}|t)$, and decode the generated samples to
            form the offspring population $P^c(t)$
    else
        Generate offspring population $P^c(t)$ by genetic oper-
            ators such as crossover and/or mutation
    end if
    Evaluate individuals in $P^c(t)$
    Select next generation $P(t+1)$ from $P(t) \cup P^c(t)$
    Set $t = t + 1$
end while
```

Whenever $\bmod(t, \Delta) = 0$, the AMT procedure is launched. As a first step, the parent individuals in P^s are encoded in the universal search space $\mathcal{X}$, forming the target population dataset D_t. For continuous optimization problems, the unification can be efficiently achieved via linear scaling of variables to a common range [16], [34]. Various ways of encompassing combinatorial problems within a continuized unification scheme can also be found in [35]. Thereafter, previous optimization experiences, represented in the form of source probabilistic models, are referenced from the stored database. In order to match the search space dimensionality

d_K of the target problem, the models may be adjusted by either marginalizing out all nonactive variables from the source, or by padding the models with additional variables (that are assumed to follow a uniform distribution) to make up for any deficit. Next, stacked density estimation (refer steps 1–4 in Section III-C) is performed for optimal blending of source and target probabilistic models and effective capturing of source–target similarities online. As mentioned earlier, a large transfer coefficient learned for the mixture model indicates that the corresponding source model is highly correlated (and therefore relevant) to the target, while a small transfer coefficient (close to zero) reflects low similarity between the source and target. The ability to automatically attenuate the effects of such an irrelevant source model is the hallmark of this paper, as it automatically curbs the threat of negative transfer. Finally, the learned mixture model $q(\mathbf{x}|t)$ is sampled to generate the subsequent offspring population in the universal search space. The sampled offspring are decoded, yielding P^c in a task-specific solution representation. The objective function values of the offspring can then be evaluated.

The iterative parent selection and offspring creation process, interweaving evolutionary search and AMT, continues until prespecified stopping criteria for the AMTEA are met. It is worth mentioning that the final probabilistic model built for the target problem can be incorporated back into the database of optimization experiences (as shown in Fig. 1), which makes it available as a new building-block of knowledge for future problem-solving exercises.

V. Analyzing the Theoretical Foundations of Adaptive Model-Based Transfer

The asymptotic global convergence of a population-based stochastic optimization algorithm can be stated as

$$\lim_{t\to\infty}\int_{\mathcal{X}} f_K(\mathbf{x})p(\mathbf{x}|t)d\mathbf{x} = f_K^* \tag{7}$$

where f_K is the objective function of the target optimization problem defined in the universal search space $\mathcal{X}$, f_K^* is its globally optimum value, and $p(\mathbf{x}|t)$ is the latent probability density function of the population at generation t.

In this section, we highlight that the proposed AMT framework facilitates global convergence characteristics. For simplicity of analysis, it is assumed that the population size employed in the AMTEA is large, i.e., $N \to \infty$, and f_K is continuous. While such an assumption may not hold in practice, it is considered reasonable for demonstrating the theoretical foundations of our proposal, which are found to be borne out by the experimental studies. In fact, similar assumptions are adopted in the theoretical analyses of [39], which serves as an important stepping-stone for the AMT procedure. Specifically, the main result of the interest is as follows.

Theorem 1 (Zhang and Muhlenbein [39]): In probabilistic-modeling-based EA, where $p(\mathbf{x}|t = 0)$ is positive and continuous in $\mathcal{X}$, asymptotic global convergence is guaranteed for continuous objective functions if $p^s(\mathbf{x}|t) = p^c(\mathbf{x}|t)$; where $p^s(\mathbf{x}|t)$ and $p^c(\mathbf{x}|t)$ are the underlying distributions of P^s and P^c, respectively, at any generation t.

In practice, there invariably exists a *gap* between $p^c(\mathbf{x}|t)$ and the *true* latent probability density function $p^s(\mathbf{x}|t)$ of the parent population. As a result, significant efforts have been made over the years for developing increasingly sophisticated methods for improved modeling of probability density functions. In this regard, *Theorem 1 suggests that the role of AMT in facilitating global convergence characteristics can be established by simply showing that the proposed mixture of all available (source + target) probabilistic models guarantees superior approximations of population distributions.*

Lemma 1: Maximizing the log-likelihood function in (6) is equivalent to minimizing the gap [measured by the Kullback–Leibler (KL) divergence] between the mixture model $q(\mathbf{x}|t)$ and the population's true latent probability density function $p(\mathbf{x}|t)$, given $N \to \infty$.

Proof: With $q(\mathbf{x}|t)$ as the finite mixture approximation of $p(\mathbf{x}|t)$, (5) can be rewritten as

$$\lim_{N\to\infty}\frac{1}{N}\cdot\log L = \lim_{N\to\infty}\frac{\sum_{i=1}^{N}\log q(\mathbf{x}_i|t)}{N}. \tag{8}$$

Since the N population samples are drawn from the true latent probability density function $p(\mathbf{x}|t)$, the Glivenko–Cantelli theorem [40] indicates that the empirical probability density function induced by $N(\to\infty)$ converges to $p(\mathbf{x}|t)$. Thus,

$$\lim_{N\to\infty}\frac{1}{N}\cdot\log L = \int_{\mathcal{X}} p(\mathbf{x}|t)\log q(\mathbf{x}|t)d\mathbf{x}. \tag{9}$$

With this, we utilize the KL divergence as a common measure of evaluating the gap between two distinct probability density functions. In particular, the measure specifies the amount of information lost when $q(\mathbf{x}|t)$ is used to approximate $p(\mathbf{x}|t)$, which is given as [41]

$$\begin{aligned}\mathrm{KL}(p||q) &= \int_{\mathcal{X}} p(\mathbf{x}|t)\log\frac{p(\mathbf{x}|t)}{q(\mathbf{x}|t)}d\mathbf{x}\\ &= \int_{\mathcal{X}} p(\mathbf{x}|t)\big[\log p(\mathbf{x}|t) - \log q(\mathbf{x}|t)\big]d\mathbf{x}. \end{aligned}\tag{10}$$

Therefore, maximizing $\int_{\mathcal{X}} p(\mathbf{x}|t)\log q(\mathbf{x}|t)d\mathbf{x}$ in (9), for a given $p(\mathbf{x}|t)$, is equivalent to minimizing $\mathrm{KL}(p||q)$. Further, since $\mathrm{KL}(p||q) \geq 0$ as per Gibbs' inequality [42], it can be concluded that maximizing the log-likelihood function in (5) minimizes the distribution gap between the mixture model $q(\mathbf{x}|t)$ and $p(\mathbf{x}|t)$, with the gap being bounded from below by zero. The same result holds for the maximization of (6) which is an equivalent form of (5). ■

Lemma 2: The EM algorithm for maximizing (6) converges to the global optimum.

Proof: The EM algorithm converges to a stationary point of the log-likelihood function [43]. Further, it is known that the KL divergence is convex in the domain of probability distributions. With this, it can be seen that since the component models (φ_k's) of (6) are pretrained, the log-likelihood function must be convex upward with respect to α_k's. Thus, the stationary point is also the global optimum. ■

Theorem 2: Stacked density estimation with all available (source + target) probabilistic models guarantees $\mathrm{KL}(p^s||p^c) \leq \mathrm{KL}(p^s||p^c_{\text{sub}})$, where $\mathrm{KL}(p^s||p^c_{\text{sub}})$ represents the distribution gap achievable by any proper subset of the models.

