Honda Research Institute EU
Honda Research Institute Europe GmbH
https://www.honda-ri.de/

Transfer Learning Based Surrogate Assisted Evolutionary Bi-objective Optimization for Objectives with Non-uniform Evaluation Times

Xilu Wang, Yaochu Jin, Sebastian Schmitt, Markus Olhofer, Richard Allmendinger

2021

Preprint:

Copyright Elsevier 2021. This manuscript version is made available under the CC-BY-NC-ND 4.0 license http://creativecommons.org/licenses/by-nc-nd/4.0/

Transfer Learning Based Surrogate Assisted Evolutionary Bi-objective Optimization for Objectives with Different Evaluation Times

Xilu Wang, Yaochu Jin, *Fellow*, *IEEE*, Sebastian Schmitt, Markus Olhofer and Richard Allmendinger, *Member*, *IEEE*

***Abstract*—Various multiobjective optimization algorithms have been proposed with a common assumption that the evaluation of each objective function takes the same period of time. Little attention has been paid to more general and realistic optimization scenarios where different objectives are evaluated by different computer simulations or physical experiments with different time complexities (latencies) and only a very limited number of function evaluations is allowed for the slow objective. In this work, we investigate benchmark scenarios with two objectives. We propose a transfer learning scheme within a surrogate-assisted evolutionary algorithm framework to augment the training data for the surrogate for the slow objective function by transferring knowledge from the fast one. Specifically, a hybrid domain adaptation method aligning the second-order statistics and marginal distributions across domains is introduced to generate promising samples in the decision space according to the search experience of the fast one. A Gaussian process model based co-training method is adopted to predict the value of the slow objective and those having a high confidence level are selected as the augmented synthetic training data, thereby enhancing the approximation quality of the surrogate of the slow objective. Our experimental results demonstrate that the proposed algorithm outperforms existing surrogate and non-surrogate-assisted delay-handling methods on a range of bi-objective optimization problems. The approach is also more robust to varying levels of latency and correlation between the objectives.**

***Index Terms*—Bi-objective optimization, different evaluation times, latency, transfer learning, domain adaptation, co-training, surrogate-assisted evolutionary algorithm, Bayesian optimization.**

I. Introduction

VARIOUS multi-objective evolutionary algorithms (MOEAs) have been developed to solve multi-objective optimization problems (MOPs) [1]–[3]. However, MOEAs usually require a large number of objective function evaluations (FEs) to generate satisfying approximations to Pareto fronts (PFs), indicating the difficulty to handel computationally expensive MOPs where FEs involve computationally intensive simulations or costly physical experiments, e.g., aerodynamic design optimization [4] and structural optimization [5]. Surrogate-assisted evolutionary algorithms (SAEAs) have emerged to be an effective methodology to overcome the computational obstacle for applying MOEAs to computationally expensive MOPs [6], [7]. Generally, SAEAs construct cheap surrogate models using historical data to approximate the original expensive objective functions, then the surrogates will replace in part the objective functions in evaluating the candidate solutions, thereby reducing the computational cost. A crucial issue in SAEAs is how to properly use surrogates and expensive objective functions together to improve the efficiency of the optimization, known as model management [8]. Various kinds of classification or regression models are employed as surrogates in SAEAs, such as support vector machines (SVMs) [9], radial basis function networks [10], and Gaussian processes (GPs), also known as Kriging [11], [12]. Among them, SAEAs that use the GP as the surrogate and infill criteria, also known as acquisition functions (AFs), for model management originated from efficient global optimization [13] or Bayesian optimization [14], have empirically shown promising in many studies [15].

Most MOEAs and SAEAs share a common assumption that the computational complexities of all objective functions are similar. Consequently, it is assumed that the evaluation of different objectives of a candidate solution is completed at the same time, before the selection operator of an MOEA can be conducted and the evolutionary search can proceed to the next generation. However, it is common in many real-world applications that different objective functions require different evaluation times (also referred to as latencies [16]) due to differences in their computational complexities. For example, fluid dynamics simulations in airfoil shape optimization are computationally expensive [17], while others are not, such as the calculation of the weight or area of the designed structure. Such MOPs are first considered by Allmendinger *et al.* in [18], and the authors further provided a general problem definition for MOPs with non-uniform evaluation times (NET-MOPs) in [16]. As done in [16], [18]–[20], we consider an expensive bi-objective optimization problem where one objective function is computationally cheap while the other is computationally expensive, NET-BOPs by short. To simulate NET-BOPs, we use the notations and assumptions as initially proposed in [16], [19], which we recap here for completeness reasons;

Xilu Wang and Yaochu Jin are with the Department of Computer Science, University of Surrey, Guildford, GU2 7XH, United Kingdom. Email: {xilu.wang;yaochu.jin}@surrey.ac.uk. (*Corresponding author: Yaochu Jin*)

Sebastian Schmitt and Markus Olhofer are with the Honda Research Institute Europe GmbH, Carl-Legien-Strasse 30, D-63073 Offenbach/Main, Germany. Email: {sebastian.schmitt;markus.olhofer}@honda-ri.de.

Richard Allmendinger is with the Alliance Manchester Business School, University of Manchester, Booth Street West, Manchester M15 6PB, UK. Email: richard.allmendinger@manchester.ac.uk.

Manuscript received December 6, 2018; revised xx, 2019.

the notations will also be to discuss related research and the proposed method for dealing with NET-BOPs.

- The slow (or delayed/expensive) objective function is denoted as f_s, while the fast (or non-delayed/cheap) one is denoted as f_f, and the corresponding surrogates for the two objective functions are denoted as GP_s and GP_f, respectively. Consequently, the slow and fast objective values of solutions $\boldsymbol{X}$ are denoted by $\boldsymbol{Y}_s$ and $\boldsymbol{Y}_f$, respectively. The ratio between the time duration for evaluating f_s to that for f_f is denoted by an integer $\tau > 1$, i.e. evaluating f_s is τ times slower/more expensive than evaluating f_f. Due to the distinct per-objective FE budgets, additional solutions $\boldsymbol{X}_f^{new}$ can be evaluated on f_f to make full use of the time during which the slow objective is being evaluated, thereby obtaining additional data $(\boldsymbol{X}_f^{new}, \boldsymbol{Y}_f^{new})$. We assume that the evaluation of the fast and slow objective functions can be done in parallel.
- It is assumed that the computation time for building surrogates and for implementing the genetic operators is negligible compared to that for evaluating the expensive objectives. Consequently, the total computation time available for solving the NET-BOPs is defined by the total budget for FEs. Specifically, the total budget is defined as the maximum number of FEs for f_s, denoted as FE_s^{max}. Consequently, the allowed maximum budget for f_f is $FE_f^{max} = FE_s^{max} * \tau$, as we assume that the evaluation of the two objectives can always be carried out in parallel.

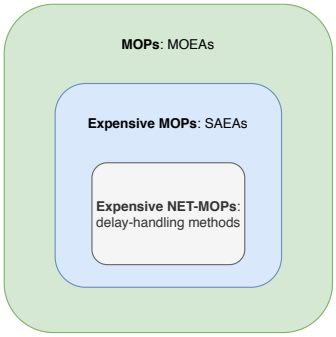

Fig. 1. The problem categorization and the corresponding solution categorization.

In expensive NET-BOPs [16], [18]–[20], given a fixed time period, the affordable number of FEs on each objective varies considerably, posing additional challenges to the development of efficient MOEAs and SAEAs. As shown in [16], [18], the presence of non-uniform per-objective evaluation times poses a number of challenges to an EA with performance being affected by the choice of the selection and recombination operators. In the context of SAEAs, varying per-objective evaluation times require to reconsider that way training data is selected and surrogate models are updated [19], [20]. Moreover, for a range of benchmark problems, it has also been shown [16], [18]–[20] that algorithm performance is a function of the problem at hand, in particular, the time lag between the slow and fast objective (as defined by τ) and the correlation between the objectives. These factors make the design and validation of effective optimization methods, be it surrogate or non-surrogate based methods, an interesting yet challenging research topic that largely remains unexplored. So far, however, only a few studies have been conducted on studying the impact of delayed objective functions in MOPs. Motivated by real-world problems in engineering and science, Allmendinger and Knowles [18] first introduced an MOP scenario where at least one objective function is subject to a longer evaluation time compared to the others. To enhance the performance in solving NET-BOPs, new assignment strategies, ranking metrics and parental selection have been designed. To more closely exploit the effects of the delayed objective functions, Allmendinger *et al.* [16] continued their work by giving a detailed problem definition and notations in the framework of MOEAs. Three distinct schemes are proposed, termed *Waiting*, *Fast-first* and *Interleaving schemes*. Intuitively, *Waiting* refers to selecting and updating the population once the evaluations on all objectives are completed so as to directly apply a standard MOEA. By contrast, *Fast-first* uses a standard single-objective EA (SOEA) to optimize the fast objective f_f for most of the optimization run, and uses the slow objective f_s only at the end of the run to ensure that at least some of the solutions are evaluated on all (both) objectives. This, however, introduces a search bias towards the fast objective, meaning only part of the Pareto front may be discovered, especially when the objectives are negatively correlated. *Interleaving schemes* are more complicated methods that are purposely designed for problems with different per-objective evaluation times. Interleaving methods coordinate the evaluation of the objectives during search so as to leverage the per-objective evaluation budgets as efficiently as possible. Two examples of interleaving methods were proposed in [16], varying in the level of focus put on optimizing the fast objective: While *Brood interleaving* was less focused on optimizing the fast objective but more on maintaining diversity, *Speculative interleaving* used an SOEA to optimize the fast objective while the slow objective was being evaluated.

Such approaches are carried out using MOEAs and have shed light on how to address NET-BOPs; however, they do not consider surrogate models that are competitive to address expensive MOPs with limited number of FEs. The first attempt to investigate the use of surrogates for solving NET-BOPs is based on the Kriging-assisted EA (K-RVEA) [11], called HK-RVEA [19]. HK-RVEA is an Interleaving scheme embedded in a within an SAEA framework. First, when evaluating the initial population on f_s and f_f, an SOEA is adopted to sample more training data from f_f while waiting for evaluations on f_s. Second, additional evaluations on f_f are available when new samples are determined to be evaluated on both the true objective functions. Thus, HK-RVEA adopts genetic operators to produce additional solutions to be evaluated on f_f, making a full use of its evaluation budget. Wang *et al.* extended an SAEA by applying transfer learning (TL) techniques to NET-BOPs and proposed a new SAEA along with a parameter-based TL strategy, called (T-SAEA) [20]. To transfer knowledge acquired from the search experience of f_f for enhancing the performance of GP_s, T-SAEA identifies common decision variables related to both f_f and f_s by means of a filter-based feature selection. Subsequently, corresponding

parameters in the surrogates can be shared to improve the quality of GP_s.

Some research areas are related to handling NET-MOPs using GPs, including co-kriging [21] in multi-fidelity optimization [22], multi-task GPs (MTGPs) in multi-task learning [23], and asynchronous GPs [24], [25]. While in NET-MOPs different objectives require distinct period of time for their FEs, in multi-fidelity optimization both a high fidelity (accurate but costly) and a low fidelity (coarse but quick) model are available for FEs of one objective. Multi-task learning involves a set of different but related tasks, so that the MTGP is a multi-output model with a augmented dataset of these different tasks; however, NET-MOPs aims to optimise competing objectives of the same task. Note that Luo *et al.* made attempts to adopt MTGPs to solve MOPs by sharing knowledge between subproblems [26], but this method can not be applied to NET-MOPs due to the extra FEs available for f_f. Finally, asynchronous GPs deals with the scenario where solutions are evaluated asynchronously in a parallel computing environment. However, unlike to NET-MOPs, asynchronous GPs assume that the asynchronouity is across the evaluation of entire solutions (i.e. per-solution) and not across the objectives of a solution.

