

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

Xilu Wang, Yaochu Jin, Sebastian Schmitt, Markus Olhofer

2021

Preprint:

This is an accepted article published in Evolutionary Computation journal (ECJ).
The final authenticated version is available online at:
https://doi.org/10.1162/evco_a_00300

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

Xilu Wang

xilu.wang@surrey.ac.uk

Department of Computer Science, University of Surrey, Guildford, GU2 7XH, United Kingdom

Yaochu Jin

yaochu.jin@uni-bielefeld.de

Faculty of Technology, Bielefeld University, D-33615 Bielefeld, Germany
Department of Computer Science, University of Surrey, Guildford, GU2 7XH, United Kingdom

Sebastian Schmitt

sebastian.schmitt@honda-ri.de

Honda Research Institute Europe GmbH, Carl-Legien-Strasse 30, D-63073 Offenbach/Main, Germany

Markus Olhofer

markus.olhofer@honda-ri.de

Honda Research Institute Europe GmbH, Carl-Legien-Strasse 30, D-63073 Offenbach/Main, Germany

Abstract

Most existing multiobjective evolutionary algorithms (MOEAs) implicitly assume that each objective function can be evaluated within the same period of time. Typically, this is untenable in many real-world optimization scenarios where evaluation of different objectives involves different computer simulations or physical experiments with distinct time complexity. To address this issue, a transfer learning scheme based on surrogate-assisted evolutionary algorithms (SAEAs) is proposed, in which a co-surrogate is adopted to model the functional relationship between the fast and slow objective functions and a transferable instance selection method is introduced to acquire useful knowledge from the search process of the fast objective. Our experimental results on DTLZ and UF test suites demonstrate that the proposed algorithm is competitive for solving bi-objective optimization where objectives have non-uniform evaluation times.

Keywords

Multi-objective optimization, non-uniform evaluation times, transfer learning, co-surrogate, Gaussian process, surrogate-assisted evolutionary algorithm, Bayesian optimization

1 Introduction

Many real-world applications can be formulated as multi-objective optimization problems (MOPs) that simultaneously optimize two or more objective functions (Abraham and Jain, 2005). Usually, different objectives are conflicting to each other and there exists a set of optimal compromise solutions, known as Pareto optimal solutions. The whole set of Pareto optimal solutions in the decision space is called the Pareto set (PS),

and the projection of PS in the objective space is called Pareto front (PF). Over the past decades, multi-objective evolutionary algorithms (MOEAs) have been very successful in solving MOPs as they can obtain a set of optimal solutions in a single run (Zhou et al., 2011). However, MOEAs require an excessive number of function evaluations (FEs) before achieving a set of acceptable non-dominated solutions, preventing them from being applied to MOPs where FEs involve computationally intensive simulations or costly physical experiments (Jin and Sendhoff, 2002).

The limited budget of FEs is one main challenge in using MOEAs to solve expensive MOPs and surrogate models are useful tools for overcoming the computational obstacle (Jin, 2011; Allmendinger et al., 2017; Jin et al., 2019). Specifically, surrogate-assisted evolutionary algorithms (SAEAs) construct cheap surrogate models to evaluate candidate solutions instead of calling the original expensive objective functions. Various efficient models have been adopted in SAEAs, including radial basis function networks (Sun et al., 2017; Li et al., 2020), polynomial regressions (Zhou et al., 2005), support vector machines (Bourinet, 2016), random forest (Wang and Jin, 2020), and Gaussian processes (GPs) (Chugh et al., 2018b; Zhou et al., 2005). As it is impossible to build very accurate and highest-quality surrogate models for realistic objective functions using only a limited amount of training data, it makes sense to use surrogate models and the original objective functions together to secure the convergence of SAEAs, which is termed model management (Jin, 2011). More precisely, model management aims to identify a limited number of new candidate solutions to be evaluated by the original objective functions, which are then used for updating the surrogates. Along this line of research, Bayesian optimization has become very popular. Bayesian optimization refers to a class of black-box optimization algorithms that use GPs as surrogates in combination with an acquisition function (AF) which proposes new data samples to be evaluated with the true objective functions for model management (Shahriari et al., 2015).

In most existing MOEAs and SAEAs, all objective functions are assumed to have similar computational complexity and consequently the evaluation of each objective takes similar amount of time. However, it is common that objective functions in MOPs have distinct complexities. This class of MOPs for objectives with non-uniform evaluation times (or latencies) is first introduced by Allmendinger and Knowles (2013), and a general problem definition is further derived by Allmendinger et al. (2015). Based on their work, MOPs for objectives with non-uniform latencies corresponds to a situation where an MOP involves both the fast (computationally cheap) and slow (computationally expensive) objective functions or an MOP involves computationally intensive objectives where evaluation time of each objective drastically varies. For example, during the design optimization of an aircraft hull, crashworthiness assessment simulations take on the order of ten to a couple of hundred hours to complete, while the aerodynamic properties can be evaluated within few hours (Wang and Shan, 2006). In such cases, there is no reason to believe all objective functions in an MOP have similar computational complexity. In this work, we consider bi-objective optimization problems with a fast and a slow objective functions.

In the following, we introduce the related notations and assumptions of the problems under consideration.

- 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 Gaussian process models (GPs) are denoted as GP_s and GP_f , respectively. The ratio of the evaluation time of f_s to that of f_f is denoted by τ . Here, we assume that τ is an

integer larger than 1. Solutions that are evaluated by both f_f and f_s are denoted as \mathbf{X} . Consequently, the slow and fast objectives of \mathbf{X} are denoted by \mathbf{Y}_s and \mathbf{Y}_f , respectively. Due to the big difference in the evaluation time of the slow and fast objectives, to make full use of the time during which the slow objective is being evaluated, additional solutions, denoted by \mathbf{X}^a can be evaluated for the fast objective function (f_f), which are denoted by \mathbf{Y}_f^a . Here, we assume that evaluation of the fast and slow objective functions can be done in parallel.

- It is assumed that the computational time for building surrogates and for implementing the genetic operators is negligible compared to that for evaluating the expensive objectives. Consequently, the total computational time available for solving the problem under consideration is defined by the total budget of function evaluations. Specifically, the total budget is defined as the maximum number of function evaluations for the slow objective f_s , denoted as FE_s^{max} . Consequently, the maximum budget for the fast objective FE_f^{max} equals $FE_s^{max} * \tau$.

Given the above assumptions, most existing MOEAs and SAEAs are not able to efficiently solve MOPs with different evaluation latencies because they need to wait for completing the evaluation of the slow objective denoted as f_s (Allmendinger and Knowles, 2013). Only recently has some work been dedicated to dealing with such problems that considers objectives with non-uniform evaluation times. A pioneering study considering MOPs with delayed objectives has been done by Allmendinger and Knowles (2013). They investigated several strategies inspired by fitness inheritance to estimate the objective values (called pseudovalues) of the delayed objective. The pending objective values will be updated as long as the true values become available. This way, all candidate solutions will have the values of both the fast objective (true values) and the slow objective functions (pseudovalues), so that ranking and selection can be implemented at the frequency of evaluating the fast objective, denoted as f_f . Unsurprisingly, the performance of such a method heavily depends on the accuracy of the fitness approximation strategies. Subsequently, Allmendinger et al. (2015) gave a detailed description of MOPs with delayed objectives and suggested three schemes, termed *Waiting*, *Fast-first* and *Interleaving schemes*. Actually *Waiting* is the usual strategy that waits for all evaluations to be accomplished before selection is conducted, which therefore can be seen as the baseline method. By contrast, *Fast-first* aims to make the best use of the fast evaluations. To achieve this, *Fast-first* adopts a single-objective evolutionary algorithm (SOEA) to explore f_f first and subsequently evaluates the slow objective of a set of solutions (the best and distinct solutions) obtained in the single-objective search. Therefore, the objective values available for f_s are not fully exploited. Finally, *Interleaving* scheme considers not only the evaluation budgets for different objectives, but also tries to co-ordinate the different evaluation times of the different objectives. Two implementations, *Brood interleaving* and *Speculative interleaving*, of Interleaving scheme were proposed in Allmendinger et al. (2015). While *Brood interleaving* focuses more on maintaining diversity, *Speculative interleaving* uses an SOEA to optimize the fast objective when the slow objective is being evaluated.

The above studies exploit the fast objectives mainly with the help of evolutionary algorithms (EAs) and have not been extended to the context of SAEAs. However, as surrogates are a powerful tool to cope with limited budgets for function evaluations, it is a promising approach and deserves more attention, despite the fact that many factors need to be considered. Chugh et al. (2018a) first attempted to extend a Kriging-assisted EA (K-RVEA) (Chugh et al., 2018b) to bi-objective optimization where

objective functions require heterogeneous evaluation times and proposed a variation of K-RVEA, called HK-RVEA containing an SOEA for selecting training data and K-RVEA for updating surrogates. Specifically, HK-RVEA employs an SOEA to optimize f_f in the initialization, and genetic operators are adopted to generate additional solutions \mathbf{X}^a for f_f when waiting for the evaluation of new samples with f_s . It turns out that HK-RVEA performs well in cases where the ratio τ of the evaluation times for f_s and f_f is low. Wang et al. (2020b) extended an SAEA by applying a parameter based transfer learning (TL) technique to bi-objective problems with heterogeneous objectives, called (T-SAEA). To transfer knowledge acquired during the search experience on f_f for enhancing the performance of the Gaussian Process surrogate model for the slow objective, 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 . In a follow-up work, Wang et al. (2021) proposed an instance-based TL for SAEA, called Tr-SAEA, where a hybrid domain adaptation method is introduced to align the source domain (the space of the fast objective f_f) and the target domain (the space of the slow objective f_s) in a Reproducing Kernel Hilbert Space, obtaining a mapping matrix. Then, additional data for the fast objective y_f^a are mapped into the latent space to generate additional data for the slow objective y_s^a , and the corresponding decision variables \mathbf{X}_s^a of these data points are obtained by enforcing its objective values on f_s to be close to y_s^a in the latent space. To leverage the unlabeled \mathbf{X}_s^a , a co-training method is introduced to boost the quality of the surrogate of f_s . However, the performance of Tr-SAEA highly depends on the dimension of the latent space to be specified, and an optimization algorithm is required to generate the transferable solutions in the decision space.

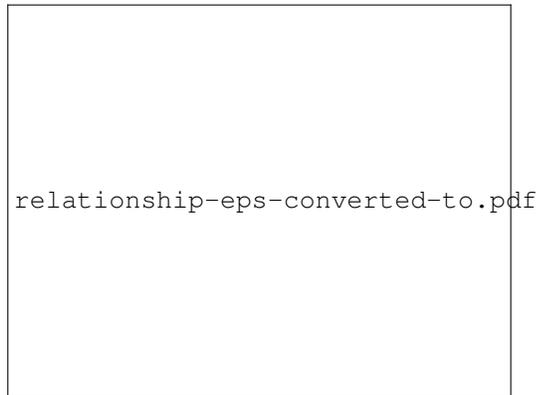


Figure 1: An illustration of the functional relationship between objectives for DTLZ2, where the red line is the PF.