Proof: Let $\mathcal{S}$ denote the set of all available (source + target) probabilistic models. Thus, $|\mathcal{S}| = K$. Further, let $\mathcal{S}_{\text{sub}}$ be any proper subset of $\mathcal{S}$, i.e., $\mathcal{S}_{\text{sub}} \subset \mathcal{S}$, and $\mathcal{S}'_{\text{sub}} = \mathcal{S} \setminus \mathcal{S}_{\text{sub}}$. The mathematical program of (6) given $\mathcal{S}$ can be written as

$$\max\ :\ \log L = \sum_{i=1}^{N} \log \sum_{\varphi_k \in \mathcal{S}} \alpha_k \varphi_k(\mathbf{x}_i). \tag{11}$$

Similarly, (6) given $\mathcal{S}_{\text{sub}}$ can be written as

$$\max\ :\ \log L_{\text{sub}} = \sum_{i=1}^{N} \log \sum_{\varphi_k \in \mathcal{S}_{\text{sub}}} \alpha_k \varphi_k(\mathbf{x}_i). \tag{12}$$

This is equivalent to

$$\begin{aligned} \max\ :\ \log L_{\text{sub}} &= \sum_{i=1}^{N} \log \sum_{\varphi_k \in \mathcal{S}} \alpha_k \varphi_k(\mathbf{x}_i) \\ \text{s.t.}\quad \alpha_k &= 0, \forall \varphi_k \in \mathcal{S}'_{\text{sub}}. \end{aligned} \tag{13}$$

Comparing (11) to (13), and given the result of Lemma 2, it is guaranteed that

$$\log L^* \geq \log L^*_{\text{sub}} \tag{14}$$

where (*) indicates the global maximum.

Next, notice that as the offspring population in AMT is sampled from the mixture model, we have

$$p^c(\mathbf{x}|t) = \sum_{\varphi_k \in \mathcal{S}} \alpha_k \varphi_k(\mathbf{x}) \tag{15}$$

and

$$p^c_{\text{sub}}(\mathbf{x}|t) = \sum_{\varphi_k \in \mathcal{S}_{\text{sub}}} \alpha_k \varphi_k(\mathbf{x}). \tag{16}$$

Further, the training dataset $D_t = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ is assumed to be drawn from $p^s(\mathbf{x}|t)$. Therefore, Lemma 1 together with (14) imply that $\text{KL}(p^s||p^c) \leq \text{KL}(p^s||p^c_{\text{sub}})$. ■

This result tells us that by combining all available (source + target) probabilistic models, we can reduce the gap between $p^s(\mathbf{x}|t)$ and $p^c(\mathbf{x}|t)$ as compared to using any subset of the models. *In fact, with increasing number of source models, we can in principle make the gap arbitrarily small.* The consequences of such a result toward guaranteeing global convergence behavior of the overall EA are already substantiated by Theorem 1.

At this juncture, it is to be observed that if the target probabilistic model φ_K were to (hypothetically) exactly replicate the parent population's latent probability density function, i.e., $\varphi_K = p^s(\mathbf{x}|t)$, then there would occur no knowledge transfer across problems. This can be shown through Gibb's inequality, which, given $\varphi_K = p^s(\mathbf{x}|t)$, implies that

$$\int_{\mathcal{X}} p^s(\mathbf{x}|t) \log p^c(\mathbf{x}|t) d\mathbf{x} \leq \int_{\mathcal{X}} p^s(\mathbf{x}|t) \log \varphi_K(\mathbf{x}) d\mathbf{x}. \tag{17}$$

Thus, based on the global convergence property of the EM algorithm (as shown in Lemma 2), $\alpha_k = 0$ for $k = 1, \ldots, K-1$, and $\alpha_K = 1$. In other words, the transfer coefficients will be zero, thereby indicating the cancellation of transfer. This suggests that while employing finite population sizes in practical applications of the AMTEA, an extra effort may be needed to prevent overfitting of the target models to the training dataset D_t. To this end, during every iteration of AMT (refer Section III-C and Algorithm 1), we propose to artificially add a small amount of random noise to the dataset while learning target probabilistic models.

VI. Experimental Study

In this section, numerical results are presented that demonstrate the efficacy of the AMTEA framework, with regard to online learning and exploitation of source–target similarities. We conduct experiments across a series of problem categories, ranging from discrete to continuous, as well as single-objective to MOO. In addition, a case study on increasingly challenging variants of the double-pole balancing controller design task is carried out, to showcase the practical utility of the method.

For rigorous investigation, the performance of the AMTEA is compared against a number of baseline solvers. First of all, we consider the basic counterpart of the AMTEA, i.e., a canonical EA (CEA) or, alternatively, a canonical memetic algorithm (CMA) if some form of local solution refinement is incorporated. The solution representation scheme employed in the CEA or CMA changes depending on the underlying problem being solved. For the case of MOO, the baseline solvers used are the popular NSGA-II [44] and MOEA/D [45]. For a representative knowledge-based (transfer) multiobjective optimizer, the recently proposed *autoencoding evolutionary* (AE) approach [13] is chosen for our comparison study. Labeled herein as AE-NSGAII, the method uses a *denoising autoencoder* to serve as a bridge between the source domain and the search space of the target optimization problem. Finally, as an instantiation of a general purpose transfer evolutionary optimization algorithm, we consider a *transfer case-injected* EA (TCIEA) in which a small number of stored solutions from the source database—those that are similar to the current best target solution—are selected (via case by case assessment) and periodically injected (at transfer interval Δ) into the target evolutionary search [6].

For the double-pole balancing controller design task, a state-of-the-art population-based stochastic optimizer, termed as *natural evolution strategies* (NESs) [46], is adopted as an additional baseline for comparison. It is worth mentioning that NES has recently garnered much attention as a powerful and scalable alternative to reinforcement learning [47].

A. Experimental Configuration

The experimental setup is outlined as follows. We use an *elitist* selection strategy throughout all experiments with the AMTEA and CEA (or CMA). The (universal) solution representation scheme, and the choice of probabilistic models, are dictated by the underlying problem specifications. For the chosen set of discrete optimization examples, the following general settings are applied.

1) *Representation:* Binary-coded.
2) *Population Size (N):* 50 for AMTEA, CEA (or CMA), and TCIEA.
3) *Maximum Function Evaluations:* 5000.

4) Evolutionary operators for AMTEA, CEA (or CMA), and TCIEA:
   a) uniform crossover with probability $(p_c) = 1$;
   b) bit-flip mutation with probability $(p_m) = 1/d$.
5) *Probabilistic Model:* Univariate marginal frequency (factored Bernoulli distribution) [48].

For continuous optimization problems, the experimental setup is outlined as follows. The same settings are incorporated for both single- and multi-objective problems. Note that for multiobjective cases, AMTEA, TCIEA, and AE-NSGAII are built upon the standard NSGA-II. Thus, for fairness of comparison, the genetic operators employed in these solvers are kept identical.

1) *Representation:* $[0, 1]^d$.
2) *Population Size (N):* 100 for AMTEA, TCIEA, AE-NSGAII, NSGA-II, and MOEA/D.
3) *Maximum Function Evaluations:* 10 000.
4) Genetic operators for AMTEA, TCIEA, AE-NSGAII, and NSGA-II:
   a) simulated binary crossover [49] with $p_c = 1$ and distribution index $\eta_c = 10$;
   b) polynomial mutation [50] with $p_m = 1/d$ and distribution index $\eta_m = 10$.
5) Probabilistic Model: Multi-variate Gaussian distribution.

The MOEA/D algorithm incorporates traditional differential evolution operators, with $F = 0.5$ and $CR = 0.5$ [45], [51].