In this work, we propose and validate a more efficient approach to exploit the transferable knowledge from the additional samples for f_f by integrating a TL strategy into a GP based SAEA for solving NET-BOPs, termed Tr-SAEA. Recall that more new samples $\boldsymbol{X}_f^{new}$ on the fast objective f_f can be collected than the slow objective owing to the big computational difference between f_f and f_s, when we use AFs to select samples to be evaluated using the true objective functions. We hypothesize that samples selected according to an AF for the fast objective f_f are not only helpful for the optimisation of f_f, but also beneficial for the optimisation of f_s. The reason for this hypothesis is that there is a functional relationship between the values of f_f and f_s (usually a trade-off relation) for solutions on or near the Pareto front [27]. Consequently, it makes sense to leverage knowledge readily available for f_f, i.e., the label-rich source domain in the light of TL, to improve the learning performance of f_s, which is the label-scarce target domain f_s. The key contributions of the proposed Tr-SAEA can be summarized as follow:

1) A TL strategy inspired by ideas from domain adaptation is proposed to reduce the difference between the source domain f_f and and target domain f_s by aligning their subspaces in terms of the second-order statistics and minimizing the distance between the source and target distributions. Such a hybrid domain adaptation method (HDA) allows us to leverage the additional data available for f_f to generate promising candidate solutions for f_s.
2) A GP-based co-training method inspired from semi-supervised learning (SSL) is introduced to predict the value of f_s for the promising candidate solutions identified by HDA. Those solutions whose predicted f_s value has a high level of confidence are then used as the augmented synthetic data for training the surrogate for the slow objective GP_s. This way, the performance of GP_s can be enhanced and the search of f_s can be accelerated.
3) The experimental study validates the proposed methods on different bi-objective benchmark problems subject to non-homogenous per-objective evaluation times, their performance is compared against existing methods for NET-BOPs, and we investigate their sensitivity to different algorithm parameter settings and influential problem settings (correlation between objectives).

The next section will cover preliminaries necessary understand the various concepts related to surrogate-based optimization algorithms applied to problems with varying per-objective evaluation times. Following this, we introduce Tr-SAEA (Section III), and provide details about the experimental settings (Section IV) as used in the experimental study (Section V). Finally, we will draw conclusions from our analysis and discuss future research (Section VI).

II. Preliminaries

A. Bayesian Optimization

As we mentioned above, Tr-SAEA adopts GPs as the surrogates and an AF as the model management strategy. In the following, we briefly introduce GPs and AFs in the context of Bayesian optimization.

Bayesian optimization [13], [14] typically models an unknown objective function using a probabilistic surrogate model, the GP. The GP is characterized by its prior mean function and covariance function (or kernel matrix). Consider a finite collection of data pairs of the unknown function $(\boldsymbol{X}, \boldsymbol{Y})$, where $\boldsymbol{X} = [\boldsymbol{x}^1, \boldsymbol{x}^2, ...\boldsymbol{x}^N]^T$, $\boldsymbol{Y} = [\boldsymbol{y}^1, \boldsymbol{y}^2, ...\boldsymbol{y}^N]^T$, we assume that the observed data are drawn from a multivariate Gaussian distribution with a Gaussian prior. Therefore, let $\boldsymbol{x}^{new}$ denote a new data point, the random variable $f(\boldsymbol{x}^{new})$ is also normal distributed with the following posterior mean and variance functions

$$\mu(\boldsymbol{x}^{new}) = \hat{\mu} + \boldsymbol{r}^T \boldsymbol{C}^{-1}(\mathbf{y} - \mathbf{1}\hat{\mu}) \tag{1}$$

$$\sigma(\boldsymbol{x}^{new})^2 = \hat{\sigma}^2[1 - \boldsymbol{r}^T \mathbf{C}^{-1}\boldsymbol{r} + \frac{(1 - \boldsymbol{r}^T \boldsymbol{C}^{-1}\boldsymbol{r})^2}{\mathbf{1}^T \boldsymbol{C}^{-1}\mathbf{1}}] \tag{2}$$

where $\boldsymbol{r} = (\mathrm{Corr}(\boldsymbol{x}^{new}, \boldsymbol{x}^1), ..., \mathrm{Corr}(\boldsymbol{x}^{new}, \boldsymbol{x}^N))^T$ presents a correlation vector between $\boldsymbol{x}^{new}$ and each element $\boldsymbol{x}^i$ in $\boldsymbol{X}$, $\hat{\mu}$ and $\hat{\sigma}^2$ are the estimations for the true values μ and σ, respectively, which will be obtained

$$\hat{\mu} = \frac{\mathbf{1}^T \boldsymbol{C}^{-1} \boldsymbol{y}}{\mathbf{1}^T \boldsymbol{C}^{-1} \mathbf{1}} \tag{3}$$

$$\hat{\sigma}^2 = \frac{(\boldsymbol{y} - \mathbf{1}\hat{\mu})^T \boldsymbol{C}^{-1}(\boldsymbol{y} - \mathbf{1}\hat{\mu})}{N} \tag{4}$$

where $\mathbf{1}$ denotes an $N \times 1$ column vector of ones, $\boldsymbol{C}$ denotes an $N \times N$ correlation matrix, defined as follows:

$$\boldsymbol{C} = \begin{pmatrix} \mathrm{Corr}(\boldsymbol{x}^1, \boldsymbol{x}^1) & \cdots & \mathrm{Corr}(\boldsymbol{x}^1, \boldsymbol{x}^N) \\ \vdots & \ddots & \vdots \\ \mathrm{Corr}(\boldsymbol{x}^N, \boldsymbol{x}^1) & \cdots & \mathrm{Corr}(\boldsymbol{x}^N, \boldsymbol{x}^N) \end{pmatrix} \tag{5}$$

where $\mathrm{Corr}(\boldsymbol{x}^i, \boldsymbol{x}^j)$ denotes the used covariance (kernel) function. Commonly used covariance functions include the squared exponential and Matérn covariance functions [14].

B. Domain Adaptation

Many machine learning methods require a massive amount of labeled data to achieve good performance; however, acquiring sufficient labeled data is often prohibitive in many application scenarios [28], such as computer vision [29] and software defect classification [30]. To address such issues, domain adaptation (DA), one of the branches of TL, has been established to reduce the need for costly labeled data in the target domain by leveraging abundant labeled data readily available in related source domains. A simple yet effective DA method is the correlation alignment (CORAL) [31], aligning both domains in terms of the second-order statistic feature. However, CORAL does not consider the distribution alignment. To reduce the difference in the distributions of source and target domains, the transfer component analysis (TCA) proposed by Pan *et al.* [32] exploits a set of common transfer components that have the same marginal distribution across both domains and preserve the structure of the original data.

DA and TL have attracted growing attention in the machine learning community in recent years [33], and some TL techniques also have been used to solve problems in the context of evolutionary computation. For example, autoencoder has been used to conduct knowledge transfer in evolutionary multitasking [34]. DA techniques, such as TCA and TrAdaBoost, are employed to reuse past experience for solving dynamic MOPs [35], [36]. In the context of computationally expensive MOPs, Le *et al.* [37] proposed cross surrogates as local search to transfer knowledge between objectives; however, they did not consider NET-MOPs. These existing works show promising performance on different optimization scenarios and provide new insights and inspiration to the evolutionary computation community, encouraging us to apply TL techniques to NET-BOPs. The main issues, how to efficiently capture the useful knowledge from the search experience of f_f and how to transfer the knowledge to benefit the optimization of f_s, need to be carefully addressed to guard against negative transfer where the transferred knowledge has a detrimental effect on a target learner [28]. The negative transfer in the context of evolutionary computation will hurt the performance of surrogates, and subsequently mislead the search direction of the population. In this regard, some approaches seek to find out related tasks by measuring the similarity using Kullback-Leibler divergence [38] or a linear meta-regression [39]. However, this becomes more challenging for NET-BOPs since there are only two objective functions, so the fast objective function is the only source task available for the slow objective function to benefit from knowledge transfer.

C. Co-training

Co-training [40] is one of the representative semi-supervised learning (SSL) methods, aiming to use the predicted target label to train an accurate model. Co-training approaches typically train two distinct classifiers with the help of multi-view learning, thereby enabling each classifier to predict the labels for the unlabeled target data from different views. Subsequently, the predictions of each classifier are selected according to their confidence level and added into the training set of the other. This way the scarce target data can be augmented by utilizing the unlabeled data. In the field of SAEA, only few works have been adopted SSL techniques to make use of unlabeled data [41]. For example, having observed that the presence of much unlabeled data can lead to a lack of exploitation, Sun *et al.* [41] augmented an SSL strategy onto an SAEA to enhance the training of the surrogate model using both labeled and unlabeled data, while most SAEAs only involve the limited number of samples evaluated by the expensive objective functions. Similarly, when dealing with NET-BOPs, there is a large amount of unlabelled data for f_s resulting from the heterogeneous evaluations times and the transfer learning, which may contain useful knowledge but has not been explored yet.

III. Proposed Algorithm

In this section, we introduce the proposed approach, Tr-SAEA [1]. It uses the hybrid domain adaptation method (HDA) to generate samples that are promising for transferring knowledge from the fast objective function f_f (source domain) to the slow objective function f_s (target domain), and then a modified co-training method is adopted to augment training data for training the surrogate for f_s. Before giving a detailed description of Tr-SAEA, we will introduce a few related notations at first.

A. Notations

- Two GPs for f_s, denoted as GP_{s1} and GP_{s2}, are constructed in the co-training paradigm in Tr-SAEA. The initial population $\boldsymbol{X}$ is evaluated on both f_f and f_s.
- While a very limited number of new samples $D_s^{new} = (\boldsymbol{X}_s^{new}, \boldsymbol{Y}_s^{new})$ is affordable for f_s, more new samples $D_f^{new} = (\boldsymbol{X}_f^{new}, \boldsymbol{Y}_f^{new})$ are allowed to be evaluated on f_f. We apply the proposed HDA method on D_f^{new} so that some promising candidate solutions $\boldsymbol{X}_s^{\prime new}$ for the target domain f_s are generated. Data $\boldsymbol{X}_f^{new}$ and $\boldsymbol{X}_s^{\prime new}$ are combined as a set of promising samples $\boldsymbol{X}_s^a$ for f_s. Note that the slow objective values associated with $\boldsymbol{X}_s^a$ are unknown; hence, a co-training based SSL algorithm is further proposed to leverage these unlabeled target solutions.
- Let $(\boldsymbol{Y}_{s1}^a, \boldsymbol{\sigma}_{s1}^a)$ and $(\boldsymbol{Y}_{s2}^a, \boldsymbol{\sigma}_{s2}^a)$ denote the predictions of GP_{s1} and GP_{s2} on $\boldsymbol{X}_s^a$, respectively, where $\boldsymbol{Y}$ denotes the predicted objective value and $\boldsymbol{\sigma}$ denotes the corresponding a confidence level. Subsequently, each of the regression models GP_{s1} and GP_{s2} will identify a subset of $\mathbf{X}_s^a$ with a higher confidence level (less uncertainty) to update each other. In this way, the chosen subsets denoted as D_{t1} and D_{t2} can augment the training data for f_s, transferring knowledge from f_f to f_s.
- $D_{s1} = (\boldsymbol{X}, \boldsymbol{Y}_s) + D_s^{new} + D_{t1}$ and $D_{s2} = (\boldsymbol{X}, \boldsymbol{Y}_s) + D_s^{new} + D_{t2}$ are defined as the training data sets for GP_{s1} and GP_{s2}, respectively, while $D_f = (\boldsymbol{X}, \boldsymbol{Y}_f) + D_f^{new}$ is defined as the training data set for GP_f.