Given that sufficient data samples of search parameter and objective value pairs $(\mathbf{X}^a, \mathbf{Y}_f^a)$ are available for the computationally cheap fast objective f_f , it is natural to leverage the knowledge hidden in $(\mathbf{X}^a, \mathbf{Y}_f^a)$ to speed up the search process for f_s . This is warranted since for solutions on the PF there exists a functional relationship between f_f and f_s (usually a trade-off relation) (Deb, 1999; Coello et al., 2007). In addition, such a functional relationship typically also approximately holds for solutions close to the PF. Take the bi-objective DTLZ2 test function (Deb, 1999) as an example, its Pareto front is defined by $f_1^2 + f_2^2 = 1$. For any solution that is not on the Pareto front, an error

function approximating the PF can be defined as $e(f_1, f_2) = f_1^2 + f_2^2 - 1$. The contour plot of this error function is shown in Fig. 1, indicating that the closer a solution is to the PF, the smaller the error is, and thus the stronger the functional relationship between f_1 and f_2 will be.

In this paper, we aim to more efficiently solve bi-objective optimization problems with delayed objectives with the help of the additional instances obtained from the optimization of f_f on the basis of SAEAs. To this end, we develop an instance TL scheme incorporated into a GP-assisted SAEA, which is a surrogate based interleaving method. The proposed algorithm, termed TC-SAEA, adopts a co-surrogate model to capture the underlying correlation between f_s and f_f on or near the PF supported by a transferable instance selection method to uncover useful knowledge. Here, GPs are employed as the surrogates and the adaptive acquisition function in Wang et al. (2020a) serves as the model management strategy. Similar to conventional SAEAs (Knowles, 2006; Naujoks et al., 2005; Zhang et al., 2009), a GP model is separately built for each objective function. The main contributions of the paper are as follows:

1. A GP surrogate model is built for the fast and slow objective functions, respectively, to assist the search of a bi-objective optimization problem with non-uniform fitness evaluation latencies. To enhance the quality of the surrogate for the slow objective function, a co-surrogate model is constructed to learn the difference between the slow and fast objective values. The co-surrogate is employed to generate synthetic data $(\mathbf{X}^a, \mathbf{Y}'^a)$ for training the surrogate of the slow objective, once the fast objective function of the additional solutions (\mathbf{X}^a) are evaluated.
2. An instance-based TL mechanism is introduced to select transferable data from the synthetic data generated by the co-surrogate. In order to identify reliable samples from $(\mathbf{X}^a, \mathbf{Y}'^a)$, a selection criterion based on the predicted value and the uncertainty of the prediction delivered by GP_s is suggested.

2 Preliminaries

In the following, we first introduce the related notations and assumptions of the problems under consideration. 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 Gaussian process models (GPs) are denoted as GP_s and GP_f , respectively. The ratio of the evaluation time of f_s to that of f_f is denoted by τ . Here, we assume that τ is an integer larger than 1. Solutions that are evaluated by both f_f and f_s are denoted as \mathbf{X} . Consequently, the slow and fast objectives of \mathbf{X} are denoted by \mathbf{Y}_s and \mathbf{Y}_f , respectively. Due to the big difference in the evaluation time of the slow and fast objectives, to make full use of the time during which the slow objective is being evaluated, additional solutions, denoted by \mathbf{X}^a can be evaluated for the fast objective function (f_f), which are denoted by \mathbf{Y}_f^a . Here, we assume that evaluation of the fast and slow objective functions can be done in parallel.

It is assumed that the computational time for building surrogates and for implementing the genetic operators is negligible compared to that for evaluating the expensive objectives. Consequently, the total computational time available for solving the problem under consideration is defined by the total budget of function evaluations. Specifically, the total budget is defined as the maximum number of function evaluations for the slow objective f_s , denoted as FE_s^{max} . Consequently, the maximum budget for the fast objective f_f equals $FE_s^{max} * \tau$.

2.1 Kriging model

Gaussian processes can be traced back to Kriging (Matheron, 1963) and have attracted increasing attention in machine learning. Meanwhile, GPs are also known as Kriging in efficient global optimization (Jones et al., 1998) as well as in SAEAs (Jin and Sendhoff, 2002; Allmendinger et al., 2017). Therefore, these two terms will be used interchangeably in the rest of the paper. A major advantage of GPs models over other surrogate models is that they not only provide a prediction for the objective value but also a confidence level of the prediction, based on which we can obtain a prediction interval. The uncertainty information can be used in an acquisition function to select new samples for effectively updating the GPs.

A GP is a collection of random variables that have a joint multivariate Gaussian distribution for any finite set of inputs. Consider a training set that includes N samples (\mathbf{X}, \mathbf{Y}) , where $\mathbf{X} = [\mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^N]^T$, $\mathbf{Y} = [y^1, y^2, \dots, y^N]^T$, $i = 1, 2, \dots, N$. A GP can be specified by a mean and a covariance function, denoted by μ and σ . The covariance function is used to describe the correlation between y^i and y^j related to the distance between \mathbf{x}^i and \mathbf{x}^j . In general, the squared exponential function with additional hyperparameters is employed to calculate the correlation:

$$d(\mathbf{x}^i, \mathbf{x}^j) = \sum_{k=1}^m \theta_k |x_k^i - x_k^j|^{p_k} \quad (1)$$

$$\text{Corr}(\mathbf{x}^i, \mathbf{x}^j) = \exp[-d(\mathbf{x}^i, \mathbf{x}^j)] \quad (2)$$

where $p_k \in (0, 1)$ controls the smoothness of the function in terms of the k -th dimension, and $\theta_k > 0$ denotes the importance of this dimension. When there are N training data, an $N \times N$ correlation matrix \mathbf{C} will be obtained,

$$\mathbf{C} = \begin{pmatrix} \text{Corr}(\mathbf{x}^1, \mathbf{x}^1) & \cdots & \text{Corr}(\mathbf{x}^1, \mathbf{x}^N) \\ \vdots & \ddots & \vdots \\ \text{Corr}(\mathbf{x}^N, \mathbf{x}^1) & \cdots & \text{Corr}(\mathbf{x}^N, \mathbf{x}^N) \end{pmatrix} \quad (3)$$

As a result, the hyperparameters will determine a GP model, which can be estimated by maximizing the following likelihood function,

$$\psi(\theta_1, \dots, \theta_N, p_1, \dots, p_N) = -\frac{1}{2} (N \ln \sigma^2 + \ln \det(\mathbf{C})) \quad (4)$$

Therefore, the estimates $\hat{\mu}$ and $\hat{\sigma}^2$ for the true values μ and σ will be obtained

$$\hat{\mu} = \frac{\mathbf{1}^T \mathbf{C}^{-1} \mathbf{y}}{\mathbf{1}^T \mathbf{C}^{-1} \mathbf{1}} \quad (5)$$

$$\hat{\sigma}^2 = \frac{(\mathbf{y} - \mathbf{1} \hat{\mu})^T \mathbf{C}^{-1} (\mathbf{y} - \mathbf{1} \hat{\mu})}{N} \quad (6)$$

where $\mathbf{1}$ denotes an $N \times 1$ column vector of ones. Based on the given parameters, the GP model can predict the mean value together with a variance for a new data point \mathbf{x}^{new} (which corresponds to the decision variables of a candidate solution when the GP is used as a surrogate),

$$f(\mathbf{x}^{new}) = \hat{\mu} + \mathbf{r}^T \mathbf{C}^{-1} (\mathbf{y} - \mathbf{1} \hat{\mu}) \quad (7)$$

$$\hat{\sigma}(\mathbf{x}^{new})^2 = \hat{\sigma}^2 \left[1 - \mathbf{r}^T \mathbf{C}^{-1} \mathbf{r} + \frac{(1 - \mathbf{r}^T \mathbf{C}^{-1} \mathbf{r})^2}{\mathbf{1}^T \mathbf{C}^{-1} \mathbf{1}} \right] \quad (8)$$

where $\mathbf{r} = (\text{Corr}(\mathbf{x}^{new}, \mathbf{x}^1), \dots, \text{Corr}(\mathbf{x}^{new}, \mathbf{x}^N))^T$ presents a correlation vector between \mathbf{x}^{new} and each element \mathbf{x}^i in \mathbf{X} .

2.2 Acquisition Function

In SAEAs, it is of paramount importance to select the right candidate solutions to be evaluated using the real objective function so that the quality of the surrogates can be improved as much as possible and that the surrogates are able to efficiently guide the evolutionary search to the optimum. This is known as model management, which plays a key role in SAEAs (Jin et al., 2000; Allmendinger et al., 2017).

In GP-assisted optimization such as Bayesian optimization, model management is done by optimizing an acquisition function (AF). AFs can generally be divided into three categories, improvement-based, information-based and optimistic (Noè and Husmeier, 2018). Improvement-based methods aim to select solutions (samples) with the highest probability of improving upon the best observed sample so far, such as probability of improvement (PI) (Kushner, 1964) and expected improvement (EI) (Jones et al., 1998). Instead of querying at solutions where we expect to obtain promising fitness values, information-based methods focus on finding out new solutions containing more information about the location of the optimum. For example, entropy search (ES) (Hennig and Schuler, 2012) and predictive entropy search (PES) (Hernández-Lobato et al., 2014) both query at solutions with the largest reduction of uncertainty of the surrogate. Optimistic methods design acquisition functions based on an optimistic attitude towards uncertainty. A representative one is the lower confidence bound (LCB) (Cox and John, 1992) combining the uncertainty with the predicted objective values (Liu et al., 2012). An adaptive acquisition function based on LCB is introduced by Wang et al. (2020a) to tune the trade-off parameter according to the search dynamics, achieving a better balance between the convergence and diversity. In this work, we adopt this acquisition function to select new samples to be evaluated by the original objective functions.

2.3 Instance-based Transfer Learning

Among various TL techniques, instance-based transfer learning is intuitively appealing: although the source domain data cannot be reused directly, there are certain parts of the data that can still be reused together with a few labeled data in the target domain (Pan and Yang, 2009). In the machine learning community, many instance-based transfer learning approaches have been proposed, most of which adopts instance weighting strategies. For example, TrAdaBoost (Dai et al., 2007) combines the labeled source-domain and target-domain instances together as the training data, and adjusts the weights of instances to reduce the negative effects of the source domain. In CP-MDA Chattopadhyay et al. (2012), source classifiers label the unlabeled target data using a weighting scheme based on the similarities between the conditional probabilities of the source and target domain data.