B. Toy Problems: Functions of Unitation

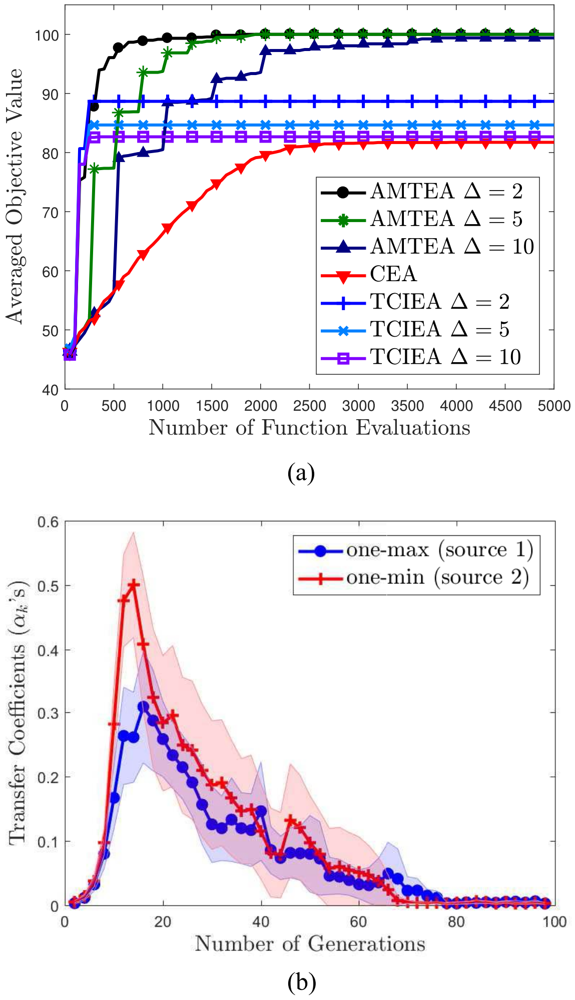
Functions of unitation encompass a class of discrete problems with binary representation for which the objective depends on the number of ones in a bitstring. For example, the commonly used one-max problem simply states to maximize the number of bits set to one in its chromosome. In contrast, the one-min problem states to maximize the number of bits set to zero (or equivalently, minimize the number of bits set to one) in its chromosome.

While both the one-max and one-min problems have a single optimum, complex (deceptive) functions of unitation with multiple local optima can also be formulated. A popular example is the trap function of order five, denoted as trap-5 [52]. In trap-5, a candidate bitstring is partitioned into groups of five bits each. The contribution of each group toward the combined objective function is calculated as

$$f_{\text{trap5}}(u) = \begin{cases} 4 - u & \text{if } u < 5 \\ 5 & \text{otherwise} \end{cases} \tag{18}$$

where u is the number of ones in the group. Trap-5 has one global optimum (when all the bits in the input string equal one), and $2^{d/5} - 1$ other local optima. Note that its global optimum is identical to that of one-max, while the *worst* (highly deceptive) local optimum corresponds to the global optimum of one-min.

In the experimental study, we solve 100-D variants of the trap-5 function and the one-min problem. For the case of trap-5, the source probabilistic models are assumed to come from optimization experiences on: 1) one-max and 2) one-min. We set the transfer interval as $\Delta = 2, 5$, and 10 to study the effect of the transfer interval on the global convergence characteristics of the AMTEA. For the case of one-min, source probabilistic models are drawn from: 1) one-max; 2) a trap-5 run where solutions reached the global optimum; and 3) a trap-5 run where solutions were trapped at the deceptive local optimum.



(a)

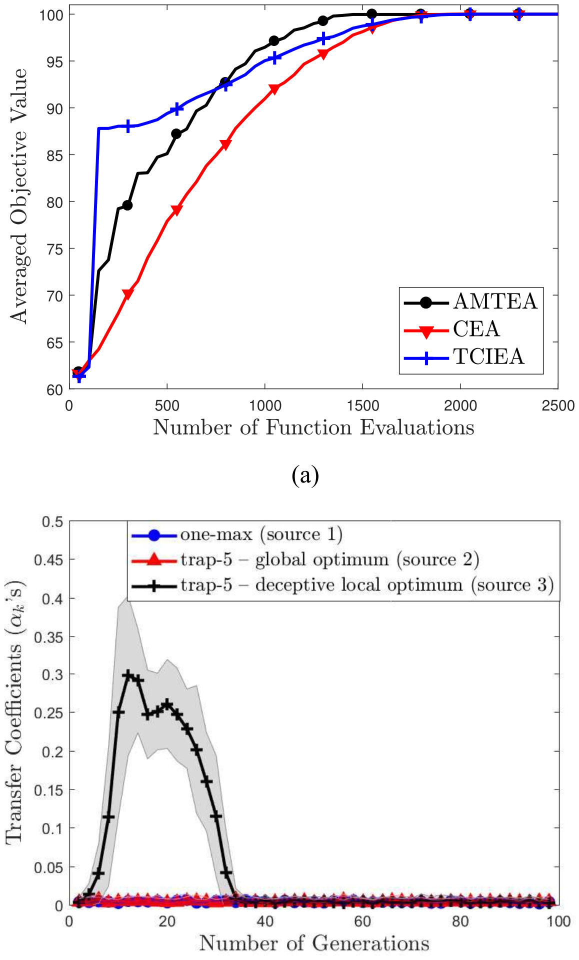
(b)

Fig. 2. Convergence trends and transfer coefficients learned for trap-5 (the shaded region spans one standard deviation either side of the mean). One-max and one-min act as source tasks. (a) Convergence trends of trap-5 using different Δ's. (b) Transfer coefficients learned when $\Delta = 2$.

The convergence trends for trap-5 are shown in Fig. 2 (results are averaged over 30 independent runs). From Fig. 2(a), it can be seen that CEA always gets trapped at the deceptive local optimum. On the other hand, augmented with the knowledge acquired from prior problem-solving experiences on one-max and one-min, AMTEAs with different transfer intervals show consistently superior performances. Notably, for $\Delta = 2, 5$, the global optimum is achieved in every run. On the other hand, for $\Delta = 10$, the global optimum is reached in 29 out of 30 runs. The minor decrease in performance for a larger transfer interval suggests that there is a tendency that the population gets trapped in a local optimum if no external knowledge is received for a prolonged duration. Under this observation, and keeping in mind the negligible computational cost of stacked density estimation given univariate marginal probabilistic models, we set $\Delta = 2$ in all subsequent discrete optimization examples.

Fig. 2(b) shows the trends of the learned transfer coefficients across generations. As indicated by the high values of the transfer coefficients, the target optimization problem can be seen to receive significant transfer from both sources, i.e., one-max and one-min. This implies that the population tends to split into two groups, one of which is inevitably drawn toward the highly deceptive local optimum (under the influence of one-min), while the other manages to converge to the global optimum (under the positive influence of one-max). One aspect of the result in Fig. 2(a) is that TCIEA often gets trapped in a local optimum. This may be attributed to the danger of negative transfer in a case by case solution assessment for transfer procedure, particularly in the absence of any automated source–target similarity modelings.

With regard to one-min as the target problem, its relative simplicity allows all three algorithms, namely, AMTEA, CEA, and TCIEA, to solve it successfully [as shown by Fig. 3(a)]. The key point to highlight here is the efficacy of the AMTEA in terms of learning the source–target similarities. As shown by the learned transfer coefficients in Fig. 3(b), the AMT procedure is able to precisely identify the overlap between the one-min problem and the deceptive local optimum of trap-5 (which is represented by one of the source probabilistic models). The benefits of deciphering such similarities on the fly, without the need for any human intervention, shall be revealed over subsequent subsections in more practically relevant scenarios.



(a)

(b)

Fig. 3. Convergence trends and transfer coefficients learned while solving one-min (the shaded region spans one standard deviation either side of the mean). Source probabilistic models are drawn from (1) one-max, (2) a trap-5 run where solutions reached the global optimum, and (3) a trap-5 run where solutions were trapped in the deceptive local optimum. (a) Convergence trends of one-min. (b) Transfer coefficients learned.