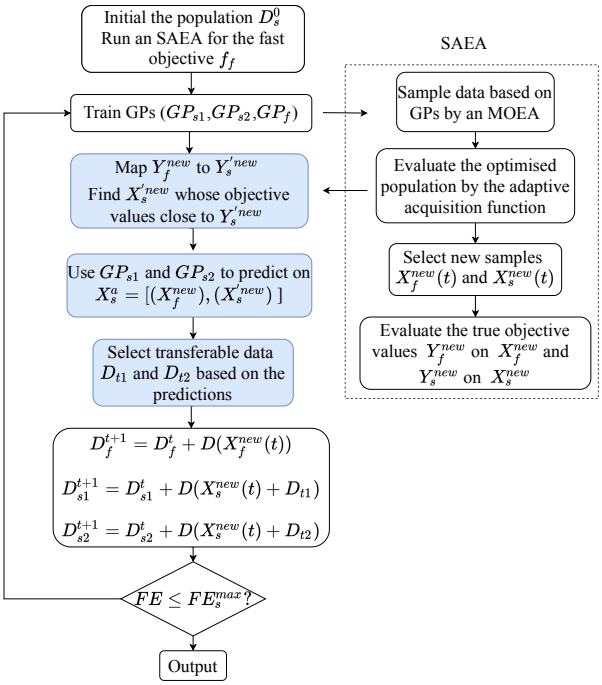

Fig. 2. The overall framework of Tr-SAEA.

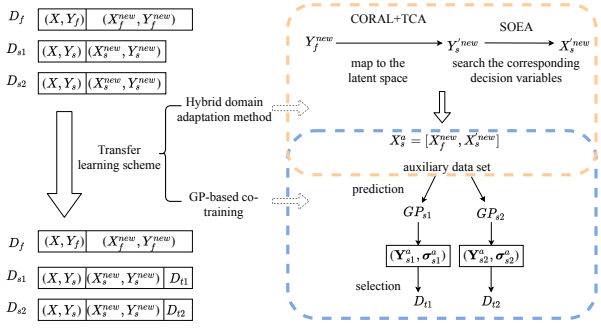

Fig. 3. An illustration of the workflow for data augmentation using the TL scheme, including the HDA and GP-based co-training strategies.

B. Algorithm Framework

Tr-SAEA focuses on how to transfer knowledge embedded in the label-rich source domain f_f to improve the learning of the label-scarce domain f_s in building surrogates. Fig. 2 illustrates the framework of the proposed method and its pseudo code is outlined in Algorithm 1. Fig. 3 presents the workflow of the key components of the data augmentation. To seek answers to the question what to transfer and how to transfer, we design two major components in Tr-SAEA, a hybrid domain adaptation (HDA) method and a co-training mechanism, as described by the following step-by-step procedure. The details of the key components in Tr-SAEA will be presented in the following subsections.

- Step 1: Construct surrogate models. Tr-SAEA begins with sampling two different training data sets (D_f^0 and D_s^0) from f_f and f_s in the same way used in [11], where the Latin hypercube sampling (LHS) is employed to initialize the population and a SOEA is performed to consume the additional evaluation budget available for f_f. Subsequently, while a GP_f is trained with D_f^0 for f_f, GP_{s1} and GP_{s2} using different kernel functions with D_s^0 are generated to implement co-training mechanism for f_s.
- Step 2: Select new samples. Similar to the standard GP-based SAEA, a baseline MOEA (in our case RVEA [2]) is employed to optimize the NET-BOP for a certain number of generations, in which the two objectives are predicted by the GPs instead of using the true objective values. The optimized population will then be evaluated according to the AFF, so that u and $u*\tau$ new samples will be identified to be evaluated using f_f and f_s, respectively, due to the different evaluation times. In this way, we can obtain $D_s^{new} = (\boldsymbol{X}_s^{new}, \boldsymbol{Y}_s^{new})$ and $D_f^{new} = (\boldsymbol{X}_f^{new}, \boldsymbol{Y}_f^{new})$.
- Step 3: Implement HDA in the objective space to align the second-order statistics and marginal distributions of the source domain (f_f) and target domain (f_s). Motivated by the fact that there is a functional relationship between f_s and f_f on and near the PF, we attempt to find a common latent space to minimize the difference between the two domains. Firstly, CORAL is adopted to construct a transformation matrix (denoted as A) to minimize the domain shift by aligning the second-order statistics of both domains. As CORAL does not consider the distribution alignment, TCA is adopted to further discover a feature representation in a latent space having the same marginal distribution across the two domains. Using the joint matching method yields the mapping $\boldsymbol{Y}_s^{'new}$ of $\boldsymbol{Y}_f^{new}$ in the latent domains. Subsequently, an SOEA is adopted to exploit promising decision variables $\boldsymbol{X}_s^{'new}$ whose associated slow objective values are close to $\boldsymbol{Y}_s^{'new}$ in the obtained latent space.
- Step 4: Implement the co-training strategy to identify reliable unlabeled data with regard to the predictions of GP_{s1} and GP_{s2}, which will be then used as labelled data for training GP_{s1} and GP_{s2} together with the real data. Let the set of promising samples be $\boldsymbol{X}_s^a = (\boldsymbol{X}_f^{new}, \boldsymbol{X}_s^{'new})$. The two GPs for f_s separately provide their predictions on $\boldsymbol{X}_s^a$, including the predicted fitness value and a confidence level. Based on the labeling confidence, GP_{s1} and GP_{s2} will select $u*\tau$ unlabeled samples (D_{t1} and D_{t2}) associated with predicted labels from $\boldsymbol{X}_s^a$, and then add these reliable unlabeled samples to their training data sets, respectively.
- Step 5: Update GPs with different training data sets. GP_{s1} and GP_{s2} will be updated with $D_{s1} = (\boldsymbol{X}, \boldsymbol{Y}_s) + D_s^{new} + D_{t1}$ and $D_{s2} = (\boldsymbol{X}, \boldsymbol{Y}_s) + D_s^{new} + D_{t2}$, respectively, while GP_f is updated with $D_f = (\boldsymbol{X}, \boldsymbol{Y}_f) + D_f^{new}$. The process repeats until the maximum number of evaluations is reached.

C. Hybrid Domain Adaptation

In Tr-SAEA, a hybrid domain adaptation method is proposed to use the available data drawn from the source domain (f_f) to generate data in the decision space for the target

[1]The source code is available at https://github.com/xw00616/Tr-SAEA.

Algorithm 1 The framework of Tr-SAEA

Input: FE_s^{max}: the maximum number of the slow objective function evaluations; τ: the ratio of the evaluation times between the slow and fast objectives; u: the number of new samples for updating the GP; w_{max}: the maximum number of generations before updating GPs;
Output: Optimal solutions in D_s;
1: Initialization: Use the Latin Hybercube Sampling method to sample an initial population $\boldsymbol{X}$ to be evaluated on both objective functions; Set $D_{s1}^0 = D_{s2}^0 = (\boldsymbol{X}, \boldsymbol{Y}_s)$ and $D_f^0 = (\boldsymbol{X}, \boldsymbol{Y}_f)$; Run a single-objective EA to optimize f_f and save data in D_f^0; set $FE_s = size(D_s^0)$ and $IterNum = 1$.
2: **while** $FE_s \leqslant FE_s^{max}$ **do**
3: Train GP_f with the training data D_f for f_f;
4: Train GP_{s1} with the training data D_{s1} for f_s;
5: Train GP_{s2} with the training data D_{s2} for f_s;
6: **while** $w \leqslant w_{max}$ **do**
7: //Using the surrogate in the EA//
8: Run RVEA to perform multi-objective optimization based on the GPs;
9: $w = w + 1$;
10: **end while**
11: Use the adaptive acquisition function [12] to evaluate the optimized solutions found by the RVEA;
12: Use APD [2] to determine u and $u * \tau$ new solutions $\boldsymbol{X}_s^{new}$ and $\boldsymbol{X}_f^{new}$ to be evaluated using f_s and f_f, respectively; obtain the new data $D_s^{new} = (\boldsymbol{X}_s^{new}, \boldsymbol{Y}_s^{new})$ and $D_f^{new} = (\mathbf{X}_f^{new}, \mathbf{Y}_f^{new})$;
13: //Using HDA to generate promising data points in the decision space//
14: Implement the HDA method for mapping $\boldsymbol{Y}_f^{new}$ to obtain $\boldsymbol{Y}_s^{'new}$, searching for corresponding $\boldsymbol{X}_s^{'new}$; let $\boldsymbol{X}_s^a = (\boldsymbol{X}_f^{new}, \boldsymbol{X}_s^{'new})$
15: //Using the co-training method to update GPs//
16: Use GP_{s1} and GP_{s2} to predict the objective values of f_s together with the uncertainty for $\boldsymbol{X}_s^a$, and select $u * \tau$ most reliable predictions (D_{t1} and D_{t2}) as labelled data and enlarge the training set of GP_{s1} and GP_{s2}, respectively;
17: Train GP_{s1} and GP_{s2} with $D_{s1} = (\boldsymbol{X}, \boldsymbol{Y}_s) + D_s^{new} + D_{t1}$ and $D_{s2} = (\boldsymbol{X}, \boldsymbol{Y}_s) + D_s^{new} + D_{t2}$, respectively, and train GP_f with $D_f = (\boldsymbol{X}, \boldsymbol{Y}_f) + D_f^{new}$.
18: Update $FE_s = FE_s + u$, $IterNum = IterNum + 1$;
19: **end while**
20: Return the optimized solutions;

domain (f_s) based on the correlation between the two domains. The key motivation is that there is a functional relationship between f_f and f_s when they are closed to the PF [27]. In other words, the closer to the PF, the stronger the functional relationship between f_f and f_s. With this in mind, we resort to DA techniques for using the knowledge about the PF acquired in optimizing f_f to improve the search efficiency in optimizing f_s.

As mentioned in Section 2, various approaches in the DA community have been proposed from different points of view. It is claimed that jointly conducting different approaches across domains can achieve better performance than exploring approaches independently [42]. For example, instance re-weighting methods match the source and target domains by re-weighting the source data; however, there will always exist some of the features that are specific to one domain and irrelevant to the other one. Similarly, it is also common that some source samples are unrelated to the target samples, even in a subspace. For this reason, some joint matching methods for DA have been proposed [42], [43]. Keeping this in mind, we adopt CORAL to align the second-order statistics of both domains due to its simplicity and efficiency, and then employ the TCA to match the marginal distributions of both domains in a Reproducing Kernel Hilbert Space (RKHS). Recall that in this work, the source and target domains are f_f and f_s, respectively, and a domain includes a sample space Z and the corresponding marginal distribution $P(Z)$, where $Z = \{z_1, z_2, \ldots, z_n\}$. Note that we apply the proposed hybrid matching method to each of the obtained objective values (in the objective space); therefore, both the source and target sample spaces $Z_S = \{z_{S_1}, z_{S_2}, \ldots, z_{S_n}\}$ and $Z_T = \{z_{T_1}, z_{T_2}, \ldots, z_{T_n}\}$ are one-dimensional. This is also why we do not adopt other subspace alignment approaches that reduce the dimension of sample space. In the following, a brief introduction of CORAL and TCA is given.

CORAL constructs a transformation matrix A of the source features by minimizing the distance between the second-order statistics across the source and target domains, and the Frobenius norm is adopted as the matrix distance metric, which can be achieved as follows,

$$\min_A \left\| C_{\hat{S}} - C_T \right\|_F^2 = \min_A \left\| A^\top C_S A - C_T \right\|_F^2 \tag{6}$$

where $C_{\hat{S}}$, C_S and C_T denote covariance matrices of the transformed source features, the source features, and target features, respectively, and $\| \cdot \|_F^2$ denotes the squared matrix Frobenius norm. It is noteworthy that, as an alternative of other approaches to domain shift, CORAL avoids subspace projection that can be computationally intensive. Moreover, CORAL is very easy to implement. However, it is not able to account for the distribution alignment.