Measuring the relatedness or similarity between the source and target domains is a vital issue for studying transferability (Duan et al., 2012). For example, in the context of machine learning, Yang et al. (2015) employed a directed cyclic network to denote transferred weights from a source domain to a target domain, and then the relatedness between the given domains is evaluated through the transfer weights. Gong et al. (2012) introduced a metric called rank of domain (ROD) to rank a list of source domains by evaluating each source domain in terms of the degree of overlap and similarity to the target domain. Similarly, in the context of evolutionary computation, the similarity between tasks is measured based on the KL-divergence of the distributions of the training data sets in (Huang et al., 2019). Min et al. (2017) proposed a model-based transfer stacking approach to combining multiple surrogates together with a linear

meta-regression model, where the meta-regression coefficients indicate the similarities between the source and target optimization tasks.

3 Proposed Algorithm

We give the related notations in the following, before introducing the proposed algorithm.

As we mentioned in Section I, surrogate models GP_s and GP_f are constructed for f_s and f_f , respectively. $D_s = (\mathbf{X}, \mathbf{Y}_s)$ is defined as the training data set for GP_s , while $D_f = \left\{ (\mathbf{X}, \mathbf{Y}_f), (\mathbf{X}^a, \mathbf{Y}_f^a) \right\}$ is defined as the training data set for GP_f . Apart from GP_s and GP_f , we construct a co-surrogate model GP_c to approximate the difference (denoted as function f_c) between f_s and f_f , indirectly describing the relationship between f_s and f_f on or near the PF. Specifically, the input-output pairs $(\mathbf{X}, \mathbf{Y}_c)$ are defined as the training data set D_c for GP_c , where $\mathbf{Y}_c = \mathbf{Y}_s - \mathbf{Y}_f$. Note that the training data for GP_c are promising solutions, as they are selected according to the AF. This way, for the additional solutions \mathbf{X}^a (defined in Section I), the synthetic objective values \mathbf{Y}_s^a of f_s can be calculated by $\mathbf{Y}_c^a + \mathbf{Y}_f^a$, where \mathbf{Y}_c^a is predicted by GP_c . The input-output pairs $(\mathbf{X}^a, \mathbf{Y}_s^a)$ are defined as an auxiliary data set D_a for f_s . To select the most reliable data in the auxiliary data set D_a , a selection criterion is designed by calculating a confidence interval CI . Specifically, for the additional solutions \mathbf{X}^a , GP_s predicts objective values \mathbf{Y}_s^a together with their variances σ_s^a . Subsequently, $CI = \mathbf{Y}_s^a \pm \sigma_s^a$ is defined as a selection threshold. Then the selected transferable data set is defined as $D_t = (\mathbf{X}^t, \mathbf{Y}_s^t)$.



Figure 2: An illustration of the proposed transfer learning scheme. Left: Augmentation of D_s using the proposed transfer learning scheme. Right: A diagram for the transfer learning method adopted in this work.

For solving MOPs with delayed objectives, the algorithm should be able to quickly find a set of diverse and well converged solutions due to the limited budget of function

evaluations. While the previous studies only focus on the utilization of per-objective evaluation budgets and straightforward instance transmission, the main motivation of this work is to augment the data set D_s by transferring knowledge acquired from the additional solutions so as to accelerate the convergence towards the Pareto front. To achieve this, there are two key challenges. One is how to generate synthetic objective values \mathbf{Y}'_s^a to the additional solutions \mathbf{X}^a when the observed data for f_s is scarce; the other is how to identify the most reliable one from the auxiliary data set $(\mathbf{X}^a, \mathbf{Y}'_s^a)$ as the transferable data set D_t . To address these issues, we propose here a transfer learning scheme to be incorporated into a GP-based SAEA, illustrated in Fig. 2. As shown in the left figure, an augmented data set D_s can be obtained by transferring some useful data from the auxiliary data set $(\mathbf{X}^a, \mathbf{Y}'_s^a)$. The right figure depicts the workflow diagram for the proposed transfer learning scheme. Specifically, a co-surrogate GP_c is built to approximate the functional relationship between the \mathbf{X} and $(\mathbf{Y}_s - \mathbf{Y}_f)$ to bridge the gap between f_s and f_f . This way, synthetic values \mathbf{Y}'_s^a for the additional solutions can be calculated by adding \mathbf{Y}'_c to \mathbf{Y}_f^a , thereby obtaining the auxiliary data set $D_a = (\mathbf{X}^a, \mathbf{Y}'_s^a)$. However, some synthetic values may be detrimental if the predicted values are extremely inaccurate. Hence, we can take advantage of the confidence level $CI = \mathbf{Y}'_s^a \pm \sigma_s^a$ given by GP_s as a selection threshold to filter out the unreliable auxiliary samples.

3.1 Algorithm Framework

The framework of the proposed TC-SAEA ¹ is shown in Fig. 3 and its pseudo code is outlined in Algorithm 1. In the following, we detail the main components of TC-SAEA.



Figure 3: The framework of TC-SAEA.

The algorithm starts with using the Latin Hypercube Sampling (LHS) method to generate an initial population \mathbf{X} , and the corresponding objective values \mathbf{Y}_s and \mathbf{Y}_f

¹The source code is available at <https://github.com/xw00616/TC-SAEA.git>.

Algorithm 1 The framework of TC-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; N_{max} : the maximum number of training data points.

Output: The optimized solutions in D_s ;

- 1: Initialization: Use the Latin Hypercube Sampling method to sample an initial population \mathbf{X} ; calculate $\mathbf{Y}_s, \mathbf{Y}_f$ and the difference $\mathbf{Y}_c = \mathbf{Y}_s - \mathbf{Y}_f$. Set $D_s^0 = (\mathbf{X}, \mathbf{Y}_s)$, $D_f^0 = (\mathbf{X}, \mathbf{Y}_f)$ and $D_c^0 = (\mathbf{X}, \mathbf{Y}_c)$. Run an SOEA to optimize f_f and save data in D_f^0 ; set $FE_s = size(D_s^0)$ and $IterNum = 1$.
- 2: **while** $FE_s \leq FE_s^{max}$ **do**
- 3: Limit the size of the training data to N_{max} and train GP_f with the training data D_f for f_f ;
- 4: Limit the size of the training data to N_{max} and train GP_s with the training data D_s for f_s ;
- 5: Limit the size of the training data to N_{max} and train GP_c with the training data D_c for f_c ;
- 6: **while** $w \leq w_{max}$ **do**
- 7: //Using the surrogate in the EA//
- 8: Run an MOEA to find samples for updating GPs;
- 9: $w = w + 1$;
- 10: **end while**
- 11: Use the adaptive acquisition function to evaluate the individuals in the optimized population;
- 12: Use the angle-penalized distance (APD) to determine u new solutions \mathbf{X}^{new} to be evaluated with both f_s and f_f , and add new data to D_s ;
- 13: Sample $u * (\tau - 1)$ additional solutions \mathbf{X}^a around \mathbf{X}^{new} to be evaluated by f_f , and add new data \mathbf{X}^{new} and \mathbf{X}^a to D_f ;
- 14: //Using the co-surrogate to select the transferable instances//
- 15: Use GP_c to predict the values of \mathbf{Y}_c^a on \mathbf{X}^a and calculate the synthetic values $\mathbf{Y}_s'^a = \mathbf{Y}_c^a + \mathbf{Y}_f^a$;
- 16: Use GP_s to predict the objective values \mathbf{Y}_s^a and the uncertainty σ_s^a on \mathbf{X}^a , calculate the confidence interval $CI = \mathbf{Y}_s^a \pm \sigma_s^a$;
- 17: **if** $\mathbf{Y}_s'^a \in CI$ **then**
- 18: The corresponding data $(\mathbf{X}^t, \mathbf{Y}_s^t)$ is transferable and saved in D_t ;
- 19: **end if**
- 20: **if** $IterNum \bmod \tau == 0$ **then**
- 21: Train GP_s with D_s
- 22: **else**
- 23: Train GP_s with $D_s + D_t$
- 24: **end if**
- 25: Update $FE_s = FE_s + u$, $IterNum = IterNum + 1$;
- 26: **end while**
- 27: Return the optimized solutions in D_s ;

on \mathbf{X} are calculated with the true objective functions f_s and f_f . The difference between the two objective values $\mathbf{Y}_c = \mathbf{Y}_s - \mathbf{Y}_f$ can be obtained for \mathbf{X} . Let $D_s = (\mathbf{X}, \mathbf{Y}_s)$, $D_f = (\mathbf{X}, \mathbf{Y}_f)$ and $D_c = (\mathbf{X}, \mathbf{Y}_c)$. In the initialization, additional function evaluations for f_f are available when waiting for the time-consuming evaluation of f_s . There are different possible ways of exploiting the additional evaluations of the fast function, and in this work we follow the *Interleaving scheme* in (Allmendinger et al., 2015), where an SOEA is employed to optimize the fast objective function f_f and the obtained data is saved in D_f . Subsequently, GP_s , GP_f and GP_c are trained with data sets D_s , D_f and D_c , respectively. To reduce the computational complexity, we used the same method in ParEGO (Knowles, 2006) to limit the size of the training data. Specifically, when the number of training data exceeds a predefined upper bound N_{max} , a selection strategy will be triggered: the first half $N_{max}/2$ training data are the best ones sorted according to the objective values, and the rest half are randomly sampled from the training dataset without replacement.

Here, we use RVEA (Cheng et al., 2016) as the baseline MOEA to perform multi-objective optimization assisted by the surrogate models GP_s and GP_f for function evaluations, instead of directly searching on the real objective functions f_s and f_f . In every

20 generations, the adaptive acquisition function (AFF) (Wang et al., 2020a) is then applied to evaluate each individual in the population, subsequently, the angle-penalized distance (APD) selection criterion in RVEA is adopted to identify a predefined number of new samples \mathbf{X}^{new} according to the AFF. Note that samples in \mathbf{X}^{new} are evaluated with the real objective functions f_s and f_f , thereby involving the time-consuming function evaluation f_s again. Hence, different from conventional SAEAs, the fast objective function can be called more times to evaluate $u * (\tau - 1)$ (where τ is an integer larger than 1 as defined in Section I) additional new solutions \mathbf{X}^a sampled using LHS around \mathbf{X}^{new} .

It is apparent that there is a considerable amount of observed data for f_f , while the data for f_s is inadequate due to the different evaluation times. As a result, the convergence of f_s will be slow and a bias towards the fast objective function will be introduced in the search (Allmendinger et al., 2015), posing challenges for SAEAs to efficiently solve such problems. Thus, it would make sense to augment the data for f_s to alleviate the problem in training GP_s caused by a limited evaluation budget for f_s . Therefore, a transfer learning scheme is proposed to make use of the knowledge extracted from the additional solutions \mathbf{X}^a . The proposed scheme includes two main components: the GP co-surrogate model, denoted as GP_c , to indirectly describe the functional relationship between the two objective functions, and the transferable instance selection criterion to determine transferable data from the auxiliary data set D_a . Specifically, given the additional solutions \mathbf{X}^a , an auxiliary data set $D_a = (\mathbf{X}^a, \mathbf{Y}'_s{}^a)$ can be attained, where the synthetic objective values are calculated using the co-surrogate and the true objective values \mathbf{Y}_f of f_f . Note that it is non-trivial to build effective surrogates, which means that the synthetic data generated by GP_c may be subject to large errors. Consequently, the instance selection method is executed to identify transferable data D_t from the auxiliary data set to promote the performance of TC-SAEA in terms of both optimization quality and convergence speed. The selection criterion is designed by calculating $CI = \mathbf{Y}_s^a \pm \sigma_s^a$, where \mathbf{Y}_s^a and σ_s^a are provided by GP_s . If the synthetic objective values $\mathbf{Y}'_s{}^a$ are within the bound CI , the corresponding auxiliary data is selected as transferable data D_t .