C. Case Study on the 0/1 Knapsack Problem

Knapsack problem (KP) is a classical NP-hard problem in discrete (combinatorial) optimization, popularly studied in the operations research literature. It offers several practical applications in many different areas, including logistics, auction winner determination, investment decision making, as well as portfolio optimization.

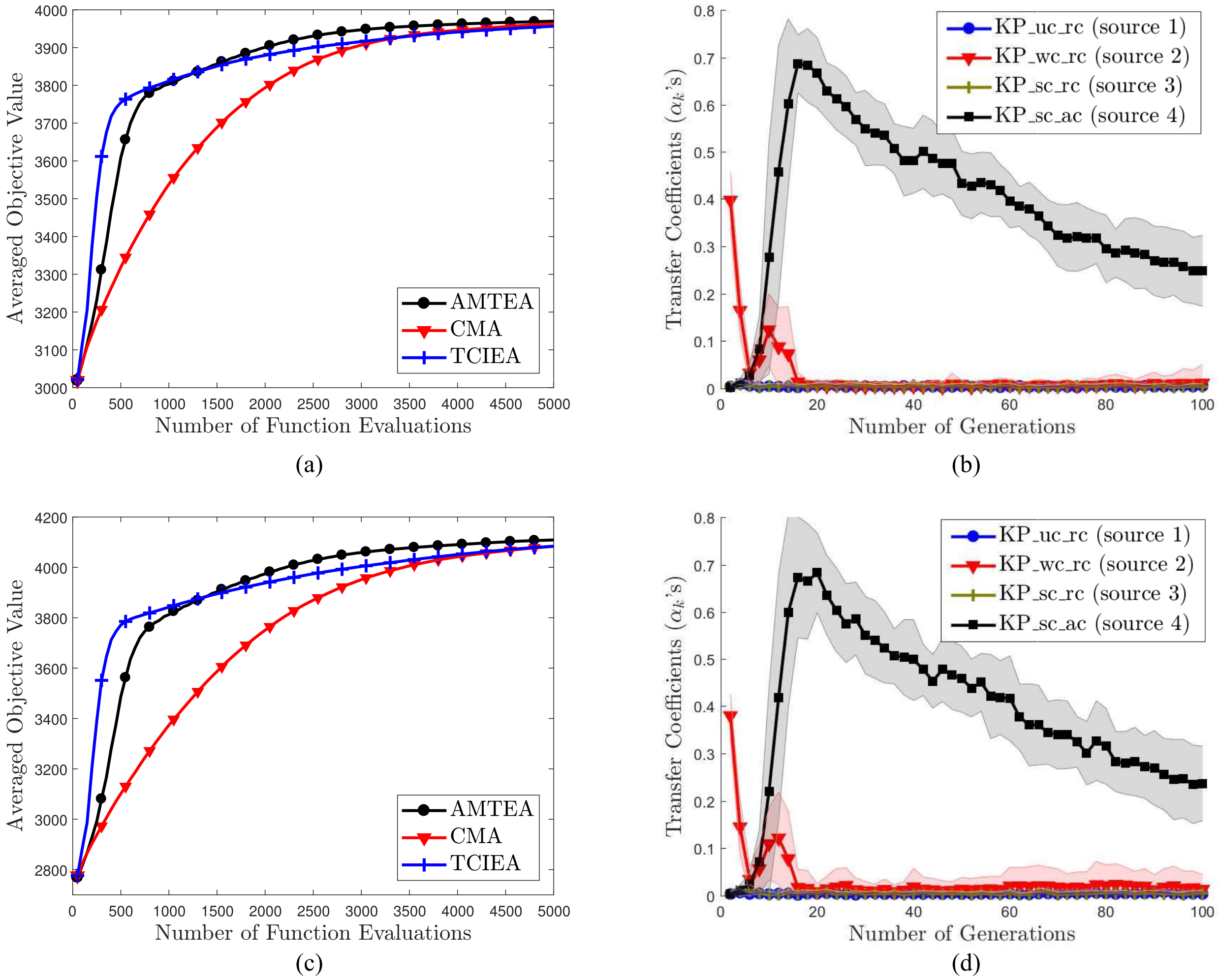
The objective of the problem, given a knapsack of capacity C, and a set of d items, each with a weight w_i and a value q_i, is to find a selection of items such that the total value is maximized without violating the capacity constraint. The mathematical formulation of the KP is defined as follows:

$$\max \sum_{i=1}^{d} q_i x_i$$
$$\text{s.t. } \sum_{i=1}^{d} w_i x_i \leq C \text{ and } x_i \in \{0, 1\}. \tag{19}$$

Here, $x_i = 1$ indicates that the ith item is selected, while $x_i = 0$ indicates that the ith item is not selected.

In this paper, artificial KP instances are generated using the technique proposed in [53]. Accordingly, the KP instances are categorized into *uncorrelated* (uc), *weakly correlated* (wc), and *strongly correlated* (sc), depending on the relationship between the w's and q's. Further, there are considered to be two types of knapsacks: *restrictive capacity* (rc) and *average capacity* (ac). For a restrictive knapsack, only a small number of items are expected to be selected, while for an average knapsack, the number of selected items will be larger. We concatenate the two subcategories to form six $d = 1000$-D KP instances. For example, a KP problem denoted as "KP_uc_rc" indicates that the w's and q's are weakly correlated, and the knapsack is of restrictive capacity.

In the experimental study, it is noted that the evolutionary search can often cause the capacity constraint of the knapsack to be violated. Therefore, we include a local solution refinement (repair) step in all solvers following Dantzig's greedy approximation algorithm [53]. Next, we consider the optimization runs of four KP instances, namely, "KP_uc_rc", "KP_wc_rc," "KP_sc_rc," and "KP_sc_ac," to act as sources from which source probabilistic models are drawn. Instances "KP_wc_ac" and "KP_uc_ac" act as the target optimization problems of interest. We compare the proposed AMTEA with CMA (since local refinement is applied) and TCIEA. The transfer interval is set as $\Delta = 2$. Fig. 4 shows the averaged



(a)

(b)

(c)

(d)

Fig. 4. Convergence trends and transfer coefficients for "KP_wc_ac" and "KP_uc_ac" (the shaded region spans one standard deviation either side of the mean). "KP_uc_rc", "KP_wc_rc", "KP_sc_rc", and "KP_sc_ac" serve as source optimization problems. (a) "KP_wc_ac". (b) Transfer coefficients learned for "KP_wc_ac". (c) "KP_uc_ac". (d) Transfer coefficients learned for "KP_uc_ac".

results obtained (over 30 independent runs). It is clear that algorithms AMTEA and TCIEA, with the scope of knowledge transfer, perform significantly better than CMA. While the TCIEA is found to rapidly increase the obtained objective function values in the initial stages of evolution, it is eventually surpassed by AMTEA on both the target tasks.

Most importantly, Fig. 4(b) and (d) shows the source–target similarities captured by the AMTEA. While all the KP instances belong to distinct subcategories, it is remarkable to note that the algorithm successfully identifies the source task that is intuitively expected to be most relevant. To elaborate, since both target tasks are characterized by average knapsack capacity, a relatively large number of items must be selected. However, among the set of source probabilistic models available, only "KP_sc_ac" belongs to the average knapsack capacity category. All other sources are characterized by restrictive knapsacks where only a small number of items can be selected. Therefore, simple intuition dictates that the optimum solutions of target tasks "KP_wc_ac" and "KP_uc_ac", respectively, should be most similar to that of source "KP_sc_ac". Our expectation is precisely borne out by experiments, which highlights the key contribution of this paper with regard to automatically unveiling the synergies between distinct optimization tasks online.