TCA is adopted to explore a set of common transfer components across the two domains by minimizing the marginal distribution discrepancy in a latent space, while preserving data properties in the original space. To achieve this, TCA minimizes the distance between the means of the source and target data based on the RKHS using the maximum mean discrepancy (MMD) [44] as a marginal distribution measurement criterion, and further enforces the scatter matrix as a constraint [32]. The distance between two distributions $P(Z_S)$ and $P(Z_T)$ can be estimated by MMD in an RKHS [32],

$$\mathrm{Dist}\left(Z_S', Z_T'\right) = \left\| \frac{1}{n_1} \sum_{i=1}^{n_1} \phi\left(z_{S_i}\right) - \frac{1}{n_2} \sum_{i=1}^{n_2} \phi\left(z_{T_i}\right) \right\|_{\mathcal{H}}^2 \tag{7}$$

where Z_S' and Z_T' denote the transformed input sets from the source and target domains, n_1 and n_2 denote the number of samples in each domain, and ϕ denotes the desired transformation mapping. By virtue of the kernel trick [45], we can rewrite (7) as follows:

$$\mathrm{Dist}\left(Z_S', Z_T'\right) = \mathrm{tr}(K \times L) \tag{8}$$

where tr is the trace of the matrix A, and

$$K = \begin{bmatrix} K_{S,S} & K_{S,T} \\ K_{T,S} & K_{T,T} \end{bmatrix} \tag{9}$$

is a $(n_1 + n_2) \times (n_1 + n_2)$ kernel matrix with $k_{i,j} = \kappa\left(z_i, z_j\right) = \phi\left(z_i\right)^T \phi\left(z_j\right)$, where κ refers to a kernel function. $L = [L_{ij}]$ are the corresponding coefficients,

$$L(i,j) = \begin{cases} \frac{1}{n_1 \times n_1}, & z_i, z_j \in Z_S \\ \frac{1}{n_2 \times n_2}, & z_i, z_j \in Z_T \\ -\frac{1}{n_1 \times n_2}, & \text{otherwise.} \end{cases} \tag{10}$$

To effectively solve such a kernel matrix learning problem, TCA introduces a low-rank matrix W, i.e., $(n_1 + n_2) \times m$ matrix, where $m \ll n_1 + n_2$ to reduce the dimension of the latent space, transforming (8) into the following form:

$$\begin{aligned} \text{Dist}\left(Z'_S, Z'_T\right) &= \text{tr}\left(\left(KWW^\top K\right) L\right) \\ &= \text{tr}\left(W^\top KLKW\right) \end{aligned} \tag{11}$$

By adding a regularization term $\text{tr}(\text{W}^\top\text{W})$, the kernel learning problem for TCA then reduces to:

$$\begin{aligned} \min_{W} \quad & \text{tr}\left(W^\top W\right) + \mu\,\text{tr}\left(W^\top KLKW\right) \\ \text{s.t.} \quad & W^\top KHKW = I \end{aligned} \tag{12}$$

where μ is a tradeoff parameter, $I \in \mathbb{R}^{m\times m}$ is an identify matrix, $H = I_{n_1+n_2} - \frac{1}{n_1+n_2}\mathbf{1}\mathbf{1}^\top$, where $\mathbf{1} \in \mathbb{R}^{n_1+n_2}$ is an all-ones column vector. This optimization problem can be reformulated as a trace maximization problem and solved via the method presented in [46].

Here, CORAL cooperates with TCA to construct a mapping matrix $A \times W$, allowing us to transfer some promising samples of the source domain to the target domain for the benefit of training the surrogate and guiding the search of the slow objective function. This can be attributed to the fact that the new samples selected by the AF are expected to improve both the accuracy of models and the efficiency of the optimization. Hence, once $\mathbf{Y}_f^{new}$ is generated, we can map them into the latent space with the help of the mapping matrix $A \times W$ in order to generate useful data $\mathbf{Y}_s^{'new}$ for the target domain. Similar to the method in [47], we further explore the decision space to find new samples $\mathbf{X}_s^{'new}$ and enforce its objective values on f_s to be close to $\mathbf{Y}_s^{'new}$ in the latent space. Specifically, an SOEA is adopted to generate $\mathbf{X}_s^{'new}$, which paves the way for further knowledge transfer.

D. GP-based Co-training Method

With the help of domain adaptation, we are able to identify a set of promising samples $\boldsymbol{X}_s^a$ (in decision space), which are considered to be useful for the target domain (f_s). Note, however, that these are unlabeled and cannot be directly used. To makes use of this data, we then adopt an SSL algorithm to predict the slow objective values for $\boldsymbol{X}_s^a$ and select a reliable subset from them to augment the training dataset of the slow objective function. As we mentioned in Section II, the co-training paradigm is a popular method for SSL, which has been widely used in practice [48], [49]. Here, a GP-based co-training method is proposed in Tr-SAEA to leverage these unlabeled data. Note that we put $\boldsymbol{X}_s^{'new}$ and $\boldsymbol{X}_f^{new}$ together as the promising candidate solutions $\boldsymbol{X}_s^a$ for f_s, therefore the proposed co-training strategies can be applied to select half of them with a high confidence level, achieving a more balanced search on the two objectives. Apart from being used as the surrogate model, the GP also serves as the base models for the co-training in our case, due to the following considerations.

First, in the co-training framework the difference between the base models is required to teach each other; otherwise, the predicted labels provided by the two models will be the same, degenerating to self-training [50]–[52]. In GP-based co-training method, we utilize different kernel functions in the two GPs (GP_{s1} and GP_{s2}) to achieve diversity, instead of requiring sufficient and redundant views or different learning algorithms. Specifically, since the covariance function in a GP represents the structure of the response functions, thus the use of different kernel functions for the two GPs will naturally introduce diversity between the two GPs. As the selection of kernel functions remains an open question, we adopt two commonly used ones, the squared exponential function and Matérn kernel. This way, GP_{s1} and GP_{s2} are able to predict objective values of the unlabeled samples $\boldsymbol{X}_s^a$ with a certain degree of diversity.

Second, it is well recognized that how to select useful unlabeled data with the predicted labels to boost the accuracy of both regressors plays a vital role in most co-training algorithms [53]. Generally, the most confidently predicted samples from the unlabeled data set are selected to augment the training sets of the two base models. For example, a popular mechanism is presented in COREG, followed by many SSR methods [54]. COREG estimates the prediction confidence by measuring the error of the regressors (i.e., the mean squared error) on the predicted training set based on the information in the unlabeled data. In the proposed GP-based co-training method, the confidence level of the predictions can be obtained by GPs in a more straightforward way. For the selected unlabeled samples $\boldsymbol{X}_s^a$, GP_{s1} and GP_{s2} provide their predictions combined with a confidence level: $(\boldsymbol{Y}_{s1}^a, \boldsymbol{\theta}_{s1}^a)$ and $(\boldsymbol{Y}_{s2}^a, \boldsymbol{\theta}_{s2}^a)$, respectively. We then rank the predicted data according to the confidence level in an ascending order. Then, the most reliable predicted samples are selected as the augmented data sets D_{t1} and D_{t2} to enlarge the training sets D_{s1} and D_{s2}, respectively:

$$D_{t1} = \left\{(y_{s2,1}^a, \theta_{s2,1}^a), \ldots, (y_{s2,i}^a, \theta_{s2,i}^a)\right\} \tag{13}$$

$$D_{t2} = \left\{(y_{s1,1}^a, \theta_{s1,1}^a), \ldots, (y_{s1,i}^a, \theta_{s1,i}^a)\right\} \tag{14}$$

where i denotes the number of $\boldsymbol{X}_f^{new}$ and half of $\boldsymbol{X}_s^a$ will be added to the training data set of each GP base learner.

E. Complexity Analysis

Here we analyze the computational complexity of the proposed Tr-SAEA algorithm. Tr-SAEA is composed of three main components, namely the construction of the GPs, TCA and CORAL. In this work, three GPs, i.e. GP_f, GP_{s1}, GP_{s2} are constructed with different numbers of training samples. We limit the number of the training data for GP_f as suggested in [10], [55] to reduce the computational cost, and set the upper bound n_{max} to $11d + 24$, where d is the dimension of the decision space. As discussed in Section III, the training data for GP_{s1} and GP_{s2} include the data set evaluated on the original f_s and the augmented data set. Correspondingly, the number of the training data for GP_{s1} and GP_{s2} is the sum of the two subsets, denoted as $FE_s + n_t$ where FE_s is the maximum number of allowed evaluations for f_s and n_t is the number of augmented data. Hence, the complexity of both GP_{s1} and GP_{s2} is $\mathcal{O}((FE_s + n_t)^3)$, while the complexity of GP_f is $\mathcal{O}(n_{max}^3)$. In addition, the computational complexity for the construction of the GPs in Tr-SAEA is $\mathcal{O}(n_{max}^3 + 2(FE_s + n_t)^3)$. As analyzed in [32], the computational complexity of TCA is $\mathcal{O}(m(n_1 + n_2)^2)$,

where m denotes the dimension of the latent space (we set $m = 20$), n_1 and n_2 denote the number of the source and target samples, respectively. In Tr-SAEA, we use the same number of source and target samples. Therefore, $n_1 = n_2$ and they are the number of data evaluated by f_s, i.e. FE_s. Correspondingly, the computational complexity of TCA is $\mathcal{O}(m(2FE_s)^2)$. Moreover, the time complexity of CORAL relies on the calculation of the covariance, which is $\mathcal{O}((n_1 + n_2)d^2)$, where d is the dimension of the samples [31]. In CORAL, $d = 1$. Therefore, CORAL has a computational complexity of $\mathcal{O}(2FE_s)$. In summary, the overall computational complexity of Tr-SAEA is $\mathcal{O}(n_{max}^3+2(FE_s+n_t)^3+4m(FE_s)^2+2FE_s)$. As FE_s+n_t and $4m$ is usually smaller than n_{max}, the computational complexity of Tr-SAEA is of the same order as other GP-based SAEAs [11], [56].

IV. Experimental Settings

A. Test Problems

There are no test problems available that have inherently heterogeneous per-objective evaluation times. However, it is easy to create such test problems by taking any multi-/bi-objective optimization problem, and then assuming and simulating that one of the objectives takes longer to evaluate than the other. Consequently, we have selected three widely used suites of bi-objective test problems for our experimental study, consisting of 22 benchmark problems in total, and extend them to simulate NET-BOPs. Nine of them are taken from the DTLZ test suite [57], including DTLZ1 to DTLZ7 and two modified counterparts (DTLZ1a and DTLZ3a) of DTLZ1 and DTLZ3. UF1-UF7 are taken from the UF test suite [58], five of them are taken from the ZDT test suite [59], and the remaining one is a continuous version of the mapped OneMax problem (cm-OneMax) [19]; cm-OneMax allows us to control the level of correlation between the objectives. Let f_2 (objective 2) in the benchmark problems be the slow objective f_s, as this work focuses on NET-BOPs. Specifically, DTLZ1a and DTLZ3a are designed to reduce the complexity to a reasonable level by changing the multi-model g function used in DTLZ1 and DTLZ3. The g function, given in the following, is suggested to control the ruggedness of DTLZ1 and DTLZ3 [57],

$$g = 100[5+\sum_{i\in 1,...,d}(x_i-0.5)^2-\cos(20\pi(x_i-0.5))], i = 1, ..., d, \tag{15}$$

where d denotes the number of decision variables. As indicated in [60], 20π within the cosine term triggers excessively ruggedness. Consequently, 2π is adopted to reduce the complexity to a reasonable level. As recommended in [57], the number of decision variables for the DTLZ test instances is set to $d = M + K - 1$, where $K = 5$ is adopted for DTLZ1 and DTLZ1a, $K = 10$ is used for DTLZ2 to DTLZ6 as well as DTLA3a, and $K = 20$ is employed in DTLZ7. M represents the number of objectives, here $M = 2$. The number of decision variables for the UF test suite is set as 30 and for the ZDT test suite is set as 8.

The cm-OneMax problem is used to demonstrate the sensitivity of the proposed algorithm on the correlation among objectives. Let $n(\mathbf{x})$ and $n(\mathbf{x}^{map})$ be the sum of all variable values in a decision vector $\mathbf{x}$ and $\mathbf{x}^{map}$, respectively. Here, $\mathbf{x}^{map}$ is a mapped version of $\mathbf{x}$. The cm-OneMax problem is defined as [19]:

$$f = (f_1, f_2) = (n(\mathbf{x}), n(\mathbf{x}^{map})) \tag{16}$$

where $x_i^{map} = |x_i - map_i|, i = 1, ..., n$, n is the number of decision variables and is set to 10; $map_i \in [0, 1]$ is set independently for each decision variable by flipping a coin biased by the degree of correction $corr \in [-1, 1]$ desired. For example, $corr = 0$ means no correlation between the objectives, while $corr = 1$ and $corr = -1$ mean maximal positive and maximal negative correlation, respectively. Given the degree of correction, map_i is set to zero with a probability of $(1 + corr)/2$.