For τ iterations, GP_s is updated once using the data in D_s without D_t ; otherwise, GP_s is trained using both the transferable data set D_t and the observed data D_s evaluated with the real objective function f_s . Augmenting the data for f_s with the help of the transfer learning scheme can alleviate the problem resulting from the insufficient data for f_s .

3.2 Co-surrogate Model

As shown in instance-based transfer learning, it is a promising way to utilize the source-domain instances, i.e. $(\mathbf{X}^a, \mathbf{Y}_f^a)$ to augment the training data for GP_s . As we introduced before, the key question is how to estimate the functional relationship between the two objectives. In bi-objective optimization in which the objectives have different evaluation times, there are only two objective functions and many solutions evaluated with each objective function may be the same due to the population-based SAEA. As a result, the slow objective function f_s only has one auxiliary task f_f and the transferable knowledge is limited to the additional data achieved on the fast objective f_f ; besides, it is impracticable to use the objective values \mathbf{Y}_s and \mathbf{Y}_f (column vectors) to directly model the relationship between f_s and f_f . Therefore, the previously discussed TL techniques are not well suited for the problem considered in this work.

To address this issue, we build a regression model, called co-surrogate model GP_c ,

to estimate the underlying relationship between f_s and f_f . Note that there is a certain functional relationship between the two objectives, which may only loosely hold in most of the search space but does hold for all Pareto optimal solutions (Deb, 1999; Coello et al., 2007). In other words, the closer to the Pareto front, the stronger the functional relationship between f_f and f_s will be. Recall that the data for training the co-surrogate GP_c are promising solutions, as they are all selected according to the acquisition function. Therefore, GP_c can describe the underlying relationship between the two objectives on/close to the PF by learning from $(\mathbf{X}, \mathbf{Y}_c)$, where \mathbf{Y}_c is the difference between the slow and fast objectives as defined in Equation (9). It should be stressed that GP_c is an approximation of \mathbf{Y}_c , which is a function of \mathbf{X} that can be linear or nonlinear. More discussions about the co-surrogate will be provided in Section 4.7.

Given N inputs $\mathbf{X} = [x^1, x^2, \dots, x^N]^T$ evaluated on both the fast and slow objectives $[\mathbf{Y}_f, \mathbf{Y}_s]$, where $\mathbf{Y}_f = [y_f^1, y_f^2, \dots, y_f^N]^T$ and $\mathbf{Y}_s = [y_s^1, y_s^2, \dots, y_s^N]^T$, the difference of the two objectives can be calculated as follows:

$$\mathbf{Y}_c = [y_s^1 - y_f^1, y_s^2 - y_f^2, \dots, y_s^N - y_f^N]^T \quad (9)$$

We can build a co-surrogate by training a GP with the input-output pairs $(\mathbf{X}, \mathbf{Y}_c)$. Compared with GP_s , GP_c learns from the solutions evaluated on both objectives from a different perspective since the difference between f_s and f_f is the output. Once the additional solutions \mathbf{X}^a associated with \mathbf{Y}_f^a are obtained, the co-surrogate GP_c can predict the difference \mathbf{Y}_c^a between f_s and f_f on \mathbf{X}^a , then the synthetic values of f_s on \mathbf{X}^a are determined as follows:

$$\mathbf{Y}_s'^a = \mathbf{Y}_c^a + \mathbf{Y}_f^a \quad (10)$$

Therefore, the auxiliary data set for f_s is defined as $D_a = (\mathbf{X}^a, \mathbf{Y}_s'^a)$. A major advantage of using GP_c is that the synthetic data generated for GP_s can make use of the true objective values of f_f and may improve the quality of the synthetic values of f_s , which is confirmed in Section 4.5.

3.3 Transferable Instance Selection

The co-surrogate model is constructed to connect the two objective functions by modelling the underlying relatedness so that the auxiliary data set can be generated and used to augment the training dataset for GP_s . Similar to the intuitive approaches in (Allmendinger and Knowles, 2013; Chattopadhyay et al., 2012), the synthetic values or pseudo labels can be utilized directly to augment the training dataset. Typically, the performance of these proposed algorithms may be improved by directly leveraging such synthetic objective values or pseudo labels, which, however, heavily depends on the quality of the synthetic objective values and pseudo labels. It is well recognized that identifying transferable data plays a vital role in transfer learning to alleviate negative transfer (Pan and Yang, 2009). In the auxiliary data set $D_a = (\mathbf{X}^a, \mathbf{Y}_s'^a)$ obtained so far, the accuracy of the synthetic objective values $\mathbf{Y}_s'^a$ cannot be guaranteed, because these data are generated by only considering the underlying relationship between f_s and f_f . For example, if the instances are incorrectly predicted, the corresponding synthetic values or pseudo labels are likely to conflict with the real values/labels, thereby misleading the training of the model and undermining the model's performance. Hence, it is expected to transfer the reliable instances to ensure positive transfer learning and filter out the inaccurate synthetic instances. Besides, considering the fact that the co-surrogate is not as accurate as we expect and the functional relationship of the two objectives may be weak, it is necessary to carefully identify which instances can be

transferred to train the surrogate model GP_s and further improve the performance of TC-SAEA in terms of convergence and diversity.

To select a subset of the data in D_a as useful information for training GP_s , the confidence level predicted by GP_s is adopted to assess the transferability of the auxiliary data $D_a = (\mathbf{X}^a, \mathbf{Y}'^a)$. Note that the GP model GP_s for f_s is initialized by training with $D_s = (\mathbf{X}, \mathbf{Y}_s)$, where \mathbf{Y}_s is the real values of f_s . Therefore, it is not unlikely that the predicted objective value of an instance in D_a is inconsistent with the prediction of GP_s . The hypothesis we make here is that the prediction given by the co-surrogate GP_c is unreliable if the predicted value is out of the boundary predicted by GP_s . Specifically, for the additional solutions \mathbf{X}^a , GP_s provides its prediction in terms of the mean objective value \mathbf{Y}_s^a and the standard deviation σ_s^a . Accordingly, the confidence interval is defined as

$$CI = \mathbf{Y}_s^a \pm \sigma_s^a \quad (11)$$

Therefore, if a synthetic value in \mathbf{Y}'^a is within the range of CI , the associated instance is believed reliable and will be saved in the transferable data set D_t . On the contrary, if not, the synthetic value is regarded as unreliable in terms of the current GP model GP_s for f_s . Fig. 4 presents an illustrative example of how the transferable instance selection works. A 1D GP model GP_s is trained with three training data, the GP model can then predict the mean fitness and the uncertainty of any new data point. For instance, for a given additional solution x_1^a , the synthetic value $y_s(x_1^a)$ can be calculated with the help of GP_c , the mean objective value $y_s(x_1^a)$, and the corresponding variances $\sigma_s^a(x_1^a)$ can be predicted by GP_s . Since the predicted objective value $y_s'(x_1^a)$ is out of the interval CI (indicated by the thick red line), this synthetic data will be considered as unreliable and will not be added to D_t . It is worth mentioning that, from the exploration perspective, selecting instances whose synthetic values are out of CI may facilitate more exploratory search. However, note that the instances with the synthetic values are adopted to train the surrogate GP_s rather than being evaluated with the true objective function f_s before updating GP_s . Therefore, the key issue is to guarantee the reliability of the synthetic values to reduce the chance of misleading the search assisted by the surrogate.

4 Comparative Studies

In this section, two groups of bi-objective test problems and a test function with a controllable correlation between objectives are employed to examine the efficiency of the proposed algorithm and we assume f_2 (objective 2) in the benchmark problems is the slow objective f_s , as this work focuses on bi-objective optimization problems with different evaluation times. We run each algorithm under comparison on each benchmark problem for 20 independent times, and the inverted generational distance (Bosman and Thierens, 2003) and hypervolume (While et al., 2006) are used as the performance measures. The Wilcoxon rank sum test is adopted to compare the mean results achieved by TC-SAEA and other algorithms under comparison at a significance level of 0.05. Symbol “(+)” and “(-)” indicate that the proposed algorithm performs significantly better or significantly worse than the compared algorithm, respectively, while “(\approx)” means there is no significant difference between them.

In the following section, a brief introduction to the test problems and the performance indicators is given at first, followed by a description of the details of the experimental settings concerning the optimization algorithms. The comparative experimental results of TC-SAEA and some state-of-the-art delay-handling methods are presented and discussed. To get a deeper insight into the proposed strategies, TC-SAEA is also



Figure 4: An illustrative example of transferable instance selection for a 1D Gaussian process GP_s with three training data points. For the three new data points (x_1^a , x_2^a and x_3^a), the corresponding confidence intervals (as shown in thick red line) and the synthetic values (red points) are obtained by GP_s and GP_c , respectively. The solid line and the shaded area indicate the mean and confidence intervals estimated with the GP model GP_s . Here, $y'_s(x_2^a)$ is considered to be reliable.

compared with its three variants. Then, the impact of the correlation between objectives on the performance is investigated. Lastly, a sensitivity analysis is given.

4.1 Test Problems

Numerical experiments are conducted on sixteen bi-objective benchmark problems, nine of them taken from the DTLZ test suite (Deb et al., 2002), including DTLZ1 to DTLZ7 and two modified counterparts (DTLZ1a and DTLZ3a) of DTLZ1 and DTLZ3, and the remaining seven (UF1-UF7) from the UF test suite (Zhang et al., 2008). 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 (Deb et al., 2002),

$$g = 100[5 + \sum_{i \in \{1, \dots, n\}} (x_i - 0.5)^2 - \cos(20\pi(x_i - 0.5))], i = 1, \dots, n. \quad (12)$$

where n denotes the number of decision variables. As indicated in (Yang et al., 2019), 20π within the cosine term triggers excessively ruggedness. Consequently, 2π is adopted to reduce the complexity to a reasonable level. As recommended in (Deb et al., 2002), the number of decision variables for the DTLZ test instances is set to $n = M + K - 1$, where $K = 5$ is adopted for DTLZ1 and DTLZ1a, $K = 10$ is used for DTLZ2 to DTLZ6 as well as DTLZ3a, and $K = 20$ is employed in DTLZ7. M repre-

sents the number of objectives, here $M = 2$. The number of decision variables for UF test suite is set as 30.