D. Case Study in Multiobjective Optimization

Population-based EAs have gained popularity over the years as noteworthy solvers of MOO problems, primarily due to their ability (derived from the implicit parallelism of the population) to simultaneously converge toward the entire set of optimal solutions (commonly referred to as Pareto optimal solutions) of MOO problems [45]. Endowing multiobjective EAs (MOEAs) with knowledge transfer capabilities, is therefore expected to further push the envelope of evolutionary methods in this domain.

In this paper, we carry out numerical experiments on three popularly used MOO benchmark test sets, namely ZDT (ZDT1–4 and ZDT6) [54], DTLZ [55], and WFG [56]. When solving any one of the problems in the three test sets, using the AMTEA, TCIEA, or AE-NSGAII, all the other problems in that set act as source tasks providing source probabilistic models for transfer. Keeping in mind the modest computational cost involved in the stacking of *multivariate*

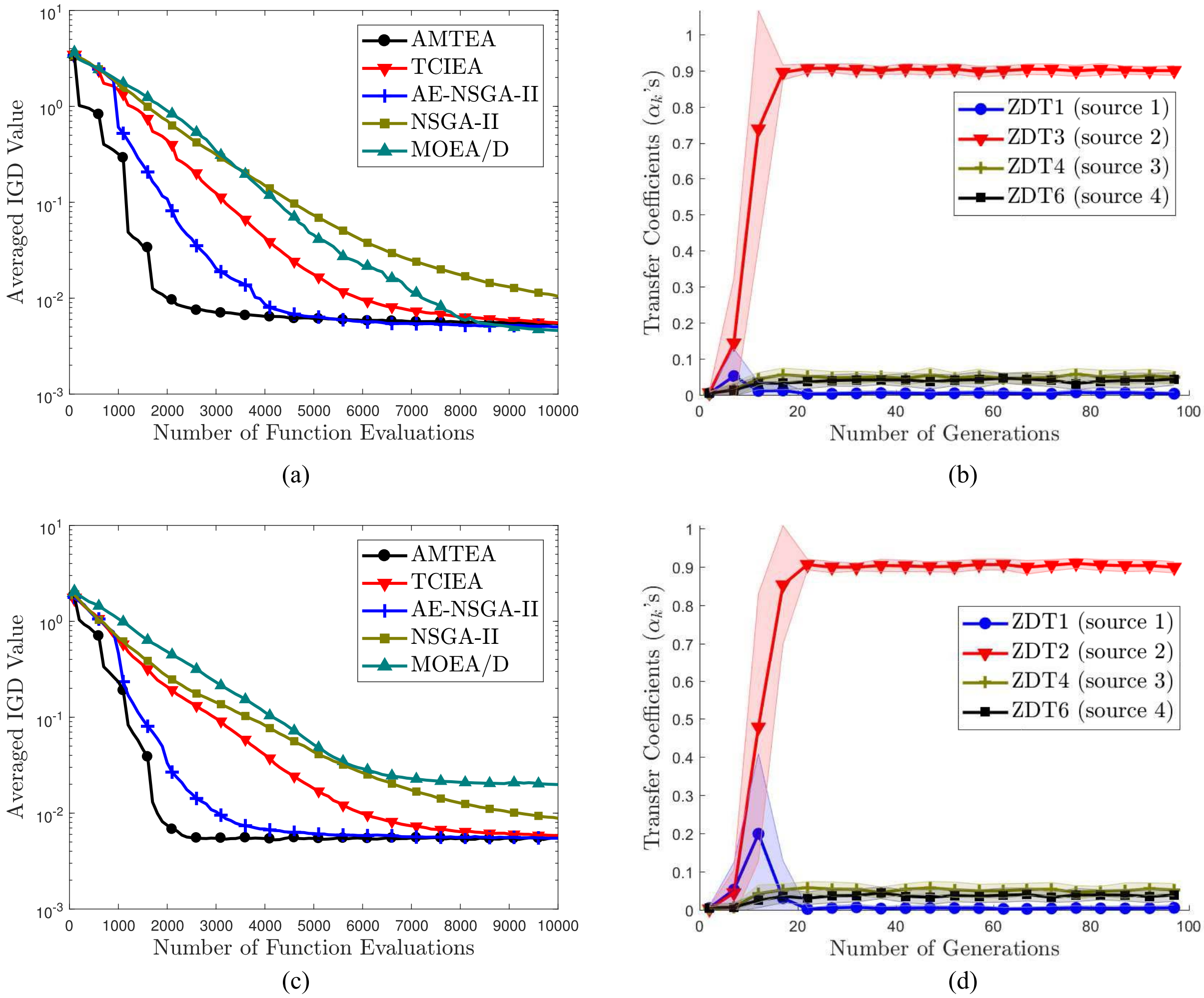


Fig. 5. Convergence trends and transfer coefficients learned for ZDT2 and ZDT3, with all other problems in the ZDT family serving as source tasks. Notice that the source–target similarities learned between ZDT2 and ZDT3 is particularly highly, which can be explained by the fact that these two problems have similar Pareto optimal solutions. (a) ZDT2. (b) Transfer coefficients learned for ZDT2. (c) ZDT3. (d) Transfer coefficients learned for ZDT3.

Gaussian distribution models, the transfer interval is relaxed to $\Delta = 10$ in these experiments. We consider the inverted generational distance (IGD) [57] as the performance metric for comparisons.

The numerical results achieved at the end of 10 000 function evaluations are shown in Table I. The nonparametric Wilcoxon signed-rank test reveals that AMTEA performs significantly the best in a majority of the test problems (precisely, in 12 out of 21 test problems). Notably, AMTEA is found to often surpass the performance of the recently proposed autoencoding-based transfer evolutionary optimizer AE-NSGAII. Fig. 5 shows the accelerated convergence characteristics achieved on ZDT2 and ZDT3 by the AMTEA, in comparison to other baseline algorithms.

On further inspection, it is revealed based on the equations of ZDT2 and ZDT3 [44], that the Pareto optimal solutions of the two tasks are indeed similar to each other when mapped to the universal search space. This latent property is automatically deciphered and harnessed by the AMTEA, as can be seen in Fig. 5(b) and (d) where the proposed algorithm is found to consistently identify high transfer coefficients between these two tasks.

E. Double-Pole Balancing Controller Design Task

Double-pole balancing is a controller design task popularly used as a case study for reinforcement learning algorithms. The goal is to prevent two poles, both affixed at the same point on a cart (which is restricted to a finite stretch of track), from falling over by applying a force to the cart [58].