B. Performance Indicators

The inverted generational distance (IGD) [61] and hypervolume (HV) [62] are adopted to assess the performance of the algorithms in terms of convergence and diversity of the obtained non-dominated solutions. The IGD value of the obtained solution set is calculated with a reference point set that uniformly sampled from the objective space along the true PF, and a point that is dominated by all the points in the reference point set is used for the HV calculation. The PlatEMO toolbox [63] is used to calculate values of IGD and HV in our experiments. We run each algorithm under comparison on each benchmark problem for 30 independent times, recording the corresponding results with regard to IGD and HV values. The Wilcoxon rank sum test is adopted to compare the mean results achieved by each of the algorithms at a significance level of 0.05. In the tables, the symbols "(+)" and "(–)" indicate that the proposed algorithm performs significantly better and significantly worse than the compared algorithm, respectively, while "(≈)" means there is no significant difference between them. Also, the values highlighted in bold in the tables indicate the algorithm with the best mean performance (excluding undelayed search) on a test problem instances.

C. Algorithms Under Comparison

To examine the performance of the proposed algorithm on MOPs with delayed objectives, Tr-SAEA is compared with several state-of-the-art delay-handling methods, including *Waiting*, *Fast-first*, *Brood interleaving*, *Speculative interleaving* [16], HK-RVEA proposed [19] and T-SAEA [20]. It is important to highlight that *Waiting*, *Fast-first* and the two *Interleaving* schemes were designed for non-surrogate based methods. Consequently, one may argue that the comparison with HK-RVEA and T-SAEA is not fair. However, since these are the only published algorithms for coping with NET-BOPs, we will include them for completeness reasons and also to investigate if surrogate-based methods can improve performance. Besides, the performance of K-RVEA [11] applied on the same problem instance but in the undelayed environment is also reported for reference; hence, we would expect this algorithm to perform best in most cases. In order to further

test the effectiveness of the proposed strategies, Tr-SAEA is compared with its following four variants.

- NT-SAEA: To validate the benefits of knowledge transfer in the proposed scheme, we compare a variant, called NT-SAEA, that selects different numbers of new samples to update GPs for handling the NET-BOPs, and further train each GP model with different number of samples due to different evaluation times. Take $\tau = 5$ as an example, while three new samples are selected to update GP_s, 15 new samples will be selected for GP_f.
- ND-SAEA: To investigate the effectiveness of the HDA method, we introduce a variant without using the HDA, called ND-SAEA. ND-SAEA directly uses the samples evaluated on f_f as the unlabeled data for f_s, and then the GP-based co-training is applied as done in Tr-SAEA.
- NCoral-SAEA: NCoral-SAEA is a variant of Tr-SAEA, where only TCA is employed for the domain adaptation task in the NET-BOPs.
- NTCA-SAEA: NTCA-SAEA is introduced for testing the benefits of TCA. In NTCA-SAEA, only CORAL is adopted to transfer the knowledge between f_s and f_s.

To compare the algorithms as fairly as possible, where needed we adopt RVEA as the multi-objective optimizer and a standard genetic algorithm (GA) using a real number coding scheme, the simulated binary crossover (SBX) and the polynomial mutation as the single-objective optimizer in all the compared algorithms. HK-RVEA, T-SAEA, and Tr-SAEA uses their respective customized approaches to update their surrogates.

Tr-SAEA and its variants are implemented in MATLAB R2009a on an Intel Core i7 with 2.21 GHz CPU, and the code of the compared algorithms is provided by their authors. The GP model is constructed using the DACE toolbox [64]. If not otherwise stated, these are the algorithm parameters settings used in the experiments, given as follows:

- The initial population size (also the number of initial training points) for all the compared algorithms is set to 100 and generated by the LHS. As recommended in [1], the probabilities for SBX and the polynomial mutation are set to 0.9 and $1/d$, respectively; the distribution indexes are set to 20.
- The maximum number of generations before updating GPs w_{max} is set to 20.
- τ is set as 5 and 10, respectively, to investigate the impact of the ratio between the two evaluation times.
- The maximum number of slow objective function evaluations $FE_s^{max} = 200$.

V. Experimental Results

A. Sensitivity Analysis

We first investigate the impact of the parameters on the performance of Tr-SAEA in NET-BOPs. Here, we consider two important parameters introduced by the proposed algorithm. One is u, the number of new samples selected in each generation to update GPs, and the other is D, the dimension of the latent space introduced by TCA. To study the sensitivity of the parameters in Tr-SAEA, some representative test instances

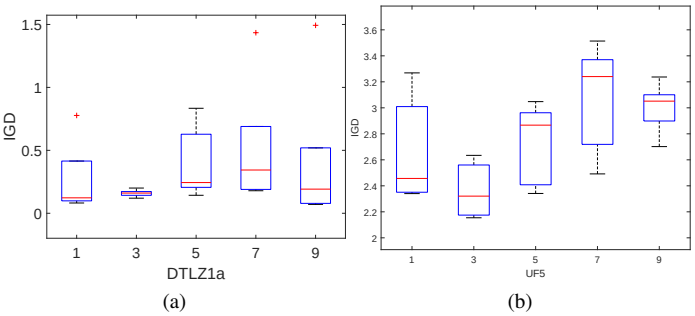

Fig. 4. Boxplots of the IGD values for parameter u in Tr-SAEA on (a) DTLZ1a and (b) UF5.

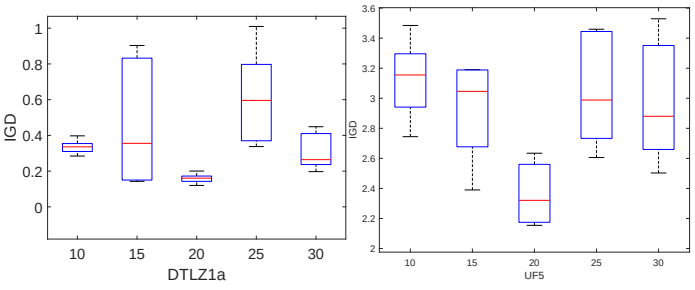

Fig. 5. Boxplots of the IGD values for parameter D in Tr-SAEA on (a) DTLZ1a and (b) UF5.

including DTLZ1a, DTLZ4, UF2, UF5, ZDT1 and ZDT6, are selected to test the performance of Tr-SAEA with different settings. Among them, DTLZ1a has a linear PF, while UF2 has a nonlinear PF and UF5 has a nonlinear and disconnected PF; the PF of DTLZ4 and ZDT1 is concave, while the PF of ZDT6 and UF2 is non-concave. Note that ZDT6 is also characterised by the non-uniformity of the search space. The delay ratio τ is set to 5 for all test instances in the sensitivity analysis.

First, as u is set to 5 in GP-based SAEAs for solving expensive MOPs [11], [12] and set to 3 for solving the NET-BOPs [19], we test the influence of u by setting it to 1, 3, 5, 7, and 9, respectively, with D being fixed to 20. Fig. 4 shows the boxplots of the IGD values over different settings of u on DTLZ1a and UF5. Due to space limit, the boxplot of the remaining results are given in Figs. S1-S2 in the Supplementary materials. It can be observed that Tr-SAEA with $u = 3$ can achieve the best overall performance on the selected test problems, and Tr-SAEA with $u = 1$ also shows competitive performance. Accordingly, we conclude that u should not be too small or too large for handling MOPs with a limited evaluation budget, as SAEAs with a small number of new samples can evolve more generations and further achieve better converged solutions. Therefore, u is set to 3 in the following experiments.

Next, we investigate the influence of D on the performance of Tr-SAEA. The parameter D is introduced by TCA when aligning the marginal distributions of two domains in a latent space. As recommended in [47], D is set to 20. It is worth noting that, the motivation behind adopting domain adaptation techniques in our case is that the two objectives, f_f and f_s,

TABLE I
STATISTICAL RESULTS OF THE IGD VALUES OBTAINED BY WAITING, FAST-FIRST, BROOD INTERLEAVING, SPECULATIVE INTERLEAVING, HK-RVEA, T-SAEA, TR-SAEA AND UNDELAYED ALGORITHN WITH $FE_s^{max} = 200$ AND $\tau = 5$

Test problem	Waiting		Fast-first		Brood		Speculative		HK-RVEA		T-SAEA		Tr-SAEA		Undelayed	
	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std
DTLZ1	30.2 +	14.3	69.7 +	24.1	28.6 +	10.2	48.1 +	10.5	42.2 +	10.5	21.7 +	11.9	**20.7**	5.38	15.8	5.71
DTLZ1a	14.2 +	8.48	2.62 +	0.25	15.6 +	6.54	28.6 +	8.22	0.52 +	0.18	1.06 +	1.00	**0.20**	0.07	0.32	0.08
DTLZ2	0.24 +	0.05	0.80 +	0.08	0.36 +	0.05	0.38 +	0.03	0.10 +	0.02	0.05 +	0.03	**0.03**	0.01	0.02	0.01
DTLZ3	349 ≈	83.4	549 +	142	357 +	74.6	462 +	67.5	354 ≈	41.9	**203** –	100	327	82.1	168	61.9
DTLZ3a	227 +	75.4	546 +	86.9	313 +	83.1	406 +	96.6	14.9 +	5.27	5.34 +	37.5	**3.39**	1.87	13.9	2.32
DTLZ4	0.51 +	0.32	0.78 +	0.11	0.54 +	0.06	0.65 +	0.10	0.23 ≈	0.11	0.60 +	0.13	**0.16**	0.07	0.05	0.07
DTLZ5	0.27 +	0.06	0.86 +	0.10	0.35 +	0.04	0.39 +	0.03	0.09 +	0.02	0.05 +	0.02	**0.03**	0.03	0.02	0.00
DTLZ6	7.31 +	0.52	8.79 +	0.11	7.63 +	0.44	8.26 +	0.13	4.10 +	0.54	2.56 +	1.21	**0.72**	0.09	2.52	0.04
DTLZ7	4.41 +	0.62	7.53 +	0.39	5.53 +	0.47	5.57 +	0.68	0.06 ≈	0.05	1.15 +	0.91	**0.03**	0.01	0.04	0.00
UF1	1.01 +	0.14	0.49 +	0.04	0.36 +	0.02	0.42 +	0.05	0.23 +	0.02	**0.19** ≈	0.02	**0.19**	0.01	0.15	0.04
UF2	0.50 +	0.07	0.58 +	0.09	0.45 +	0.03	0.51 +	0.03	0.15 +	0.02	0.14 ≈	0.02	**0.12**	0.01	0.09	0.02
UF3	0.97 +	0.08	1.22 +	0.06	0.96 +	0.04	1.08 +	0.07	0.54 ≈	0.05	**0.19** –	0.08	0.49	0.01	0.44	0.02
UF4	**0.21** ≈	0.01	0.24 ≈	0.02	0.23 ≈	0.00	0.23 ≈	0.01	0.22 ≈	0.00	0.23 ≈	0.02	0.22	0.00	0.12	0.01
UF5	4.75 +	0.42	3.53 +	0.26	2.84 +	0.18	3.25 +	0.16	2.46 ≈	0.43	2.49 ≈	0.44	**2.43**	0.28	1.55	0.31
UF6	4.36 +	0.63	2.28 +	0.24	1.69 +	0.13	1.99 +	0.17	1.34 ≈	0.13	**1.01** –	0.25	1.32	0.39	0.58	0.13
UF7	1.20 +	0.12	0.58 +	0.06	0.38 +	0.03	0.47 +	0.05	**0.21** –	0.04	0.37 ≈	0.06	0.32	0.11	0.32	0.13
ZDT1	6.37 +	1.63	17.5 +	3.22	11.9 +	1.43	12.8 +	2.49	0.21 +	0.07	1.05 +	0.59	**0.02**	0.01	0.02	0.00
ZDT2	8.30 +	1.89	20.6 +	2.20	11.6 +	2.17	12.9 +	4.21	**0.54** –	0.14	1.05 +	0.56	0.56	0.12	0.02	0.00
ZDT3	6.30 +	1.75	17.3 +	3.01	11.2 +	1.86	13.0 +	2.45	**0.15** ≈	0.02	0.70 +	0.37	**0.15**	0.21	0.04	0.02
ZDT4	34.6 +	6.47	36.0 +	8.96	18.8 +	5.62	21.8 +	6.69	**2.25** –	1.92	33.0 +	7.70	3.29	2.74	17.2	9.8
ZDT6	9.38 +	0.64	11.3 +	0.29	9.91 +	0.49	10.4 +	0.37	1.11 +	0.49	2.44 +	0.88	**0.61**	0.06	0.49	0.10
+/≈/-	19/2/0		20/1/0		20/1/0		20/1/0		10/8/2		13/5/3					