To demonstrate the sensitivity of the proposed algorithm on the correlation among objectives, a continuous version of the mapped OneMax problem (cm-OneMax) in (Chugh et al., 2018b) is used. 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:

$$f = (f_1, f_2) = (n(\mathbf{x}), n(\mathbf{x}^{map})) \quad (13)$$

where $x_i^{map} = |x_i - map_i|$, $i = 1, \dots, n$, 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 correlation $corr \in [-1, 1]$ 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 correlation, map_i is set to zero with a probability of $(1 + corr)/2$.

4.2 Performance Metrics

The inverted generational distance (IGD) is adopted to assess the performance of the algorithms in terms of the convergence and diversity of the obtained non-dominated solutions. The PlatEMO toolbox (Tian et al., 2017) is used to calculate values of IGD in our experiments. Let P^* be a set of uniformly distributed solutions sampled from the objective space along the true PF, and P be an obtained approximation to the PF. *IGD* measures the IGD from P^* to P as follows:

$$IGD(P^*, P) = \frac{\sum_{v \in P^*} d(v, P)}{|P^*|} \quad (14)$$

where $d(v, P)$ is the minimum Euclidean distance between v and all points in P . The smaller IGD value, the better the achieved solution set is.

4.3 Experimental Settings

In order to test the effectiveness of the proposed TL scheme, two state-of-the-art SAEAs, HK-RVEA (Chugh et al., 2018b) and T-SAEA (Wang et al., 2020b), and a representative non-surrogate based method including four algorithm schemes in (Allmendinger et al., 2015) are adopted for comparison with TC-SAEA. Specifically, HK-RVEA and T-SAEA are GP-based EA for solving bi-objective optimization problems under consideration; *Waiting*, *Fast-first*, *Brood interleaving* (BI) and *Speculative interleaving* (SI) in (Allmendinger et al., 2015) are four distinct EAs without surrogates. We also include K-RVEA (Chugh et al., 2018a) for comparison, which is a GP-based SAEA and proposed for solving MOPs in undelayed environment. Here, we adopt RVEA as the multi-objective optimizer and the standard genetic algorithm (GA) as the single-objective optimizer in all the compared algorithms for the sake of fairness.

TC-SAEA and its variants are implemented in MATLAB R2019a 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 (Lophaven et al., 2002). The general parameter settings in the experiments are given as follows:

- The number of initial training points $N_{train} = 100$.
- The maximum number of generations before updating GPs $w_{max} = 20$.

- The number of new samples for updating GPs .
- 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 $FE_s^{max} = 200$.

4.4 Comparison with Some State-of-the-art algorithms

Table 1: Statistical results of the IGD values obtained by Waiting, Fast-first, BI, SI, K-RVEA, HK-RVEA, T-SAEA and TC-SAEA with $FE_s^{max} = 200$ and $\tau = 5$

Problem	Waiting		Fast-first		BI		SI		K-RVEA		HK-RVEA		T-SAEA		TC-SAEA	
	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	29.8 + 17.8	42.2 + 10.5	21.7 + 11.9	20.1	8.16							
DTLZ1a	14.2 + 8.48	2.62 + 0.25	15.6 + 6.54	28.6 + 8.22	1.03 + 0.32	0.52 + 0.18	1.06 + 1.00	0.36	0.04							
DTLZ2	0.24 + 0.05	0.80 + 0.08	0.36 + 0.05	0.38 + 0.03	0.13 + 0.06	0.10 + 0.02	0.05 \approx 0.03	0.02	0.00							
DTLZ3	349 + 83.4	549 + 142	357 + 74.6	462 + 67.5	385 + 59.4	354 + 41.9	203 \approx 100	132	79.28							
DTLZ3a	227 + 75.4	546 + 86.9	313 + 83.1	406 + 96.6	73.7 + 29.2	14.9 + 5.27	5.34 + 37.5	2.30	0.66							
DTLZ4	0.51 + 0.32	0.78 + 0.11	0.54 + 0.06	0.65 + 0.10	0.45 \approx 0.23	0.23 - 0.11	0.60 + 0.13	0.44	0.13							
DTLZ5	0.27 + 0.06	0.86 + 0.10	0.35 + 0.04	0.39 + 0.03	0.14 + 0.05	0.09 + 0.02	0.05 \approx 0.02	0.03	0.00							
DTLZ6	7.31 + 0.52	8.79 + 0.11	7.63 + 0.44	8.26 + 0.13	5.14 + 0.77	4.10 + 0.54	2.56 \approx 1.21	2.62	1.95							
DTLZ7	4.41 + 0.62	7.53 + 0.39	5.53 + 0.47	5.57 + 0.68	5.54 + 0.47	0.06 \approx 0.05	1.15 + 0.91	0.05	0.08							
UF1	1.01 + 0.14	0.49 + 0.04	0.36 + 0.02	0.42 + 0.05	1.20 + 0.11	0.23 + 0.02	0.19 \approx 0.02	0.19	0.02							
UF2	0.50 + 0.07	0.58 + 0.09	0.45 + 0.03	0.51 + 0.03	0.58 + 0.05	0.15 \approx 0.02	0.14 \approx 0.02	0.13	0.02							
UF3	0.97 + 0.08	1.22 + 0.06	0.96 + 0.04	1.08 + 0.07	1.10 + 0.06	0.54 + 0.05	0.19 - 0.08	0.42	0.03							
UF4	0.21 \approx 0.01	0.24 + 0.02	0.23 + 0.00	0.23 + 0.01	0.19 \approx 0.00	0.22 \approx 0.00	0.23 + 0.02	0.19	0.01							
UF5	4.75 + 0.42	3.53 + 0.26	2.84 + 0.18	3.25 + 0.16	5.05 + 0.96	2.46 + 0.43	2.49 + 0.44	2.42	0.38							
UF6	4.36 + 0.63	2.28 + 0.24	1.69 + 0.13	1.99 + 0.17	5.26 + 0.59	1.34 + 0.13	1.01 + 0.25	0.81	0.19							
UF7	1.20 + 0.12	0.58 + 0.06	0.38 + 0.03	0.47 + 0.05	1.24 + 0.14	0.21 - 0.04	0.37 \approx 0.06	0.33	0.05							

In this subsection, the performance of TC-SAEA on the above test instances is compared with the aforementioned algorithms in terms of convergence and diversity, indicated by IGD values. To investigate how different the delay length τ affects the algorithm's performance, the settings $\tau = 5$ and $\tau = 10$ are used, and the mean values and the standard deviations of IGD over 20 times are collected and presented in Tables 1 and 2, where the best result of each benchmark function is highlighted. To further illustrate the advantage of the proposed algorithm, the non-dominated solution set obtained by each algorithm (in the run that achieved the medium performance out of 20 independent runs) on DTLZ2, DTLZ7 and UF3 are visualized in Figs. 5-7.

According to the results in Table 1, we can observe that TC-SAEA outperforms the compared algorithms on all test instances except on DTLZ4, UF3 and UF7, demonstrating the effectiveness of the suggested strategies for solving bi-objective optimization problems with a delayed objective. The reason for TC-SAEA being outperformed by HK-RVEA on DTLZ4 may be that the density of the points on the true PF of DTLZ4 is strongly biased, and as a result, the synthetic values \mathbf{Y}_s^a in TC-SAEA are too far away from the true objective values of f_s . Consequently, the knowledge learned from the auxiliary data set $D_a = (\mathbf{X}^a, \mathbf{Y}_s^a)$ is very limited and the risk of misleading the optimization process will be increased. It is important to observe that TC-SAEA is successful in accelerating convergence and achieving evenly distributed non-dominated

Table 2: Statistical results of the IGD values obtained by Waiting, Fast-first, BI, SI, K-RVEA, HK-RVEA, T-SAEA and TC-SAEA with $FE_s^{max} = 200$ and $\tau = 10$

Problem	Waiting		Fast-first		BI		SI		K-RVEA		HK-RVEA		T-SAEA		TC-SAEA	
	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std
DTLZ1	30.2 + 14.3		102 + 35.5		30.1 + 8.24		44.9 + 15.1		29.8 + 17.8		41.5 + 11.8		35.8 + 18.4		16.8	13.9
DTLZ1a	14.2 + 8.48		24.3 + 10.8		13.1 + 7.95		34.8 + 12.3		1.03 + 0.32		1.04 + 0.47		0.81 + 0.68		0.60	0.40
DTLZ2	0.24 + 0.05		0.96 + 0.11		0.35 + 0.04		0.39 + 0.04		0.13 + 0.06		0.08 + 0.01		0.06 \approx 0.03		0.03	0.02
DTLZ3	349 + 83.4		651 + 115		369 + 59.6		430 + 90.7		385 + 59.4		379 + 30.5		385 + 118		137	77.0
DTLZ3a	227 + 75.4		598 + 139		284 + 68.7		418 + 90.2		73.7 + 29.2		19.8 + 8.50		6.71 + 6.68		0.43	0.21
DTLZ4	0.51 + 0.32		0.75 + 0.13		0.48 + 0.07		0.66 + 0.12		0.45 - 0.15		0.17 - 0.10		0.47 \approx 0.17		0.48	0.17
DTLZ5	0.27 + 0.06		0.97 + 0.10		0.37 + 0.04		0.41 + 0.03		0.14 + 0.05		0.09 + 0.01		0.06 \approx 0.03		0.03	0.01
DTLZ6	7.31 + 0.52		8.96 + 0.19		7.71 + 0.32		8.28 + 0.16		5.14 + 0.77		4.23 + 0.35		6.49 + 0.47		3.31	0.65
DTLZ7	4.41 + 0.62		7.81 + 0.64		5.89 + 0.36		5.79 + 0.50		5.54 + 0.47		0.04 - 0.03		4.26 + 0.66		0.08	0.21
UF1	1.01 + 0.14		0.54 + 0.07		0.42 + 0.04		0.41 + 0.04		1.20 + 0.11		0.26 + 0.04		0.64 + 0.18		0.15	0.02
UF2	0.50 + 0.07		0.74 + 0.10		0.44 + 0.03		0.52 + 0.04		0.58 + 0.05		0.15 + 0.01		0.27 + 0.08		0.12	0.02
UF3	0.97 + 0.08		1.21 + 0.07		0.91 + 0.05		1.09 + 0.08		1.10 + 0.06		0.53 \approx 0.03		0.58 + 0.07		0.51	0.06
UF4	0.21 \approx 0.01		0.58 + 0.02		0.23 \approx 0.00		0.23 \approx 0.01		0.19 - 0.00		0.23 \approx 0.01		0.19 - 0.01		0.22	0.00
UF5	4.75 + 0.42		3.46 + 0.28		2.96 + 0.14		3.24 + 0.19		5.05 + 0.31		2.43 \approx 0.32		4.12 + 0.63		2.39	0.20
UF6	4.36 + 0.63		2.17 + 0.25		1.55 + 0.14		2.00 + 0.18		5.26 + 0.59		1.00 + 0.23		1.42 + 0.79		0.69	0.12
UF7	1.11 + 0.18		0.54 + 0.06		0.37 + 0.03		0.46 + 0.05		1.24 + 0.14		0.26 \approx 0.05		0.59 + 0.20		0.27	0.13

solutions on the rest problems of the DTLZ test suite. The non-dominated solutions obtained by each algorithm on DTLZ2 and DTLZ7 are presented in Fig. 5 and Fig. 6, respectively, confirming the effectiveness of the proposed instance transfer together with the co-surrogate. Similar conclusion can be drawn from the results on UF1-UF7. Figs. 7 shows the performance of each algorithm on UF3, indicating that TC-SAEA is more likely to converge to the true PF with the help of knowledge transfer.