In the problem setup, the state of the system can be fully defined by six variables: the angle of each pole from vertical, the angular velocity of each pole, the position of the cart on the track, and the velocity of the cart (see [59] for the equations of motion and parameters used in this task). The length of the long pole is set to 1 m, and we use e_l to denote the task where l is the length of the shorter pole (also in meters). The Runge–Kutta fourth-order method is used to numerically simulate the system, with a step size of 0.01 s. During simulations, a simple feedforward neural network (FNN) controller, with fixed structure, is used to output a force that acts on the

TABLE I
AVERAGED IGD VALUES OBTAINED BY AMTEA, NSGA-II, TCIEA, AE-NSGAII, AND MOEA/D OVER 30 INDEPENDENT RUNS. VALUES IN BRACKETS INDICATE STANDARD DEVIATIONS. NUMBERS WITH (*) INDICATE THE BEST PERFORMING ALGORITHMS AT 95% CONFIDENCE LEVEL AS PER THE WILCOXON SIGNED-RANK TEST

Problem	AMTEA	NSGA-II	TCIEA	AE-NSGAII	MOEA/D
ZDT1	0.0051	0.0095	0.0056	***0.0049**	0.0054
($d = 30$)	(0.0002)	(0.0014)	(0.0003)	(0.0002)	(0.0004)
ZDT2	0.0054	0.0105	0.0055	0.0050	***0.0046**
($d = 30$)	(0.0003)	(0.0015)	(0.0003)	(0.0003)	(0.0001)
ZDT3	***0.0055**	0.0088	0.0058	***0.0055**	0.0198
($d = 30$)	(0.0004)	(0.0013)	(0.0003)	(0.0002)	(0.0197)
ZDT4	0.4479	***0.2926**	0.5751	0.4189	2.0804
($d = 10$)	(0.1973)	(0.1354)	(0.2372)	(0.2285)	(0.6588)
ZDT6	0.0216	0.0723	0.0481	***0.0042**	0.0111
($d = 10$)	(0.0068)	(0.0365)	(0.0196)	(0.0003)	(0.0306)
DTLZ1	***25.8171**	94.3008	69.5432	167.3022	78.9990
($d = 30$)	(33.1267)	(19.3544)	(16.1795)	(21.5871)	(16.9633)
DTLZ2	***0.0730**	0.0932	0.0755	0.0954	0.0783
($d = 30$)	(0.0035)	(0.0063)	(0.0039)	(0.0064)	(0.0065)
DTLZ3	171.2978	225.2871	***160.5694**	477.5128	303.5445
($d = 30$)	124.4363	(57.8071)	(43.8617)	(59.8171)	(76.4256)
DTLZ4	***0.0724**	0.0940	0.0791	0.1000	0.1485
($d = 30$)	(0.0074)	(0.0126)	(0.0409)	(0.0389)	(0.0997)
DTLZ5	***0.3143**	0.4259	0.3857	0.4995	0.4389
($d = 30$)	(0.0275)	(0.0431)	(0.0489)	(0.0887)	(0.1319)
DTLZ6	8.1979	13.7044	12.5287	***1.8073**	4.4750
($d = 30$)	(0.9992)	(0.9382)	(2.4065)	(0.5296)	(1.4279)
DTLZ7	0.1152	0.1148	0.1064	***0.0806**	0.2540
($d = 30$)	(0.0094)	0.0148	(0.0496)	(0.0051)	(0.1833)
WFG1	***1.0705**	1.0807	1.0719	1.0986	1.1655
($d = 24$)	(0.0076)	(0.0099)	(0.0100)	(0.0074)	(0.0205)
WFG2	***0.0290**	0.1036	0.0697	0.0938	0.2460
($d = 24$)	(0.0224)	(0.0209)	(0.0152)	(0.0227)	(0.0826)
WFG3	***0.1464**	0.1924	0.1623	0.1945	0.1731
($d = 24$)	(0.0014)	(0.0162)	(0.0067)	(0.0111)	(0.0133)
WFG4	***0.0173**	0.0357	0.0274	0.0362	0.0794
($d = 24$)	(0.0010)	(0.0045)	(0.0023)	(0.0045)	(0.0094)
WFG5	0.0732	0.0729	0.0736	0.0725	***0.0696**
($d = 24$)	(0.0013)	(0.0014)	(0.0011)	(0.0013)	(0.0006)
WFG6	***0.0195**	0.0875	0.0380	0.0957	0.0960
($d = 24$)	(0.0010)	(0.0113)	(0.0155)	(0.0138)	(0.0211)
WFG7	***0.0185**	0.0267	0.0228	0.0294	0.0271
($d = 24$)	(0.0013)	(0.0035)	(0.0015)	(0.0032)	(0.0021)
WFG8	0.1943	0.1635	0.1708	0.1671	***0.1501**
($d = 24$)	(0.0115)	(0.0062)	(0.0080)	(0.0109)	(0.0090)
WFG9	***0.0257**	0.0918	0.0349	0.0913	0.0954
($d = 24$)	(0.0017)	(0.0437)	(0.0178)	(0.0420)	(0.0321)

TABLE II
PERFORMANCES WHILE SOLVING THE DOUBLE-POLE BALANCING PROBLEM USING DIFFERENT SOLVERS. SUPERIOR PERFORMANCE IS HIGHLIGHTED IN BOLD

Methods	Successes	Function Evaluations
CEA	0/50	NA
NES	1/50	8977
TCIEA	1/50	8900
AMTEA	**22/50**	**7918**±1241

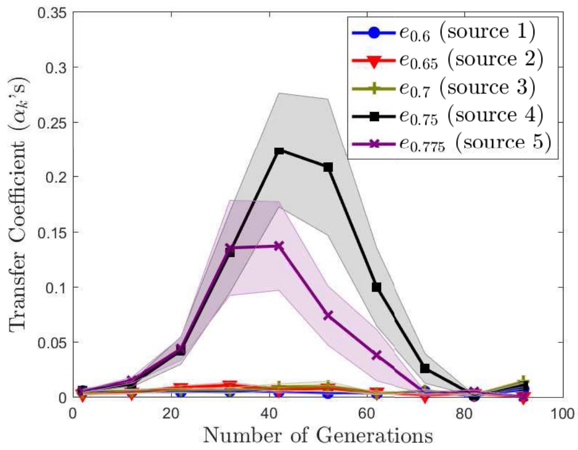


Fig. 6. Transfer coefficient trends learned by AMTEA while solving $e_{0.8}$. The shaded region spans one standard deviation either side of the mean.

cart every 0.02 s in the range of $[-10, 10]N$. The initial angle of the long pole is set to 1°. The control task is considered to be a failure if the cart goes out of bounds of a 4.8 m track, or else, one of the two poles drops beyond ±36° from the vertical. The fitness of a candidate solution (which encodes the synaptic weights of the FNN controller) is the number of time steps taken for the system to fail. Evidently, the problem is that of fitness maximization. Note that a task is considered solved if the fitness of the corresponding solution is over 100 000 time steps, which is approximately 30 min in simulated time.

From the previous studies, it is well-established that the double-pole system becomes more difficult to control as the poles assume similar lengths [60], i.e., as the length of the shorter pole l approaches 1 m. A number of optimization efforts with state-of-the-art solvers were performed to verify that the problem indeed becomes progressively harder to solve within a reasonable amount of time as the length of the shorter pole approaches 0.8 m. Thus, in the context of AMT, it is natural to raise the question: *is it possible to utilize previous problem-solving experiences on easier problems to help solve increasingly challenging variants of a problem at hand?*

To investigate this matter further, we use probabilistic models drawn from source optimization tasks with $l = 0.1, 0.2, \ldots, 0.775$ (13 source tasks in all) while tackling $e_{0.8}$ as the target task. A two-layer FNN controller with ten hidden neurons is applied. Bias parameters are removed due to the symmetry property of the system. This leads to a total of 70 weights to be tuned by the optimizer.

In the experimental study, *success rate* is the performance metric used to compare AMTEA against TCIEA, CEA, and the recently proposed NES algorithm.[1] According to the results tabulated in Table II, the CEA can never achieve success in any of its 50 independent runs. Even NES and TCIEA succeed only once out of their 50 runs, thereby demonstrating the considerable difficulty of $e_{0.8}$. On the other hand, the AMTEA achieves significantly higher success rate, effectively balancing the pole in 22 out of 50 runs. Notably, among the successfully completed runs, the averaged number of function evaluations consumed by AMTEA is approximately 7918, while that consumed by TCIEA and NES is more than 8900.

With regard to the transfer coefficients learned in the AMTEA, we refer to Fig. 6. Note that the figure only presents transfer coefficients corresponding to 5 (out of the 13) source optimization problems, as the remaining sources showcase consistently low knowledge transfer. It can be seen that a

[1] With default parameter setting as available from http://people.idsia.ch/∼tom/nes.html.

lot of transfer occurs to the target task $e_{0.8}$ from sources $e_{0.75}$ and $e_{0.775}$. This is once again a noteworthy example where the AMTEA is automatically able to identify what are intuitively expected to be the most relevant sources of information. Furthermore, the considerably superior performance of AMTEA as compared to TCIEA substantiates the impact of the proposed approach in enhancing optimization performance in general. The benefits of online source–target similarity modeling, as opposed to a case by case solution assessment for transfer procedure, with no similarity learning capability, are amply revealed.