tend to be related to each other near the PF; thus, the source and target sample spaces are the one-dimensional objective values of f_f and f_s, respectively. As our aim is to explore the corresponding decision variables whose objective values are close to the transformed values in the obtained latent space, it is necessary to discuss the setting of D. In the sensitivity test of D in Tr-SAEA, D is set to 10, 15, 20, 25 and 30, respectively. The results achieved by Tr-SAEA with different settings of D on DTLZ1a and UF5 are plotted in Fig. 5. It is evident from the results that Tr-SAEA with $D = 20$ is able to achieve lower IGD values compared with other cases, indicating that $D = 20$ is preferred in Tr-SAEA. Similar conclusions can be drawn from the boxplots of the IGD values on DTLZ4, UF2, ZDT1 and ZDT6, which are shown in Figs. S3-S4 in the Supplementary materials.

B. Comparison with State-of-the-art methods

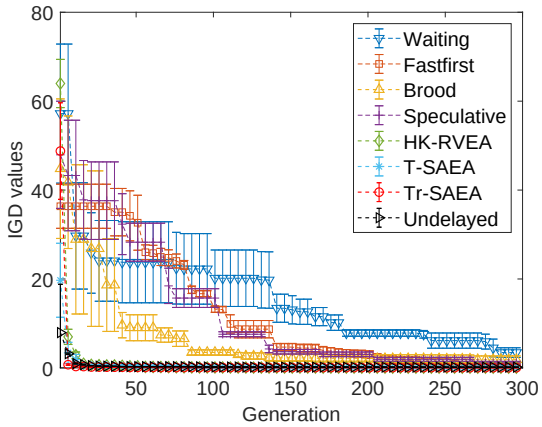

Fig. 6. IGD values obtained on DTLZ1a by each algorithm considered as a function of the generation counter.

Comparative results in terms of the IGD values of each algorithm on the test problems with τ setting as 5 and 10 are presented in Table I and Table S1 in the Supplementary materials, respectively, where the best results in the delayed environment are highlighted. Tables S2-S3 present the HV values obtained by each algorithms on the test instances with $\tau = 5$ and 10, respectively. Note that the number of FEs for K-RVEA in the absence of delays is set to $FE_s^{max} * \tau$. An attainment surface is the surface formed by the obtained non-dominated solutions and allows visual inspection of performance in terms of the convergence and diversity [65]. To further evaluate the PF approximation obtained by each algorithm, we select some test instances with different characters, i.e., DTLZ1a, DTLZ4, DTLZ7, UF2, UF5 and ZDT1, and show attainment surfaces that can be obtained by half of the runs, known as median attainment surfaces [65]. Fig. S5 in the Supplementary material depicts the median attainment surfaces obtained by each algorithm across 30 runs on the selected test instances.

From these results, we can see that Tr-SAEA has achieved the best overall performance on the three test suites, followed by HK-RVEA. Overall, the non-surrogate based methods (*Waiting*, *Fast-first*, *Brood* and *Speculative interleaving*) cannot compete with the surrogate-based methods (HK-RVEA, Tr-SAEA, T-SAEA), confirming the observation already made in [19]. More specifically, Tr-SAEA produces the best results in terms of IGD values on 13 of totally 21 test instances when $\tau = 5$ and on 14 test instances when $\tau = 10$. In addition, the proposed algorithm significantly outperforms the compared delay-handling algorithms on most test instances, confirming the effectiveness of Tr-SAEA for solving NET-BOPs. In fact, on some test instances, Tr-SAEA (and in some cases also the other two surrogate-based methods) outperform the undelayed performance, which is something that was also observed in [19]. Taking a closer look at the tables, the following key observations can be made.

- Results on the DTLZ Problems: Tr-SAEA yields the best results on all DTLZ instances except for DTLZ3 with

$\tau = 5$ and DTLZ4 with $\tau = 10$. Despite that T-SAEA shows its superiority on DTLZ3 instance, its performance degenerates as τ increases to ten, and in this case Tr-SAEA is the most effective delay-handling method for all DTLZ instances. It is interesting to notice that *Waiting* performs more stably than others, as confirmed in Table S1 with $\tau = 10$. This is because *Waiting* always waits for the slow evaluations before continuing its search process, thereby avoiding the search bias. As presented in Fig. S5, we also note that by learning knowledge from the optimization of f_f and further augmenting the training data for f_s, Tr-SAEA can mitigate the search bias towards f_f. The median attainment surfaces obtained by each algorithm also demonstrate that Tr-SAEA achieves a set of well converged and evenly distributed solutions. Besides, Tr-SAEA has shown good performance on DTLZ1, DTLZ1a, DTLZ3 and DTLZ3a that are generally employed to test an MOEA's convergence, indicating that the transfer learning scheme can help Tr-SAEA effectively converge towards the right direction.

- Results on the UF Problems: Both Tr-SAEA and T-SAEA can achieve promising results on all instances with $\tau = 5$, which can be confirmed by Fig. S5. The reason may be that the initial population generated by the algorithms for UF test instances is generally closed to the true PF. However, an increase of τ tends to worsen significantly the performance of T-SAEA. By contrast, the proposed Tr-SAEA keeps performing competitively, which can be explained by the difference between the transfer learning scheme used in T-SAEA and Tr-SAEA. T-SAEA captures the common pivot features of each objective using feature selection methods, and the corresponding parameters of the surrogates of the objectives are shared to benefit the optimization of f_s. Hence, the performance of T-SAEA relies on the effectiveness of the feature selection. Instead of using the parameter-based transfer learning, Tr-SAEA aims to transfer some useful samples from the search experience of f_f to enhance the search efficiency of f_s. These transferable samples are learned by aligning the second-order statistics and marginal distributions of the two domains, and carefully selected according to the co-training strategy, thereby reducing the negative transfer as much as possible.
- Results on the ZDT Problems: Both Tr-SAEA and HK-RVEA perform quite well on the ZDT problems, and Tr-SAEA significantly outperforms the remaining algorithms on all the ZDT instances. For instances with $\tau = 5$, Tr-SAEA shows the overwhelmingly superior performance on ZDT1 and ZDT6, while HK-RVEA is the most competitive method on ZDT2 and ZDT4. It is interesting to see that, for instances with $\tau = 10$, Tr-SAEA achieves the best performance on four out of six test instances, while HK-RVEA is negatively affected with the increase of the delay length. This confirms that the proposed HDA can effectively capture the similarity between objectives, so that Tr-SAEA can maintain its superiority with the growing τ. Note that HK-RVEA has achieved the best performance on ZDT4 (containing many local PFs) with both $\tau = 5$ and $\tau = 10$. The reason may be that HK-EVEA adopts K-RVEA as its framework, where an adequate degree of diversity is maintained in selecting new samples.
- By comparing results listed in Table I and Table S1, we can observe that although more evaluation times for f_f are available with increasing τ (FE_s is fixed as 200), the corresponding impact is not always positive. For example, as we mentioned before, the results obtained by T-SAEA on DTLZ3 become worse when $\tau = 10$. This can be explained by two main reasons. First, except for *Waiting*, the remaining algorithms consume different per-objective budgets, and much more intensive search is conducted on the cheap objective function, resulting in a strong search bias. Although *Waiting* can avoid such biases, it still cannot achieve good performance due to its slow convergence speed (as *Waiting* goes at the rate of the slow objective). Hence, how to avoid such biases as much as possible while speeding up convergence is a key concern in solving NET-BOPs (as already highlighted in [16]). Thus, it makes good sense to transfer knowledge between objectives for alleviating search biases. Second, it is noticed that algorithms making use of the correlation between the slow and fast objectives are outperformed by *Waiting* on some instances, indicating that negative knowledge transfer cannot be completely avoided. Nonetheless, Tr-SAEA is least affected by the ratio τ compared with other algorithms.
- According to the results, we can see that the obtained solutions by all algorithms on some test instances, such as DTLZ1a, are not yet well converged due to the small number of FEs resulting from the limited computational budget. This is why SAEAs generally are required to achieve fast convergence. To further investigate the performance in terms of the convergence and diversity, we run each algorithm on DTLZ1a with $FE_s^{max} = 1000$. The IGD and HV values obtained by each algorithm as a function of the generation counter are plotted in Fig. 6 and Fig. S6, respectively, indicating the fast convergence of Tr-SAEA.

C. Impact of the Correlation Between Objectives

The test problems considered so far do not allow us to control the correlation between objectives, which is a problem feature that has a considerable impact on algorithm performance [16], [19]. In this section we use a tuneable toy problem, cm-OneMax with different correlation levels (i.e., $corr \in \{-1, -0.75, -0.5, -0.25, 0, 0.25, 0.5, 0.75, 1\}$), to investigate the impact of the correlation between objectives onto performance. Tr-SAEA performs the domain adaptation technique and the co-training to discover the transferable knowledge between f_f and f_s, so that the optimization of f_s can be enhanced. As a result, the accomplishment of Tr-SAEA is based on the assumption that there is a functional relationship between f_f and f_s when near and on the PF. The IGD values obtained by each algorithm are presented in Tables II.

TABLE II
THE IGD VALUES OBTAINED BY WAITING, FAST-FIRST, BROOD INTERLEAVING, SPECULATIVE INTERLEAVING, HK-RVEA, T-SAEA AND TR-SAEA ON THE CM-ONEMAX PROBLEM WITH $FE_s^{max} = 200$