Our observations in terms of the three delay-handling schemes (i.e. *Waiting*, *Fast-first* and *Interleaving schemes*) agree with the point made in (Allmendinger et al., 2015): *Interleaving schemes* are generally the best. We note that the surrogate based delay-handling methods, such as HK-RVEA and T-SAEA, generally show better performance than the non-surrogate based ones on the problems with a limited budget of FEs, which can be explained in the sense that the former can benefit from surrogate models, while the latter suffers a slow convergence. It is interesting to compare K-RVEA with the surrogate based methods as K-RVEA is a surrogate based *Waiting* method and HK-RVEA, T-SAEA and the proposed method are surrogate based *Interleaving* methods. We can observe that by learning from the additional knowledge obtained on f_f , one can achieve a better trade-off between the convergence and diversity on bi-objective problems with heterogeneous objectives. To further demonstrate how the learning process proposed in our work helps the optimization, we run each algorithm on DTLZ1a with $FE_s^{max} = 1000$, and the IGD values of the solution set over the generations obtained by each algorithm is plotted in Fig. 8, indicating the fast convergence of TC-SAEA.

To assess the impact of the delay length on the performance, Table 2 presents the results obtained by each algorithm when $\tau = 10$. Note that the relationship between the objective functions or the fitness landscape is unknown in our test problems. Therefore, it is unsurprising to observe that the performance on some test problems is degraded due to the increasing search bias resulting from the much more intensive search on

the fast objective, even though the evaluation budget for the non-delayed objective is increased. For example, the performance of HK-RVEA on DTLZ1, DTLZ1a, DTLZ2, DTLZ3a and DTLZ6 is impacted negatively by a larger τ . It is also noteworthy that TC-SAEA is still the most promising delay-handling method in comparison with the other algorithms in terms of diversity and convergence, which is consistent with the observations in Table 1.

We also compare TC-SAEA with Tr-SAEA (Wang et al., 2021), a recently reported algorithm for solving bi-objective problems with non-uniform evaluation times. The performance of K-RVEA on the same test instances but in the undelayed environment where all objectives are fast objectives and have the same time complexity is also included for reference to the best achievable results. The results in terms of IGD values on test instance with $\tau = 5$ and $\tau = 10$ are presented in Table 3. Although TC-SAEA and Tr-SAEA show similar performance on most test instances, it should be noted that TC-SAEA outperforms Tr-SAEA on DTLZ1 and DTLZ3. As suggested in (Deb et al., 2002), DTLZ1 and DTLZ3 can be adopted to investigate an MOEA’s ability to converge to the Pareto front. Hence, these results confirm the desirable strong convergence capability of TC-SAEA, implying potential advantages of direct knowledge transfer between different objectives.



Figure 5: The final non-dominated solutions obtained by the compared algorithms on the bi-objective DTLZ2 in the run associated with the median IGD value.

4.5 Ablation Studies

To effectively address the optimization problems with delayed objectives, we propose an instance-based transfer learning scheme in the framework of a GP-based SAEA, hoping to transfer the knowledge readily available from the fast objective function for the optimization of the slow one. The proposed TC-SAEA is composed of two main components, a co-surrogate model to capture the hidden correlation between the ob-

Table 3: Statistical results of the IGD values obtained by Tr-SAEA, TC-SAEA and Undelayed algorithm with $FE_s^{max} = 200$ and different τ

Problem	Tr-SAEA		TC-SAEA		Undelayed		Tr-SAEA		TC-SAEA		Undelayed	
	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std
	$\tau = 5$						$\tau = 10$					
DTLZ1	20.7	≈ 5.38	20.1	8.16	15.8	5.71	29.5	+ 18.3	16.8	13.9	12.2	5.52
DTLZ1a	0.21	≈ 0.07	0.36	0.04	0.32	0.08	0.25	- 0.05	0.60	0.40	0.23	0.04
DTLZ2	0.03	≈ 0.01	0.02	0.00	0.02	0.01	0.03	≈ 0.01	0.03	0.02	0.02	0.00
DTLZ3	327	+ 82.1	132	79.28	168	61.9	348	+ 41.6	137	77.0	119	24.5
DTLZ3a	3.39	≈ 1.87	2.30	0.66	13.9	2.32	4.84	+ 2.97	0.43	0.21	9.00	1.64
DTLZ4	0.16	- 0.07	0.44	0.13	0.05	0.07	0.19	- 0.11	0.48	0.17	0.02	0.00
DTLZ5	0.03	≈ 0.03	0.03	0.00	0.02	0.00	0.04	≈ 0.01	0.03	0.01	0.02	0.00
DTLZ6	0.72	- 0.09	2.62	1.95	2.52	0.04	0.74	- 0.88	3.31	0.65	1.80	0.43
DTLZ7	0.03	≈ 0.01	0.05	0.08	0.04	0.00	0.02	≈ 0.01	0.08	0.21	0.05	0.01
UF1	0.19	≈ 0.01	0.19	0.02	0.15	0.04	0.39	+ 0.10	0.15	0.02	0.15	0.02
UF2	0.12	≈ 0.01	0.13	0.02	0.09	0.02	0.14	≈ 0.01	0.12	0.02	0.08	0.01
UF3	0.49	≈ 0.01	0.42	0.03	0.44	0.02	0.50	≈ 0.03	0.51	0.06	0.42	0.02
UF4	0.22	≈ 0.00	0.19	0.01	0.12	0.01	0.22	≈ 0.00	0.22	0.00	0.10	0.00
UF5	2.43	≈ 0.28	2.42	0.38	1.55	0.31	2.81	+ 0.40	2.39	0.20	1.22	0.23
UF6	1.32	+ 0.39	0.81	0.19	0.58	0.13	1.14	+ 0.27	0.69	0.12	0.52	0.05
UF7	0.32	≈ 0.11	0.33	0.05	0.32	0.13	0.36	+ 0.07	0.27	0.13	0.28	0.20

Table 4: Statistical results of the IGD values obtained by TC-SAEAp, NT-SAEA, NS-SAEA and TC-SAEA with $FE_s^{max} = 200$ and $\tau = 5$

Problem	TC-SAEAp		NT-SAEA		NS-SAEA		TC-SAEA	
	mean	std	mean	std	mean	std	mean	std
DTLZ1	26.6	≈ 10.5	46.3	+ 12.9	30.0	+ 12.7	20.1	8.16
DTLZ1a	19.9	+ 0.19	0.87	+ 0.14	1.35	+ 0.35	0.36	0.04
DTLZ2	0.02	≈ 0.03	0.05	+ 0.01	0.03	≈ 0.01	0.02	0.00
DTLZ3	467	+ 72.9	322	+ 140	241	+ 163	132	79.28
DTLZ3a	155	+ 5.28	10.1	+ 15.0	2.59	+ 0.89	2.30	0.66
DTLZ4	0.55	+ 0.12	0.29	- 0.52	0.52	+ 0.19	0.44	0.17
DTLZ5	0.02	- 0.03	0.09	+ 0.03	0.07	+ 0.01	0.03	0.00
DTLZ6	3.76	+ 0.54	6.60	+ 7.18	3.04	+ 2.36	2.62	1.95
DTLZ7	0.02	- 0.06	1.29	+ 0.40	0.45	+ 0.29	0.05	0.08
UF1	0.79	+ 0.03	1.35	+ 0.03	0.24	+ 0.03	0.19	0.02
UF2	0.14	≈ 0.03	0.61	+ 0.05	0.20	+ 0.06	0.13	0.01
UF3	0.92	+ 0.06	0.55	+ 0.01	0.52	+ 0.02	0.42	0.03
UF4	0.22	≈ 0.00	0.19	≈ 0.01	0.22	≈ 0.02	0.19	0.01
UF5	3.24	+ 0.44	4.32	+ 1.07	2.88	+ 0.61	2.42	0.38
UF6	2.14	+ 0.13	4.13	+ 3.06	0.98	≈ 0.12	0.81	0.19
UF7	0.67	+ 0.05	0.50	+ 0.09	0.37	+ 0.09	0.33	0.05



Figure 6: The final non-dominated solutions obtained by the compared algorithms on the bi-objective DTLZ7 in the run associated with the median IGD value.

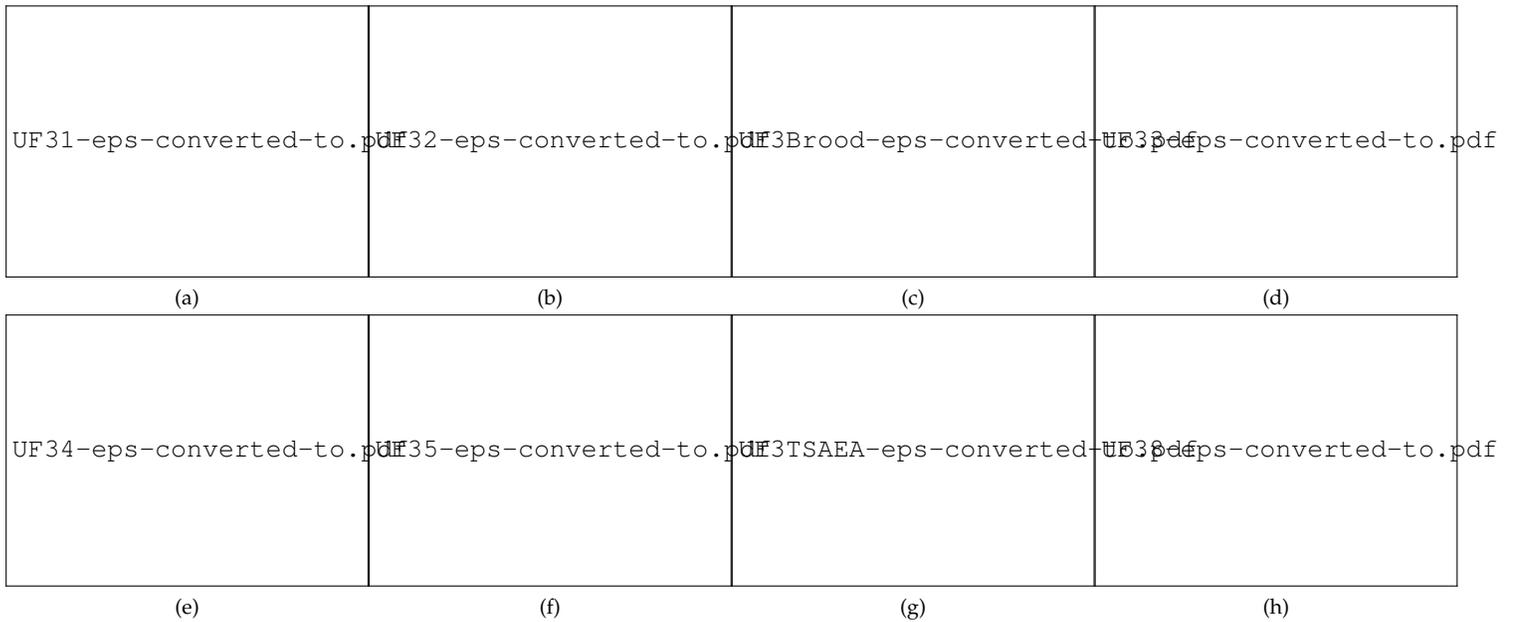


Figure 7: The final non-dominated solutions obtained by the compared algorithms on the bi-objective UF3 in the run associated with the median IGD value.

jectives and a transferable instance selection strategy to identify the useful knowledge. In the following, we compare TC-SAEA with its three variants to further investigate



Figure 8: IGD values obtained on DTLZ1a by each algorithm over the generations.