VII. Conclusion

In this paper, we have proposed a novel evolutionary computation paradigm inspired by observed human-like problem-solving capabilities of seamlessly applying the knowledge acquired from previous experiences to new and more challenging tasks. As in machine learning, where the concept of transfer learning allows data from related source tasks to be reused for improving predictive performance on the target task, the goal of this paper is to develop a theoretically principled realization of black-box *transfer optimization*.

To elaborate, our proposal enables online learning and exploitation of source–target similarities, potentially revealing latent synergies even during the course of the optimization search. In our proposal, we encode the knowledge acquired from past optimization exercises in the form of probabilistic models that bias the search toward promising solutions. The modulation of the amount of transfer between multiple sources and the target task of interest is achieved through an AMT procedure, where automatic learning of the optimal blend of source and target probabilistic models is carried out via *stacked density estimation*. Notably, the method eliminates the need for any human intervention or ad-hoc rules for ascertaining source–target similarities. We introduce an instantiation of the framework as a nested module within a CEA, which we label as an AMTEA. Subsequently, a theoretical analysis is provided to substantiate the impact of the proposal in facilitating improved performance of the transfer optimization algorithm.

In addition to presenting theoretical justifications, a series of numerical experiments on benchmark and practical examples, covering discrete and continuous domains, as well as single-objective and MOO, were carried out to test the efficacy of the AMTEA. The results showed that while existing techniques (including those endowed with the scope of knowledge transfer) often suffered from premature convergence at local optima of complex tasks, the AMTEA was able to automatically distinguish positive and negative transfers when faced with multiple sources, thereby leading to consistently superior performance.

In the future, one of our main focuses will be on further generalizing the approach to encompass an even larger variety of practically relevant optimization problems. Furthermore, system-level implementations will be considered to assess the scalability of the algorithm for potential deployment in large-scale IoT/cloud-based applications.

References

[1] S. Thrun, “Is learning the n-th thing any easier than learning the first?” in *Proc. Adv. Neural Inf. Process. Syst.*, Denver, CO, USA, 1996, pp. 640–646.

[2] S. J. Pan and Q. Yang, “A survey on transfer learning,” *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.

[3] K. Weiss, T. M. Khoshgoftaar, and D. Wang, “A survey of transfer learning,” *J. Big Data*, vol. 3, no. 1, p. 9, 2016.

[4] A. T. W. Min, R. Almandoz, A. Gupta, and O. Y. Soon, “Coping with data scarcity in aircraft engine design,” in *Proc. AIAA Multidisciplinary Anal. Optim. Conf.*, 2017, pp. 1–9.

[5] Y. Hou, Y. Ong, L. Feng, and J. M. Zurada, “An evolutionary transfer reinforcement learning framework for multiagent systems,” *IEEE Trans. Evol. Comput.*, vol. 21, no. 4, pp. 601–615, Aug. 2017.

[6] S. J. Louis and J. McDonnell, “Learning with case-injected genetic algorithms,” *IEEE Trans. Evol. Comput.*, vol. 8, no. 4, pp. 316–328, Aug. 2004.

[7] A. Gupta, Y.-S. Ong, and L. Feng, “Insights on transfer optimization: Because experience is the best teacher,” *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 2, no. 1, pp. 51–64, Feb. 2018.

[8] A. T. W. Min, R. Sagarna, A. Gupta, Y.-S. Ong, and C. K. Goh, “Knowledge transfer through machine learning in aircraft design,” *IEEE Comput. Intell. Mag.*, vol. 12, no. 4, pp. 48–60, Nov. 2017.

[9] P. Cunningham and B. Smyth, “Case-based reasoning in scheduling: Reusing solution components,” *Int. J. Prod. Res.*, vol. 35, no. 11, pp. 2947–2962, 1997.

[10] M. Kaedi and N. Ghasem-Aghaee, “Biasing Bayesian optimization algorithm using case based reasoning,” *Knowl. Based Syst.*, vol. 24, no. 8, pp. 1245–1253, 2011.

[11] M. Hauschild and M. Pelikan, “An introduction and survey of estimation of distribution algorithms,” *Swarm Evol. Comput.*, vol. 1, no. 3, pp. 111–128, 2011.

[12] L. Feng, Y.-S. Ong, A.-H. Tan, and I. W. Tsang, “Memes as building blocks: A case study on evolutionary optimization+ transfer learning for routing problems,” *Memetic Comput.*, vol. 7, no. 3, pp. 159–180, 2015.

[13] 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.

[14] M. Salamó and M. López-Sánchez, “Adaptive case-based reasoning using retention and forgetting strategies,” *Knowl. Based Syst.*, vol. 24, no. 2, pp. 230–247, 2011.

[15] R. Meuth, M.-H. Lim, Y.-S. Ong, and D. C. Wunsch, “A proposition on memes and meta-memes in computing for higher-order learning,” *Memetic Comput.*, vol. 1, no. 2, pp. 85–100, 2009.

[16] A. Gupta, Y.-S. Ong, L. Feng, and K. C. Tan, “Multiobjective multifactorial optimization in evolutionary multitasking,” *IEEE Trans. Cybern.*, vol. 7, no. 7, pp. 1652–1665, Jul. 2017.

[17] Y.-S. Ong, M. H. Lim, and X. Chen, “Memetic computation—Past, present & future [research frontier],” *IEEE Comput. Intell. Mag.*, vol. 5, no. 2, pp. 24–31, May 2010.

[18] X. Chen, Y.-S. Ong, M.-H. Lim, and K. C. Tan, “A multi-facet survey on memetic computation,” *IEEE Trans. Evol. Comput.*, vol. 15, no. 5, pp. 591–607, Oct. 2011.

[19] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, “How transferable are features in deep neural networks?” in *Proc. NIPS*, Montreal, QC, Canada, 2014, pp. 3320–3328.

[20] S. Thrun and T. M. Mitchell, “Learning one more thing,” in *Proc. IJCAI*, 1995, pp. 1217–1225.

[21] S. Israel and A. Moshaiov, “Bootstrapping aggregate fitness selection with evolutionary multi-objective optimization,” in *Proc. Parallel Prob. Solving Nat. PPSN XII*, Taormina, Italy, 2012, pp. 52–61.

[22] B. Koçer and A. Arslan, “Genetic transfer learning,” *Expert Syst. Appl.*, vol. 37, no. 10, pp. 6997–7002, 2010.

[23] L. Feng, Y.-S. Ong, M.-H. Lim, and I. W. Tsang, “Memetic search with interdomain learning: A realization between CVRP and CARP,” *IEEE Trans. Evol. Comput.*, vol. 19, no. 5, pp. 644–658, Oct. 2015.

[24] M. Iqbal, W. N. Browne, and M. Zhang, “Reusing building blocks of extracted knowledge to solve complex, large-scale Boolean problems,” *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 465–480, Aug. 2014.

[25] M. Iqbal, W. N. Browne, and M. Zhang, “Extracting and using building blocks of knowledge in learning classifier systems,” in *Proc. ACM 14th Annu. Conf. Genet. Evol. Comput.*, Philadelphia, PA, USA, 2012, pp. 863–870.

[26] I. M. Alvarez, W. N. Browne, and M. Zhang, "Human-inspired scaling in learning classifier systems: Case study on the n-bit multiplexer problem set," in *Proc. ACM Genet. Evol. Comput. Conf.*, Denver, CO, USA, 2016, pp. 429–436.

[27] K. Swersky, J. Snoek, and R. P. Adams, "Multi-task Bayesian optimization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 2004–2012.