cm-OneMax	τ	Waiting		Fast-first		Brood		Speculative		HK-RVEA		T-SAEA		Tr-SAEA		Undelayed	
		mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std
corr=-1	τ=5	0.41 +	0.12	2.26 +	0.07	0.52 +	0.09	0.45 +	0.10	0.20 +	0.09	0.18 +	0.06	**0.09**	0.04	0.24	0.05
	τ=10	0.41 +	0.12	2.65 +	0.15	0.51 +	0.11	0.42	0.07	0.22 +	0.07	0.22 +	0.08	**0.08**	0.04	0.22	0.05
corr=-0.75	τ=5	0.90 +	0.08	1.82 +	0.22	1.13 +	0.08	1.09 +	0.17	0.14 +	0.03	0.46 +	0.10	**0.07**	0.02	0.13	0.04
	τ=10	0.90 +	0.08	2.08 +	0.25	1.14 +	0.07	1.04 +	0.09	0.16 +	0.03	0.55 +	0.15	**0.07**	0.02	0.11	0.02
corr=-0.5	τ=5	0.97 +	0.08	1.82 +	0.22	1.11 +	0.12	1.02 +	0.14	0.13 +	0.02	0.43 +	0.12	**0.07**	0.01	0.14	0.03
	τ=10	0.97 +	0.08	1.98 +	0.28	1.06 +	0.07	1.05 +	0.14	0.15 +	0.04	0.47 +	0.10	**0.07**	0.01	0.10	0.02
corr=-0.25	τ=5	1.20 +	0.11	1.83 +	0.21	1.32 +	0.16	1.35 +	0.11	0.13 +	0.03	0.42 +	0.14	**0.06**	0.01	0.11	0.03
	τ=10	1.20 +	0.11	2.27 +	0.21	1.42 +	0.10	1.30 +	0.13	0.14 +	0.03	0.58 +	0.20	**0.07**	0.01	0.10	0.02
corr=0	τ=5	1.50 +	0.21	2.02 +	0.30	1.70 +	0.18	1.57 +	0.16	0.11 +	0.02	0.38 +	0.09	**0.07**	0.02	0.11	0.02
	τ=10	1.50 +	0.21	2.35 +	0.25	1.60 +	0.10	1.50 +	0.15	0.11 +	0.02	0.51 +	0.19	**0.06**	0.01	0.09	0.01
corr=0.25	τ=5	1.52 +	0.15	2.06 +	0.22	1.70 +	0.12	1.47 +	0.13	0.12 +	0.01	0.60 +	0.12	**0.06**	0.01	0.09	0.01
	τ=10	1.52 +	0.15	2.28 +	0.24	1.71 +	0.22	1.53 +	0.13	0.12 +	0.01	0.62 +	0.21	**0.06**	0.01	0.08	0.01
corr=0.5	τ=5	1.78 +	0.18	2.32 +	0.31	2.01 +	0.24	1.75 +	0.28	0.14 +	0.02	0.45 +	0.20	**0.05**	0.01	0.08	0.01
	τ=10	1.78 +	0.18	2.27 +	0.34	1.92 +	0.23	1.78 +	0.22	0.13 +	0.01	0.54 +	0.11	**0.05**	0.01	0.06	0.01
corr=0.75	τ=5	1.96 +	0.30	2.61 +	0.35	2.33 +	0.27	2.28 +	0.26	0.13 +	0.01	0.40 +	0.17	**0.04**	0.00	0.07	0.01
	τ=10	1.96 +	0.30	2.60 +	0.34	2.31 +	0.25	1.87 +	0.18	0.13 +	0.02	0.39 +	0.11	**0.04**	0.01	0.06	0.00
corr=1	τ=5	2.77 +	0.56	3.86 +	0.33	3.66 +	0.45	3.16 +	0.34	**0.00** ≈	0.00	**0.00** ≈	0.00	**0.00**	0.00	0.00	0.00
	τ=10	2.77 +	0.56	3.81 +	0.55	3.24 +	0.67	3.27 +	0.33	**0.00** ≈	0.00	**0.00** ≈	0.00	**0.00**	0.00	0.00	0.00

According to Table II, it is clear that Tr-SAEA significantly outperforms the compared algorithms on all the cm-OneMax problem with different degree of correlations between objectives. Particularly, the performance of Tr-SAEA is better than that of K-RVEA in the absence of delays, implying the effectiveness of the transfer learning strategy in Tr-SAEA. Tr-SAEA is capable of capturing both the negative and positive correlations between objectives, indicating that the proposed hybrid DA strategy can effectively learn common transfer components due to the alignment in terms of both the second-order statistics and marginal distributions of the source and target domains. It is noteworthy that for a loose correlation, i.e. $corr \in \{-0.25, +0.25\}$, Tr-SAEA also obtained the best performance, followed by HK-RVEA and T-SAEA, which can be explained by Tr-SAEA's ability of acquiring the mutual feature representation and identifying the useful samples with high confidence. The results also show that the surrogate based delay-handling methods achieve better performance than the non-surrogate based ones on the cm-OneMax, indicating the effectiveness of the surrogates.

TABLE III
STATISTICAL RESULTS OF THE IGD VALUES OBTAINED BY NT-SAEA, ND-SAEA, NCORAL-SAEA, NTCA-SAEA AND TR-SAEA WITH $FE_s^{max} = 200$ AND $\tau = 5$

Test problem	NT-SAEA		ND-SAEA		NCoral-SAEA		NTCA-SAEA		Tr-SAEA	
	mean	std	mean	std	mean	std	mean	std	mean	std
DTLZ1	59.2 +	15.1	45.8 +	15.7	44.6 +	9.34	38.3 +	11.4	**20.7**	5.39
DTLZ1a	13.5 +	6.10	0.67 +	0.85	0.30 ≈	0.31	0.54 +	0.64	**0.20**	0.07
DTLZ2	0.28 +	0.04	0.06 +	0.02	0.04 ≈	0.01	0.04 ≈	0.02	**0.03**	0.01
DTLZ3	416 ≈	95.6	367 ≈	71.1	395 ≈	72.0	374 ≈	99.1	**327**	82.1
DTLZ3a	211 +	84.7	25.0 +	23.8	14.3 +	11.8	47.6 +	54.9	**3.39**	1.88
DTLZ4	0.30 +	0.13	0.25 ≈	0.12	0.37 +	0.23	0.26 ≈	0.24	**0.16**	0.08
DTLZ5	0.28 +	0.06	0.06 +	0.01	0.05 +	0.01	**0.03** ≈	0.01	**0.03**	0.04
DTLZ6	5.99 +	1.40	1.32 ≈	0.78	1.13 ≈	0.64	1.14 ≈	0.74	**0.72**	0.10
DTLZ7	1.11 +	0.63	0.52 +	0.18	0.38 +	0.32	0.05 ≈	0.01	**0.03**	0.01
UF1	0.45 +	0.03	0.24 +	0.04	0.33 +	0.14	0.41 +	0.11	**0.19**	0.01
UF2	0.35 +	0.06	0.15 +	0.02	0.15 +	0.02	0.14 ≈	0.01	**0.12**	0.01
UF3	0.78 +	0.11	0.55 +	0.05	0.50 ≈	0.03	0.51 ≈	0.03	**0.49**	0.02
UF4	**0.20** ≈	0.07	0.23 ≈	0.01	0.23 ≈	0.01	0.22 ≈	0.01	0.22	0.00
UF5	3.05 +	0.21	2.77 ≈	0.43	2.67 ≈	0.40	2.78 ≈	0.31	**2.43**	0.29
UF6	1.70 +	0.27	1.56 ≈	0.40	1.59 ≈	0.46	1.37 ≈	0.42	**1.32**	0.39
UF7	0.40 ≈	0.03	**0.29** ≈	0.06	0.38 ≈	0.08	0.40 ≈	0.08	0.32	0.12
ZDT1	1.87 +	1.02	0.28 +	0.19	0.28 +	0.14	0.03 ≈	0.03	**0.02**	0.01
ZDT2	0.83 +	0.99	**0.52** ≈	0.24	0.58 ≈	0.13	0.60 +	0.01	0.56	0.13
ZDT3	0.63 +	0.16	0.29 ≈	0.13	0.30 ≈	0.16	**0.13** –	0.06	0.16	0.21
ZDT4	**1.62** ≈	3.06	8.40 +	4.65	10.2 +	3.29	5.03 +	3.52	3.29	2.74
ZDT6	7.93 +	0.52	3.84 +	2.38	3.68 +	2.53	0.63 ≈	0.08	**0.61**	0.06
+/≈/-	17/4/0		12/9/0		10/11/0		14/6/1			

D. Ablation Studies

According to the experimental results above, it is clear that the proposed algorithm is competitive among the compared delay-handling methods for solving various NET-BOPs. To provide more insights into the role of the HDA and GP-based co-training in Tr-SAEA play, we compare the performance of Tr-SAEA and the four variants of Tr-SAEA on DTLZ, UF and ZDT test suites.

The experimental results in terms of the IGD values on test instances with $\tau = 5$ are presented in Table III. It is evident from Table III that Tr-SAEA shows the best performance over the compared variants on 16 out of 21 test problems in total. According to the comparative results between Tr-SAEA and its four variants, the following observations can be made:

- Tr-SAEA achieves overwhelmingly superior performance on 16 out of 21 test instances when compared with NT-SAEA, and performs comparably well on the other test instances. NT-SAEA handles the NET-BOPs in the framework of SAEAs by straightforwardly sampling different numbers of data from f_f and f_s, thus running a risk of considerable search biases. This is similar to *Fast-first*, where a search bias towards the fast objective f_f can be induced due to the focus on optimizing the fast objective only for most of the search. By contrast, Tr-SAEA takes advantage of the additional samples obtained by f_f, which is able to reduce search biases. Therefore, we can conclude from these comparative results that the use of TL techniques is helpful for handling the NET-BOPs.
- Compared with ND-SAEA, Tr-SAEA yields the best IGD results on all test instances except for UF7 and ZDT2, on which ND-SAEA shows slightly better performance. From these results, we see that ND-SAEA is less effective in solving NET-BOPs, which can be attributed to the fact that ND-SAEA does not consider the correlation

between f_f and f_s in co-training. As a result, the obtained solutions for f_f may fail to help the optimization of f_s.
- Compared with NCoral-SAEA, Tr-SAEA shows significantly better results on 10 out of 21 test instances, confirming the importance of the combination of TCA and CORAL in domain adaptation. The results also reveal that CORAL plays a vital role in Tr-SAEA.
- Similar to NCoral-SAEA, NTCA-SAEA also shows worst overall performance compared with Tr-SAEA. Note that Tr-SAEA achieves much superior performance on DTLZ1, DTLZ1a, DTLZ3a, UF1, ZDT2 and ZDT4, while NTCA-SAEA yields good results on DTLZ7 and ZDT3, both featuring disconnected Pareto fronts. The comparison between Tr-SAEA and its two variants that use one single DA technique only (NCoral-SAEA and NTCA-SAEA) implies that it is more effective to combine CORAL and TCA for domain adaptation.

VI. Conclusion

In this paper, an effort is made to solve MOPs where different objective functions take considerably different evaluation times. To more efficiently solve bi-objective optimization problems with one fast objective and one slow objective (NET-BOPs), we integrate a transfer learning scheme into an SAEA framework to augment the training data for the slow objective by transferring knowledge from the fast one, thereby alleviating the search biases caused by different computational budgets needed for evaluating the two objectives. We compare the proposed algorithm, Tr-SAEA, with five state-of-the-art (surrogate and non-surrogate-based) delay-handling strategies, as well as four variants of Tr-SAEA, on three widely used test suites. The experimental results on these test instances with different delay lengths demonstrate that the proposed algorithm is capable of efficiently solving NET-BOPs, and the impact of the correlation between objectives on the performance of Tr-SAEA is also investigated (using a tuneable toy problem). The ablation studies further confirm the effectiveness of the proposed strategies.

This work demonstrates that properly utilizing transferable knowledge between two different objective functions is a promising approach to to solve NET-BOPs efficiently. However, the performance of the proposed algorithm degrades as the delay length increases. Therefore, it deserves further efforts to develop more sophisticated transfer learning schemes to alleviate negative transfer so as to further enhance the performance. Another promising avenue for future work is to investigate how to effectively apply domain adaptation approaches to MOPs with more than two objectives, where different objectives have different computational complexities and the Pareto front may be degenerate [66]. In these cases, identifying the most effective and efficient source domains becomes extremely challenging. Finally, it is of interest to consider more computationally scalable surrogate models other than Gaussian processes in order to extend the proposed ideas to surrogate-assisted high-dimensional multi-objective optimization problems.

References

[1] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE transactions on evolutionary computation*, vol. 6, no. 2, pp. 182–197, 2002.

[2] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff, "A reference vector guided evolutionary algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 5, pp. 773–791, 2016.

[3] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.

[4] Y. Jin and B. Sendhoff, "A systems approach to evolutionary multiobjective structural optimization and beyond," *IEEE Computational Intelligence Magazine*, vol. 4, no. 3, pp. 62–76, 2009.

[5] Y. Jin, "Surrogate-assisted evolutionary computation: Recent advances and future challenges," *Swarm and Evolutionary Computation*, vol. 1, no. 2, pp. 61–70, 2011.

[6] Y. Jin, H. Wang, T. Chugh, D. Guo, and K. Miettinen, "Data-driven evolutionary optimization: An overview and case studies," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 3, pp. 442–458, 2019.