Table 5: Statistical results of the IGD values obtained by TC-SAEAp, NT-SAEA, NS-SAEA and TC-SAEA with $FE_s^{max} = 200$ and $\tau = 10$

Problem	TC-SAEAp		NT-SAEA		NS-SAEA		TC-SAEA	
	mean	std	mean	std	mean	std	mean	std
DTLZ1	34.1	+ 29.9	50.7	+ 20.4	27.9	+ 18.60	16.8	13.91
DTLZ1a	1.27	+ 1.26	0.75	+ 0.10	0.48	- 0.16	0.60	0.40
DTLZ2	0.03	\approx 0.02	0.03	\approx 0.00	0.11	+ 0.09	0.03	0.02
DTLZ3	376	+ 141	382	+ 137	336	+ 85.0	137	77.0
DTLZ3a	314.8	+ 546	11.6	+ 12.6	3.65	+ 4.08	0.43	0.21
DTLZ4	0.92	+ 0.27	0.67	+ 0.30	0.54	\approx 0.16	0.48	0.17
DTLZ5	0.02	- 0.01	0.03	\approx 0.01	0.05	+ 0.05	0.03	0.03
DTLZ6	4.41	+ 0.48	6.46	+ 1.18	3.50	\approx 0.62	3.51	0.65
DTLZ7	0.16	+ 0.00	0.13	+ 0.10	0.45	+ 0.00	0.08	0.29
UF1	0.39	+ 0.12	0.44	+ 0.26	0.24	+ 0.03	0.15	0.02
UF2	0.15	+ 0.03	0.27	+ 0.01	0.23	+ 0.07	0.12	0.02
UF3	0.68	+ 0.27	0.59	+ 0.08	0.55	+ 0.04	0.51	0.06
UF4	0.22	\approx 0.01	0.19	- 0.01	0.23	\approx 0.01	0.22	0.01
UF5	2.76	\approx 0.64	3.78	+ 0.86	2.60	\approx 0.48	2.39	0.28
UF6	1.81	+ 0.83	2.31	+ 0.06	1.00	+ 0.26	0.69	0.05
UF7	0.63	+ 0.12	0.64	+ 0.13	0.35	+ 0.02	0.27	0.13

the performance impact of the suggested strategies.

- A GP-based SAEA without the transfer learning scheme (NT-SAEA): In order to demonstrate the effectiveness of the transfer learning scheme, an SAEA without the TL scheme is introduced. In NT-SAEA, the GP model is built separately for each objective with different available training data sets. Due to the different evaluation times, the number of new samples for updating each surrogate is also different.
- A GP-based SAEA with a polynomial regression as the co-surrogate (TC-SAEAp): Instead of the GP, a polynomial regression is used as the co-surrogate to model the correlation between the objectives. The proposed TL scheme is adopted in TC-SAEAp.

- A GP-based SAEA without the transferable instance selection (NS-SAEA): In this variant, only the co-surrogate component in the proposed TL scheme is adopted in NS-SAEA. Due to the lack of selection, all the auxiliary data $D_a = (\mathbf{X}^a, \mathbf{Y}_s^a)$ will be added to the training data set of the slow objective.

Firstly, it is observed that TC-SAEA achieves the better performance than TC-SAEAp in terms of convergence and diversity on both test instances with $\tau = 5$ and $\tau = 10$. Recall that the difference between TC-SAEA and TC-SAEAp lies in their co-surrogate model; therefore, this observation confirms that the GP co-surrogate model can capture the hidden correlation between objectives more accurately and effectively than the polynomial regression model. Secondly, it is clear to see from Tables 4 and 5 that TC-SAEA has superior performance to NT-SAEA on most test instances, while NT-SAEA yields the better results on DTLZ4 when $\tau = 5$ and on UF4 when $\tau = 10$. Compared with NT-SAEA, TC-SAEA is able to learn some useful knowledge from the auxiliary data set, when the observed data for f_s is insufficient to train a good model. The comparison between NT-SAEA and TC-SAEA confirms that the performance of the surrogate model GP_s for f_s can benefit from the transferable data set D_t . Thirdly, it should be pointed out that NS-SAEA can achieve good performance on DTLZ1a and DTLZ6 only when $\tau = 10$, which means that transferable instance selection is helpful in most cases. Despite the use of transfer learning in NS-SAEA, it fails to converge towards a set of acceptable solutions. This observation clearly indicates the benefit of the proposed transferable instance selection. Note that the effectiveness of the transfer learning scheme heavily relies on the quality of the auxiliary data \mathbf{Y}_s^a generated by transfer learning. Compared with the carefully selected transferable data, applying the auxiliary data directly to train GP_s without filtering out unreliable synthetic data may be more likely to lead to negative transfer. TC-SAEA adopts an instance selection method according to the uncertainty level provided by GP_s in order to mitigate possible negative transfer, which is demonstrated to be effective by the comparative results between TC-SAEA and NS-SAEA. In summary, it is evident that TC-SAEA is better suited for solving bi-objective optimization problems for objectives with distinct evaluation times in comparison with NT-SAEA and NS-SAEA.

4.6 Impact of the Correlation between Objectives

The rationale behind the proposed transfer learning scheme is that there will be a certain degree of correlation between the objectives of the Pareto optimal solutions, although such a correlation may be weak or not exist in most part of the search space. Here, we further investigate the impact of the correlation between objectives on the performance of TC-SAEA using the cm-OneMax test problem, since the strength of the correlation between its objectives is controllable. The IGD values of the obtained solutions by each algorithm under comparison on the cm-OneMax problem with $corr = -1, -0.75, -0.5, -0.25, 0, 0.25, 0.5, 0.75, 1$ are presented in Table 6.

These results indicate that the proposed TC-SAEA exhibits the best performance on the cm-OneMax problem when there is a positive or negative correlation between the objectives, and HK-RVEA is the second best. This observation demonstrates the ability of the proposed TC-SAEA for transferring useful knowledge between the objectives. Interestingly, if we take into account the results obtained on the DTLZ and UF test suites, where the correlation between the objectives on the PF is defined by a function, we can conclude that TC-SAEA generally has achieved good performance, confirming its ability to utilize the knowledge obtained from f_f to assist the optimization of f_s . The enhanced performance of TC-SAEA can be attributed to capability of the

Table 6: The IGD values obtained by Waiting, Fast-first, Brood interleaving, Speculative interleaving, K-RVEA, HK-RVEA, T-SAEA, and TC-SAEA on the cm-OneMax problem with $FE_s^{max} = 200$

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

co-surrogate of capturing the relationship between the objectives and the instance selection strategy to reduce the negative transfer. Secondly, HK-RVEA has also obtained good performance on the cm-OneMax with $corr = -0.75, -0.25, 0, 0.25, 1$, which is consistent with the results in (Chugh et al., 2018b). Thirdly, it is observed that *Waiting* shows stable and good performance compared with the delay-handling methods without surrogates, particularly in case there is a negative correlation between the objectives. However, accounting for expensive MOPs with a limited amount of evaluation budgets, surrogate assisted methods, such as K-RVEA, HK-RVEA and TC-SAEA, are more promising compared with methods without surrogates.

4.7 Effects of the Co-surrogate

To gain deeper insights into the relationship between the performance of TC-SAEA and the approximation quality of the co-surrogate GP_c , we examine the mean square error (MSE) of the estimated difference between f_s and f_f on the additional solutions \mathbf{X}^a provided by GP_c . Specifically, we run TC-SAEA on each bi-objective DTLZ test instance and calculate the MSE of the estimation provided by the co-surrogate on \mathbf{X}^a at each generation. Ten independent runs, each with a maximum of 200 expensive function evaluations, are performed and the mean and variance of the MSE of the co-surrogate are calculated and plotted in Fig. S1 in the Supplementary material. We also plot the achieved non-dominated solutions for each instance in the run associated with the median IGD value in Fig. S2. From these results, we can observe that the approximation error of the co-surrogate converges over the generations if the obtained non-dominated solutions are close to the Pareto front, such as on DTLZ2, DTLZ3a, DTLZ4, DTLZ5 and

DTLZ7. By contrast, the approximation error of the co-surrogate strongly oscillates on DTLZ1, DTLZ2, and DTLZ6 because the obtained non-dominated solutions are still far away from the true Pareto front, where no clear relationship between f_s and f_f exists. These results agree well with our hypothesis discussed in Section 3.2.

4.8 Parameter Sensitivity Analysis

The sensitivity analysis of the performance to the number of new samples u for updating GPs in TC-SAEA is presented in the Supplementary material. Fig. S3 in the Supplementary material shows the boxplots in terms of IGD over different u values on the DTLZ test suite. We can see that TC-SAEA with different u exhibits distinct performance on each test instance, indicating that the best setting of u may be problem-dependent. For example, TC-SAEA shows similar performance on DTLZ2 and DTLZ3a as u changes, which is not the case on DTLZ3 and DTLZ4. Interestingly, setting too small or too large values to u , e.g., $u = 1$ or $u = 10$, will negatively impact the performance of TC-SAEA on most test instances. By contrast, TC-SAEA with $u = 3, 5$ show competitive results. Therefore, $u = 3$ is considered to be appropriate for our algorithm for handling bi-objective optimization problems with a delayed objective.

5 Conclusion

In this paper we focus on the bi-objective optimization problems where the evaluation of one objective takes longer time than another one, and propose a transfer learning scheme within a GP-assisted multi-objective evolutionary algorithm. The transfer learning scheme includes two key components, a co-surrogate model to indirectly relate the fast objective function to the slow objective function and a transferable instance selection to identify the useful knowledge in the auxiliary dataset. The co-surrogate model is used to generate synthetic data for the expensive objective function by transferring knowledge from additional function evaluations of the fast objective. To reduce the risk of negative transfer, the confidence bound of the surrogate for the slow objective function is used to filter out unreliable synthetic data.