[28] E. V. Bonilla, K. M. Chai, and C. Williams, "Multi-task Gaussian process prediction," in *Proc. Adv. Neural Inf. Process. Syst.*, 2008, pp. 153–160.

[29] A. W. M. Tan, Y.-S. Ong, A. Gupta, and C.-K. Goh, "Multi-problem surrogates: Transfer evolutionary multiobjective optimization of computationally expensive problems," *IEEE Trans. Evol. Comput.*, to be published.

[30] E. Snelson and Z. Ghahramani, "Sparse Gaussian processes using pseudo-inputs," in *Proc. Adv. Neural Inf. Process. Syst.*, Vancouver, BC, Canada, 2006, pp. 1257–1264.

[31] M. Zaefferer and T. Bartz-Beielstein, "Efficient global optimization with indefinite kernels," in *Proc. Int. Conf. Parallel Prob. Solving Nat.*, 2016, pp. 69–79.

[32] K. Sastry, D. E. Goldberg, and X. Llorà, "Towards billion-bit optimization via a parallel estimation of distribution algorithm," in *Proc. GECCO ACM*, 2007, pp. 577–584.

[33] P. Smyth and D. Wolpert, "Stacked density estimation," in *Proc. NIPS*, 1997, pp. 668–674.

[34] A. Gupta, Y.-S. Ong, and L. Feng, "Multifactorial evolution: Toward evolutionary multitasking," *IEEE Trans. Evol. Comput.*, vol. 20, no. 3, pp. 343–357, Jun. 2016.

[35] Y.-S. Ong and A. Gupta, "Evolutionary multitasking: A computer science view of cognitive multitasking," *Cogn. Comput.*, vol. 8, no. 2, pp. 125–142, 2016.

[36] J. C. Bean, "Genetic algorithms and random keys for sequencing and optimization," *ORSA J. Comput.*, vol. 6, no. 2, pp. 154–160, 1994.

[37] T. K. Moon, "The expectation-maximization algorithm," *IEEE Signal Process. Mag.*, vol. 13, no. 6, pp. 47–60, Nov. 1996.

[38] M. Blume, *Expectation Maximization: A Gentle Introduction*, Inst. Comput. Sci., Tech. Univ. Munich, Munich, Germany, 2002.

[39] Q. Zhang and H. Muhlenbein, "On the convergence of a class of estimation of distribution algorithms," *IEEE Trans. Evol. Comput.*, vol. 8, no. 2, pp. 127–136, Apr. 2004.

[40] L. Devroye, L. Györfi, and G. Lugosi, *A Probabilistic Theory of Pattern Recognition*, vol. 31. New York, NY, USA: Springer, 2013.

[41] S. Kullback and R. A. Leibler, "On information and sufficiency," *Anna. Math. Stat.*, vol. 22, no. 1, pp. 79–86, 1951.

[42] D. J. MacKay, *Information Theory, Inference and Learning Algorithms*. Cambridge, U.K.: Cambridge Univ. Press, 2003.

[43] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *J. Roy. Stat. Soc. B (Methodol.)*, vol. 39, no. 1, pp. 1–38, 1977.

[44] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.

[45] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712–731, Dec. 2007.

[46] D. Wierstra *et al.*, "Natural evolution strategies," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 949–980, 2014.

[47] T. Salimans, J. Ho, X. Chen, and I. Sutskever, "Evolution strategies as a scalable alternative to reinforcement learning," *arXiv preprint arXiv:1703.03864*, 2017.

[48] H. Mühlenbein, "The equation for response to selection and its use for prediction," *Evol. Comput.*, vol. 5, no. 3, pp. 303–346, 1997.

[49] R. B. Agrawal, K. Deb, and R. Agrawal, "Simulated binary crossover for continuous search space," *Complex Syst.*, vol. 9, no. 2, pp. 115–148, 1995.

[50] K. Deb and D. Deb, "Analysing mutation schemes for real-parameter genetic algorithms," *Int. J. Artif. Intell. Soft Comput.*, vol. 4, no. 1, pp. 1–28, 2014.

[51] R. Storn and K. Price, "Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces," *J. Glob. Optim.*, vol. 11, no. 4, pp. 341–359, 1997.

[52] K. Deb and D. E. Goldberg, "Analyzing deception in trap functions," in *Proc. 2nd Workshop Found. Genet. Algorithms*, Vail, CO, USA, Jul. 1992, pp. 93–108. [Online]. Available: https://dblp.org/rec/bib/conf/foga/DebG92

[53] Z. Michalewicz and J. Arabas, "Genetic algorithms for the 0/1 knapsack problem," in *Proc. Methodol. Intell. Syst.*, Charlotte, NC, USA, 1994, pp. 134–143.

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

[55] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable test problems for evolutionary multiobjective optimization," in *Evolutionary Multiobjective Optimization. Theoretical Advances and Applications*. London, U.K.: Springer, 2005, pp. 105–145.

[56] S. Huband, P. Hingston, L. Barone, and R. L. While, "A review of multiobjective test problems and a scalable test problem toolkit," *IEEE Trans. Evol. Comput.*, vol. 10, no. 5, pp. 477–506, Oct. 2006.

[57] S. Jiang, Y.-S. Ong, J. Zhang, and L. Feng, "Consistencies and contradictions of performance metrics in multiobjective optimization," *IEEE Trans. Cybern.*, vol. 44, no. 12, pp. 2391–2404, Dec. 2014.

[58] F. J. Gomez and R. Miikkulainen, "Solving non-Markovian control tasks with neuroevolution," in *Proc. IJCAI*, vol. 99. Stockholm, Sweden, 1999, pp. 1356–1361.

[59] A. P. Wieland, "Evolving neural network controllers for unstable systems," in *Proc. IEEE Neural Netw. Seattle Int. Joint Conf. (IJCNN)*, vol. 2. Seattle, WA, USA, 1991, pp. 667–673.

[60] F. Gomez, J. Schmidhuber, and R. Miikkulainen, "Accelerated neural evolution through cooperatively coevolved synapses," *J. Mach. Learn. Res.*, vol. 9, pp. 937–965, May 2008.

Bingshui Da received the B.Eng. degree in electronic engineering from the School of Information Science and Technology, University of Science and Technology of China, Hefei, China, in 2014. He is currently pursing the Ph.D. degree in artificial intelligence with the School of Computer Science and Engineering, Nanyang Technological University, Singapore.

He is a Research Assistant with SAP Machine Learning, Leonardo, Singapore. His current research interests include evolutionary computation, transfer learning, and Gaussian process.

Abhishek Gupta received the Ph.D. degree in engineering science from the University of Auckland, Auckland, New Zealand, in 2014.

He is currently a Research Scientist with the School of Computer Science and Engineering, Nanyang Technological University, Singapore. He has diverse research experiences in computational science, ranging from numerical methods in engineering physics, to topics in computational intelligence. His current research interests include development of memetic computing as an approach for automatic learning and transfer of knowledge across optimization problems, with applications in design.

Yew-Soon Ong (F'18) received the Ph.D. degree in artificial intelligence in complex design from the Computational Engineering and Design Center, University of Southampton, Southampton, U.K., in 2003.

He is a Professor of the School of Computer Science and Engineering, Nanyang Technological University, Singapore, where he is also the Director of the Data Science and Artificial Intelligence Research Center and the Principal Investigator of the Data Analytics and Complex Systems Programme with the Rolls-Royce@NTU Corporate Lab, Singapore. His current research interests include computational intelligence spans across memetic computing, complex design optimization, and big data analytics.

Dr. Ong is the founding Editor-in-Chief of the IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTATIONAL INTELLIGENCE, an Associate Editor of the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, and the IEEE TRANSACTIONS ON CYBERNETICS.