[7] S. G. Carlos A. Coello Coello, J. Figueroa Gamboa, M. Guadalupe Castillo Tapia, and R. Hernández Gómez, "Evolutionary multiobjective optimization: open research areas and some challenges lying ahead," *Complex & Intelligent Systems*, vol. 6, no. 2, pp. 221–236, 2020.

[8] A. Díaz-Manríquez, G. Toscano, J. H. Barron-Zambrano, and E. Tello-Leal, "A review of surrogate assisted multiobjective evolutionary algorithms," *Computational Intelligence and Neuroscience*, vol. 2016, 2016.

[9] S. D. Handoko, C. K. Kwoh, and Y.-S. Ong, "Feasibility structure modeling: an effective chaperone for constrained memetic algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 5, pp. 740–758, 2010.

[10] D. Guo, Y. Jin, J. Ding, and T. Chai, "Heterogeneous ensemble-based infill criterion for evolutionary multiobjective optimization of expensive problems," *IEEE Transactions on Cybernetics*, vol. 49, no. 3, pp. 1012–1025, 2018.

[11] T. Chugh, Y. Jin, K. Miettinen, J. Hakanen, and K. Sindhya, "A surrogate-assisted reference vector guided evolutionary algorithm for computationally expensive many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 1, pp. 129–142, 2018.

[12] X. Wang, Y. Jin, S. Schmitt, and M. Olhofer, "An adaptive Bayesian approach to surrogate-assisted evolutionary multi-objective optimization," *Information Sciences*, vol. 519, pp. 317–331, 2020.

[13] D. R. Jones, M. Schonlau, and W. J. Welch, "Efficient global optimization of expensive black-box functions," *Journal of Global optimization*, vol. 13, no. 4, pp. 455–492, 1998.

[14] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. De Freitas, "Taking the human out of the loop: A review of Bayesian optimization," *Proceedings of the IEEE*, vol. 104, no. 1, pp. 148–175, 2015.

[15] R. Allmendinger, M. T. Emmerich, J. Hakanen, Y. Jin, and E. Rigoni, "Surrogate-assisted multicriteria optimization: Complexities, prospective solutions, and business case," *Journal of Multi-Criteria Decision Analysis*, vol. 24, no. 1-2, pp. 5–24, 2017.

[16] R. Allmendinger, J. Handl, and J. Knowles, "Multiobjective optimization: When objectives exhibit non-uniform latencies," *European Journal of Operational Research*, vol. 243, no. 2, pp. 497–513, 2015.

[17] H. Wang, J. Doherty, and Y. Jin, "Hierarchical surrogate-assisted evolutionary multi-scenario airfoil shape optimization," in *2018 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1–8, IEEE, 2018.

[18] R. Allmendinger and J. Knowles, "Hang On a Minute: Investigations on the effects of delayed objective functions in multiobjective optimization," in *International Conference on Evolutionary Multi-Criterion Optimization*, pp. 6–20, Springer, 2013.

[19] T. Chugh, R. Allmendinger, V. Ojalehto, and K. Miettinen, "Surrogate-assisted evolutionary biobjective optimization for objectives with non-uniform latencies," in *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 609–616, ACM, 2018.

[20] X. Wang, Y. Jin, S. Schmitt, and M. Olhofer, "Transfer learning for Gaussian process assisted evolutionary bi-objective optimization for objectives with different evaluation times," in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2020)*, ACM, 2020.

[21] A. I. Forrester, A. Sóbester, and A. J. Keane, "Multi-fidelity optimization via surrogate modelling," *Proceedings of the royal society a: mathematical, physical and engineering sciences*, vol. 463, no. 2088, pp. 3251–3269, 2007.

[22] L. Le Gratiet and J. Garnier, "Recursive co-kriging model for design of computer experiments with multiple levels of fidelity," *International Journal for Uncertainty Quantification*, vol. 4, no. 5, 2014.

[23] K. Swersky, J. Snoek, and R. P. Adams, "Multi-task bayesian optimization," in *Advances in Neural Information Processing Systems*, pp. 2004–2012, 2013.

[24] H. Peng, S. Zhe, X. Zhang, and Y. Qi, "Asynchronous distributed variational Gaussian process for regression," in *International Conference on Machine Learning*, pp. 2788–2797, PMLR, 2017.

[25] V. Drouet, S. Verel, and J.-M. Do, "Surrogate-assisted asynchronous multiobjective algorithm for nuclear power plant operations," in *Proceedings of the 2020 Genetic and Evolutionary Computation Conference*, pp. 1073–1081, 2020.

[26] J. Luo, A. Gupta, Y.-S. Ong, and Z. Wang, "Evolutionary optimization of expensive multiobjective problems with co-sub-pareto front gaussian process surrogates," *IEEE transactions on cybernetics*, vol. 49, no. 5, pp. 1708–1721, 2018.

[27] K. Deb, *Multi-objective optimization using evolutionary algorithms*, vol. 16. John Wiley & Sons, 2001.

[28] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.

[29] M. Wang and W. Deng, "Deep visual domain adaptation: A survey," *Neurocomputing*, vol. 312, pp. 135–153, 2018.

[30] J. Nam, W. Fu, S. Kim, T. Menzies, and L. Tan, "Heterogeneous defect prediction," *IEEE Transactions on Software Engineering*, vol. 44, no. 9, pp. 874–896, 2017.

[31] B. Sun, J. Feng, and K. Saenko, "Correlation alignment for unsupervised domain adaptation," in *Domain Adaptation in Computer Vision Applications*, pp. 153–171, Springer, 2017.

[32] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," *IEEE Transactions on Neural Networks*, vol. 22, no. 2, pp. 199–210, 2010.

[33] G. Csurka, "A comprehensive survey on domain adaptation for visual applications," in *Domain Adaptation in Computer Vision Applications*, pp. 1–35, Springer, 2017.

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

[35] M. Jiang, Z. Huang, L. Qiu, W. Huang, and G. G. Yen, "Transfer learning-based dynamic multiobjective optimization algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 4, pp. 501–514, 2018.

[36] M. Jiang, Z. Wang, S. Guo, X. Gao, and K. C. Tan, "Individual-based transfer learning for dynamic multiobjective optimization," *IEEE Transactions on Cybernetics*, 2020.

[37] M. N. Le, Y. S. Ong, S. Menzel, C.-W. Seah, and B. Sendhoff, "Multi co-objective evolutionary optimization: Cross surrogate augmentation for computationally expensive problems," in *2012 IEEE Congress on Evolutionary Computation*, pp. 1–8, IEEE, 2012.

[38] S. Huang, J. Zhong, and W. Yu, "Surrogate-Assisted Evolutionary Framework with Adaptive Knowledge Transfer for Multi-task Optimization," *IEEE Transactions on Emerging Topics in Computing*, 2019.

[39] A. T. W. Min, Y.-S. Ong, A. Gupta, and C.-K. Goh, "Multiproblem surrogates: transfer evolutionary multiobjective optimization of computationally expensive problems," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 1, pp. 15–28, 2017.

[40] K. Nigam and R. Ghani, "Analyzing the effectiveness and applicability of co-training," in *Proceedings of the Ninth International Conference on Information and Knowledge Management*, pp. 86–93, 2000.

[41] C. Sun, Y. Jin, and Y. Tan, "Semi-supervised learning assisted particle swarm optimization of computationally expensive problems," in *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 45–52, 2018.

[42] M. Baktashmotlagh, M. T. Harandi, B. C. Lovell, and M. Salzmann, "Unsupervised domain adaptation by domain invariant projection," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 769–776, 2013.

[43] J. Wang, Y. Chen, W. Feng, H. Yu, M. Huang, and Q. Yang, "Transfer learning with dynamic distribution adaptation," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 11, no. 1, pp. 1–25, 2020.

[44] K. M. Borgwardt, A. Gretton, M. J. Rasch, H.-P. Kriegel, B. Schölkopf, and A. J. Smola, "Integrating structured biological data by kernel maximum mean discrepancy," *Bioinformatics*, vol. 22, no. 14, pp. e49–e57, 2006.

[45] J. Shawe-Taylor, N. Cristianini, *et al.*, *Kernel methods for pattern analysis*. Cambridge University Press, 2004.

[46] S. Mika, G. Ratsch, J. Weston, B. Scholkopf, and K.-R. Mullers, "Fisher discriminant analysis with kernels," in *Neural Networks for Signal Processing IX: Proceedings of the 1999 IEEE Signal Processing Society Workshop (cat. no. 98th8468)*, pp. 41–48, Ieee, 1999.

[47] M. Jiang, Z. Huang, L. Qiu, W. Huang, and G. G. Yen, "Transfer learning-based dynamic multiobjective optimization algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 4, pp. 501–514, 2017.

[48] I. Triguero, S. García, and F. Herrera, "Self-labeled techniques for semi-supervised learning: taxonomy, software and empirical study," *Knowledge and Information Systems*, vol. 42, no. 2, pp. 245–284, 2015.

[49] G. Kostopoulos, S. Karlos, S. Kotsiantis, and O. Ragos, "Semi-supervised regression: A recent review," *Journal of Intelligent & Fuzzy Systems*, vol. 35, no. 2, pp. 1483–1500, 2018.

[50] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, pp. 92–100, 1998.

[51] Z.-H. Zhou and M. Li, "Tri-training: Exploiting unlabeled data using three classifiers," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 11, pp. 1529–1541, 2005.

[52] M. Li and Z.-H. Zhou, "Improve computer-aided diagnosis with machine learning techniques using undiagnosed samples," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 37, no. 6, pp. 1088–1098, 2007.

[53] P. Kang, D. Kim, and S. Cho, "Semi-supervised support vector regression based on self-training with label uncertainty: An application to virtual metrology in semiconductor manufacturing," *Expert Systems with Applications*, vol. 51, pp. 85–106, 2016.

[54] X. Sun, D. Gong, and W. Zhang, "Interactive genetic algorithms with large population and semi-supervised learning," *Applied Soft Computing*, vol. 12, no. 9, pp. 3004–3013, 2012.

[55] J. Knowles, "ParEGO: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 1, pp. 50–66, 2006.

[56] Q. Zhang, W. Liu, E. Tsang, and B. Virginas, "Expensive multiobjective optimization by MOEA/D with Gaussian process model," *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 3, pp. 456–474, 2009.

[57] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable multi-objective optimization test problems," in *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600)*, vol. 1, pp. 825–830, IEEE, 2002.

[58] Q. Zhang, A. Zhou, S. Zhao, P. N. Suganthan, W. Liu, and S. Tiwari, "Multiobjective optimization test instances for the cec 2009 special session and competition," tech. rep., The School of Computer Science and Electronic Engieering, University of Essex,Tech. Rep. CES-487, 2009.

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

[60] C. Yang, J. Ding, Y. Jin, and T. Chai, "Off-line data-driven multi-objective optimization: Knowledge transfer between surrogates and generation of final solutions," *IEEE Transactions on Evolutionary Computation*, 2020.

[61] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. Da Fonseca, "Performance assessment of multiobjective optimizers: An analysis and review," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 117–132, 2003.

[62] L. While, P. Hingston, L. Barone, and S. Huband, "A faster algorithm for calculating hypervolume," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 1, pp. 29–38, 2006.

[63] Y. Tian, R. Cheng, X. Zhang, and Y. Jin, "Platemo: A matlab platform for evolutionary multi-objective optimization [educational forum]," *IEEE Computational Intelligence Magazine*, vol. 12, no. 4, pp. 73–87, 2017.

[64] S. N. Lophaven, H. B. Nielsen, and J. Søndergaard, "DACE-a matlab kriging toolbox, version 2.0," 2002.

[65] C. M. Fonseca and P. J. Fleming, "On the performance assessment and comparison of stochastic multiobjective optimizers," in *International Conference on Parallel Problem Solving from Nature*, pp. 584–593, Springer, 1996.

[66] Y. Hua, Y. Jin, K. Hao, and Y. Cao, "Generating multiple reference vectors for a class of many-objective optimization problems with degenerate Pareto fronts," *Complex & Intelligent Systems*, vol. 6, no. 2, pp. 275–285, 2020.