The proposed algorithm is tested on sets of widely used benchmark problems for different delay lengths. Our experimental results demonstrate that the proposed algorithm achieves significantly better performance than the state-of-the-art delay-handling methods on most test instances studied in this work. Comparisons are also carried out to investigate the effectiveness of the individual mechanisms of the proposed algorithm and the empirical results equally confirm that both the co-surrogate and the transferable instance selection are indispensable for the good performance of the proposed algorithm.

Despite the encouraging results, research on MOPs with delayed objectives is still in its infancy stage and many challenges remain to be addressed in the future. For example, it will become more challenging to achieve efficient knowledge transfer when the number of objectives increases. In addition, it is of interest to investigate the most effective way of making use of the additional function evaluations of the cheap objective. Finally, an optimal resource allocation between fast and slow objective is conceivable to achieve the best performance for a given total computational budget.

Acknowledgements

This work was supported in part by a PhD studentship funded by the Honda Research Institute Europe, Germany. YJ is supported in part by an Alexander von Humboldt Professorship endowed by the German Federal Ministry of Education and Research.

References

- Abraham, A. and Jain, L. (2005). Evolutionary multiobjective optimization. In *Evolutionary Multiobjective Optimization*, pages 1–6. Springer.
- Allmendinger, R., Emmerich, M. T., Hakanen, J., Jin, Y., and Rigoni, E. (2017). Surrogate-assisted multicriteria optimization: Complexities, prospective solutions, and business case. *Journal of Multi-Criteria Decision Analysis*, 24(1-2):5–24.
- Allmendinger, R., Handl, J., and Knowles, J. (2015). Multiobjective optimization: When objectives exhibit non-uniform latencies. *European Journal of Operational Research*, 243(2):497–513.
- Allmendinger, R. and Knowles, J. (2013). ‘hang on a minute’: Investigations on the effects of delayed objective functions in multiobjective optimization. In *International Conference on Evolutionary Multi-Criterion Optimization*, pages 6–20. Springer.
- Bosman, P. A. and Thierens, D. (2003). The balance between proximity and diversity in multiobjective evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 7(2):174–188.
- Bourinet, J.-M. (2016). Rare-event probability estimation with adaptive support vector regression surrogates. *Reliability Engineering & System Safety*, 150:210–221.
- Chattopadhyay, R., Sun, Q., Fan, W., Davidson, I., Panchanathan, S., and Ye, J. (2012). Multi-source domain adaptation and its application to early detection of fatigue. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 6(4):1–26.
- Cheng, R., Jin, Y., Olhofer, M., and Sendhoff, B. (2016). A reference vector guided evolutionary algorithm for many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 20(5):773–791.
- Chugh, T., Allmendinger, R., Ojalehto, V., and Miettinen, K. (2018a). Surrogate-assisted evolutionary biobjective optimization for objectives with non-uniform latencies. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 609–616. ACM.
- Chugh, T., Jin, Y., Miettinen, K., Hakanen, J., and Sindhya, K. (2018b). A surrogate-assisted reference vector guided evolutionary algorithm for computationally expensive many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 22(1):129–142.
- Coello, C. A. C., Lamont, G. B., Van Veldhuizen, D. A., et al. (2007). *Evolutionary algorithms for solving multi-objective problems*, volume 5. Springer.
- Cox, D. D. and John, S. (1992). A statistical method for global optimization. In *Proceedings 1992 IEEE International Conference on Systems, Man, and Cybernetics*, pages 1241–1246. IEEE.
- Dai, W., Yang, Q., Xue, G.-R., and Yu, Y. (2007). Boosting for transfer learning. In *Proceedings of the 24th International Conference on Machine Learning*, pages 193–200.
- Deb, K. (1999). Multi-objective genetic algorithms: Problem difficulties and construction of test problems. *Evolutionary Computation*, 7(3):205–230.
- Deb, K., Thiele, L., Laumanns, M., and Zitzler, E. (2002). Scalable multi-objective optimization test problems. In *Proceedings of the 2002 Congress on Evolutionary Computation. CEC’02 (Cat. No. 02TH8600)*, volume 1, pages 825–830. IEEE.
- Duan, L., Xu, D., and Tsang, I. (2012). Learning with augmented features for heterogeneous domain adaptation. *arXiv preprint arXiv:1206.4660*.
- Gong, B., Shi, Y., Sha, F., and Grauman, K. (2012). Geodesic flow kernel for unsupervised domain adaptation. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2066–2073. IEEE.
- Hennig, P. and Schuler, C. J. (2012). Entropy search for information-efficient global optimization. *Journal of Machine Learning Research*, 13(Jun):1809–1837.

- Hernández-Lobato, J. M., Hoffman, M. W., and Ghahramani, Z. (2014). Predictive entropy search for efficient global optimization of black-box functions. In *Advances in Neural Information Processing Systems*, pages 918–926.
- Huang, S., Zhong, J., and Yu, W. (2019). Surrogate-assisted evolutionary framework with adaptive knowledge transfer for multi-task optimization. *IEEE Transactions on Emerging Topics in Computing*.
- Jin, Y. (2011). Surrogate-assisted evolutionary computation: Recent advances and future challenges. *Swarm and Evolutionary Computation*, 1(2):61–70.
- Jin, Y., Olhofer, M., and Sendhoff, B. (2000). On evolutionary optimization with approximate fitness functions. In *Proceedings of the 2nd Annual Conference on Genetic and Evolutionary Computation*, pages 786–793.
- Jin, Y. and Sendhoff, B. (2002). Fitness approximation in evolutionary computation - A survey. In *Proceedings of the 4th Annual Conference on Genetic and Evolutionary Computation Conference*, pages 1105–1112. ACM.
- Jin, Y., Wang, H., Chugh, T., Guo, D., and Miettinen, K. (2019). Data-driven evolutionary optimization: An overview and case studies. *IEEE Transactions on Evolutionary Computation*, 23(3):442–458.
- Jones, D. R., Schonlau, M., and Welch, W. J. (1998). Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4):455–492.
- Knowles, J. (2006). ParEGO: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation*, 10(1):50–66.
- Kushner, H. J. (1964). A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise. *Journal of Basic Engineering*, 86(1):97–106.
- Li, F., Cai, X., Gao, L., and Shen, W. (2020). A surrogate-assisted multiswarm optimization algorithm for high-dimensional computationally expensive problems. *IEEE Transactions on Cybernetics*, 51(3):1390–1402.
- Liu, J., Han, Z., and Song, W. (2012). Comparison of infill sampling criteria in kriging-based aerodynamic optimization. In *28th Congress of the International Council of the Aeronautical Sciences*, pages 23–28.
- Lophaven, S. N., Nielsen, H. B., Søndergaard, J., et al. (2002). *DACE: a Matlab Kriging toolbox*, volume 2. Citeseer.
- Matheron, G. (1963). Principles of geostatistics. *Economic Geology*, 58(8):1246–1266.
- Min, A. T. W., Ong, Y.-S., Gupta, A., and Goh, C.-K. (2017). Multiproblem surrogates: transfer evolutionary multiobjective optimization of computationally expensive problems. *IEEE Transactions on Evolutionary Computation*, 23(1):15–28.
- Naujoks, B., Beume, N., and Emmerich, M. (2005). Metamodel-assisted SMS-EMOA applied to airfoil optimization tasks. In *Proceedings EUROGEN*, volume 5.
- Noè, U. and Husmeier, D. (2018). On a new improvement-based acquisition function for bayesian optimization. *arXiv preprint arXiv:1808.06918*.
- Pan, S. J. and Yang, Q. (2009). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359.
- Shahriari, B., Swersky, K., Wang, Z., Adams, R. P., and De Freitas, N. (2015). Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175.

- Sun, C., Jin, Y., Cheng, R., Ding, J., and Zeng, J. (2017). Surrogate-assisted cooperative swarm optimization of high-dimensional expensive problems. *IEEE Transactions on Evolutionary Computation*, 21(4):644–660.
- Tian, Y., Cheng, R., Zhang, X., and Jin, Y. (2017). PlatEMO: A MATLAB platform for evolutionary multi-objective optimization [educational forum]. *IEEE Computational Intelligence Magazine*, 12(4):73–87.
- Wang, G. G. and Shan, S. (2006). Review of Metamodeling Techniques in Support of Engineering Design Optimization. *Journal of Mechanical Design*, 129(4):370–380.
- Wang, H. and Jin, Y. (2020). A random forest assisted evolutionary algorithm for data-driven constrained multi-objective combinatorial optimization of trauma systems. *IEEE Transactions on Cybernetics*, 50(2):536–549.
- Wang, X., Jin, Y., Schmitt, S., and Olhofer, M. (2020a). An adaptive bayesian approach to surrogate-assisted evolutionary multi-objective optimization. *Information Sciences*, 519:317–331.
- Wang, X., Jin, Y., Schmitt, S., and Olhofer, M. (2020b). Transfer learning for Gaussian process assisted evolutionary bi-objective optimization for objectives with different evaluation times. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference*, pages 587–594.
- Wang, X., Jin, Y., Schmitt, S., Olhofer, M., and Allmendinger, R. (2021). Transfer learning based surrogate assisted evolutionary bi-objective optimization for objectives with different evaluation times. *Knowledge-Based Systems*, page 107190.
- While, L., Hingston, P., Barone, L., and Huband, S. (2006). A faster algorithm for calculating hypervolume. *IEEE Transactions on Evolutionary Computation*, 10(1):29–38.
- Yang, C., Ding, J., Jin, Y., and Chai, T. (2019). Off-line data-driven multi-objective optimization: Knowledge transfer between surrogates and generation of final solutions. *IEEE Transactions on Evolutionary Computation*.
- Yang, L., Jing, L., Yu, J., and Ng, M. K. (2015). Learning transferred weights from co-occurrence data for heterogeneous transfer learning. *IEEE Transactions on Neural Networks and Learning Systems*, 27(11):2187–2200.
- Zhang, Q., Liu, W., Tsang, E., and Virginas, B. (2009). Expensive multiobjective optimization by MOEA/D with Gaussian process model. *IEEE Transactions on Evolutionary Computation*, 14(3):456–474.
- Zhang, Q., Zhou, A., Zhao, S., Suganthan, P. N., Liu, W., Tiwari, S., et al. (2008). Multiobjective optimization test instances for the CEC 2009 special session and competition. Technical report, University of Essex, Colchester, UK and Nanyang technological University.
- Zhou, A., Qu, B.-Y., Li, H., Zhao, S.-Z., Suganthan, P. N., and Zhang, Q. (2011). Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation*, 1(1):32–49.
- Zhou, Z., Ong, Y. S., Nguyen, M. H., and Lim, D. (2005). A study on polynomial regression and gaussian process global surrogate model in hierarchical surrogate-assisted evolutionary algorithm. In *2005 IEEE Congress on Evolutionary Computation*, volume 3, pages 2832–2839. IEEE.