Transfer Stacking from Low- to High-Fidelity: A Surrogate-Assisted Bi-Fidelity Evolutionary Algorithm

Handing Wang[a], Yaochu Jin[b,c,*], Cuie Yang[c], Licheng Jiao[a]

[a]*School of Artificial Intelligence, Xidian University, Xi'an 710071, China*
[b]*Department of Computer Science, University of Surrey, Guildford GU2 7XH, U.K.*
[c]*State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University, Shenyang 110819, China*

Abstract

Optimization of many real-world optimization problems relies on numerical simulations for function evaluations. In some cases, both high- and low-fidelity simulations are available, where the high fidelity evaluation is accurate but time-consuming, whereas the low-fidelity evaluation is less accurate but computationally cheap. To find an acceptable optimum within a limited budget, it is economical for evolutionary algorithms to use both high- and low-fidelity evaluations in a single optimization search. This paper proposes a novel surrogate-assisted evolutionary algorithm using the transfer stacking technique for bi-fidelity optimization. To this end, a radial basis function network is firstly built to approximate the high-fidelity fitness function as additional low-fidelity evaluation, then a surrogate model transferring the original and additional low-fidelity evaluations to the expensive high-fidelity evaluation is adapted to guide the search. The simulation results on a series of bi-fidelity optimization benchmark problems with resolution, stochastic, and instability errors and a benficiation processes optimization problem show that the proposed algorithm is both effective and efficient for solving bi-fidelity optimization problems, when their low-fidelity evaluations have resolution and stochastic errors.

*Corresponding author.
Email addresses: hdwang@xidian.edu.cn (Handing Wang), yaochu.jin@surrey.ac.uk (Yaochu Jin), cuieyang@outlook.com (Cuie Yang), lchjiao@mail.xidian.edu.cn (Licheng Jiao)

Keywords: Bi-fidelity optimization, transfer stacking, surrogate, evolutionary computation, radial basis function network.

1. Introduction

Evolutionary algorithms (EAs) have been applied to many engineering optimization problems [1, 2], where function evaluations can be calculated from numerical simulations. Usually, those simulations of high fidelity levels are typically time-consuming, which make EAs computationally expensive [3]. In fact, the complexity of simulations can be controlled, yet at the cost of degenerating approximation accuracy [4, 5]. For example, a single 3D computational fluid dynamics (CFD) simulation with a high-fidelity level requires up to several hours of running time, but the simulation can be accelerated by using a 2D CFD simulation with a low-fidelity level instead [6, 7]. The trade-off between the fidelity and complexity of simulations creates a dilemma, in which EAs using simulations of one single fidelity level cannot be accurate as well as efficient [8]. Therefore, employing simulations of different available fidelity levels in one evolutionary search is a solution to balance the optimization performance and efficiency [9], which is known as multi-fidelity optimization or variable fidelity optimization [10].

The foundation of multi-fidelity EAs lays in the correlation and complexity differences between high- and low-fidelity simulations [11]. During the optimization process, low-fidelity simulations can be frequently adopted to roughly approximate the fitness landscape, due to their relatively low computational costs. Based on the general fitness landscape from low-fidelity simulations, a small number of high-fidelity simulations are used to refine the search for precisely exploiting the optimum [10]. Thus, the main task of multi-fidelity EAs is to adjust the fidelity [5] for either different generations [12] or individuals [13]. The simplest implementation of generation-based fidelity adjustment strategy is re-evaluating the population by high-fidelity simulations at a certain frequency, which has been applied to a particle swarm optimization algorithm in [5]. Com-

pared with generation-based fidelity adjustment strategies, individual-based fidelity adjustment strategies are flexible, where a small number of solutions are pre-selected based on the low-fidelity simulations and re-evaluated using the high-fidelity simulations. Such pre-selection can be based on the evaluation from the low-fidelity simulations [14], local correlation between simulations with different fidelity levels [11], or comparison inconsistency [4].

Bi-fidelity optimization applications are special multi-fidelity optimization problems, in which only two fidelity simulations (high- and low-fidelity levels) are available. However, due to its simple formulation, bi-fidelity optimization is a popular topic in the community of engineering design [13, 15, 16, 17]. Those mentioned fidelity adjustment techniques can be applied to bi-fidelity optimization problems to save computation cost of the high-fidelity evaluations, but the cost of the low-fidelity evaluations are not zero, which makes fidelity adjustment-based algorithms not efficient enough. To further reduce the cost of bi-fidelity EAs, surrogate models [18] that approximate the high- or low-fidelity fitness evaluations can be used [15, 19, 20]. For example, Kriging model [21], radial basis function network (RBFN) [22], polynomial regression [17], and support vector machine [23] have been employed to replace either low- or high-fidelity evaluation in bi-fidelity EAs.

However, most existing surrogate-assisted bi-fidelity EAs use both fidelity levels separately except a few algorithms using co-Kriging models [24], where the low-fidelity fitness evaluations are usually used together with the high-fidelity ones. Co-Kriging models predict the high-fidelity fitness evaluations with help of the low-fidelity fitness evaluations [25]. Thus, based on the correlation and difference between low- and high-fidelity fitness evaluations, the co-Kriging model transfers knowledge from the low-fidelity model to the high-fidelity model [26, 27, 28, 29]. It is known that the complexity of a Kriging model is $O(n^3)$ [30], where the computational time dramatically increases with the number of training data. With a large number of low-fidelity fitness evaluations as partial training data, co-Kriging models take even longer computational time than Kriging models. Furthermore, the hyper-parameter estimation in co-Kriging is

sensitive to the training dataset [31].

Inspired by co-Kriging models, bi-fidelity optimization problems can be solved by transfer learning [32, 33], where the optimization of the low-fidelity fitness is the source task while the optimization of the high-fidelity fitness is the target task. Transfer learning [34] has been applied to EAs [35], such as robotic tasks [36], genetic programming [37, 38], Bayesian optimization [39], single-objective optimization [40, 41, 42, 35], multi-objective optimization [43, 44, 45], and dynamic optimization [46, 47, 48]. In this work, we use a surrogate model transferring knowledge from low- to high-fidelity fitness evaluations in the evolutionary optimization process. The proposed algorithm, named transfer surrogate-assisted bi-fidelity evolutionary algorithm (TSA-BFEA), is applied to bi-fidelity optimization problems which are the formulation of many simulation-driven optimization applications [49].

The rest of the paper is organized as follows. Section 2 presents the mathematical description of bi-fidelity optimization problems and a transfer learning technique (transfer stacking). Section 3 details the proposed algorithm TSA-BFEA, especially for the transfer surrogate model building and infill sampling strategy. To analyze the performance of the proposed algorithm, we conduct the comparative experiments on a series of new multi-fidelity benchmarks (MFB problems) [5] in Section 4. Further, we apply TSA-BFEA to a benficiation processes optimization problem with double fidelity levels in Section 5. Finally, Section 6 concludes the paper.

2. Preliminaries

2.1. Bi-Fidelity Optimization Problems

A simulation driven optimization problem that has two-level fidelity simulations can be modeled as a bi-fidelity optimization problem according to the definition in [5]:

$$\begin{aligned} &\min f_h(\mathbf{x}) = f_l(\mathbf{x}) - e(\mathbf{x}) \\ &s.t.\ \mathbf{x} \in \Omega \end{aligned} \tag{1}$$

where $\mathbf{x}$ is a d-dimensional decision variable in its feasible region Ω, $f_l(\mathbf{x})$ and $f_h(\mathbf{x})$ are the low- and high-fidelity fitness evaluations of $\mathbf{x}$ respectively, and $e(\mathbf{x})$ is the error of $f_l(\mathbf{x})$ to $f_h(\mathbf{x})$. The difficulty of the optimization of Equation (1) comes from the high computational cost of $f_h(\mathbf{x})$, which makes EAs unaffordable due to a large number of high-fidelity fitness evaluations. Although low-fidelity fitness evaluations are of low computation cost and can be repeatedly used, EAs may be misled by $f_l(\mathbf{x})$ due to the error $e(\mathbf{x})$. Therefore, the main challenge of bi-fidelity optimization is how to efficiently make use of both low- and high-fidelity fitness evaluations to find the optimum.

2.2. Transfer Stacking

Unlike traditional machine learning, transfer learning [32] uses source data D_S, source task T_S, target data D_T, and target task T_T to improve the accuracy of T_T. If the learning tasks are to obtain the regression functions $\mathbf{y} = h_S(\mathbf{x})$ and $\mathbf{y} = h_T(\mathbf{x})$, D_S with n_S instances and D_T with n_T instances can be denoted as $\{(\mathbf{x}_S^1, \mathbf{y}_S^1), ..., (\mathbf{x}_S^{n_S}, \mathbf{y}_S^{n_S})\}$ and $\{(\mathbf{x}_T^1, \mathbf{y}_T^1), ..., (\mathbf{x}_T^{n_T}, \mathbf{y}_T^{n_T})\}$ respectively. Generally speaking, the source task is different from the target task ($T_S \neq T_T$) and the source data is easier to be obtained than the target data, which makes n_S much larger than n_T.

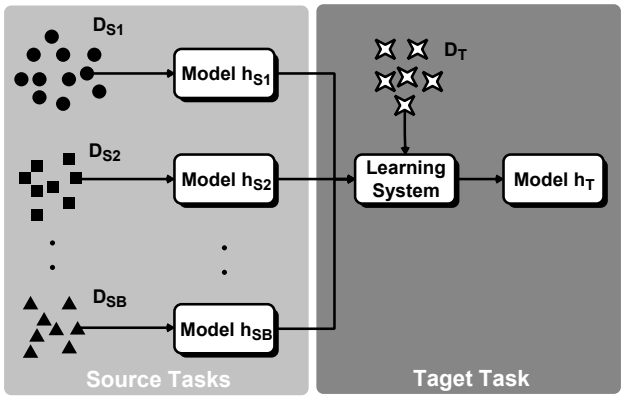

Figure 1: A diagram of Transfer Stacking

Transfer stacking [50] is a popular transfer learning algorithm for regres-

sion. As shown in Fig. 1, transfer stacking uses B trained regression models h_{S1},h_{S2},...,h_{SB} from B pieces of source data D_{S1},D_{S2},...,D_{SB} to learn the target model h_T, where h_T is a combination of source models as shown in Equation (2).

$$h_T(\mathbf{x}) = \sum_{i=1}^{B} a_i h_{Si}(\mathbf{x}) + a_{B+1} \tag{2}$$

Then, $\mathbf{a} = \{a_1, a_2, ..., a_{B+1}\}$ where a_1,a_2,...,a_B are weights of source models, and a_{B+1} is a bias, are learned by minimizing the mean squared error (MSE) of D_T in Equation (3).

$$\begin{aligned} MSE(\mathbf{a}) &= \frac{1}{n_T} \sum_{j}^{n_T} \left(h_T(\mathbf{x}_T^j) - \mathbf{y}_T^j \right)^2 \\ &= \frac{1}{n_T} \sum_{j}^{n_T} \left(\sum_{i=1}^{B} a_i h_{Si}(\mathbf{x}_T^j) + a_{B+1} - \mathbf{y}_T^j \right)^2 \end{aligned} \tag{3}$$

3. Surrogate-Assisted Bi-Fidelity Evolutionary Algorithm Using Transfer Stacking

Inspired by the transfer stacking algorithm [50], we find that both low-fidelity fitness evaluations and surrogate models can provide a large number of source data to assist the target task of locally approximating high-fidelity fitness function in each generation of EAs. Therefore, we propose a new surrogate-assisted bi-fidelity EA using a surrogate model transferring the knowledge from the low-fidelity fitness functions to the high-fidelity fitness function.

3.1. Framework

Both the evaluations using the low-fidelity fitness functions and surrogate models are computationally cheap and correlated to the high-fidelity fitness function, but neither is accurate enough to find a satisfactory solution of a bi-fidelity optimization problem. In the proposed algorithm, we use both cheap fitness evaluations to approximate the high-fidelity fitness evaluations via transfer learning, because a bi-fidelity optimization problem can be viewed as a transfer learning problem below:

- **Source task 1**: Find the optimum of an RBF model $\hat{f}_{RBF}(\mathbf{x})$ which is built from the source data D_S to approximate the high-fidelity fitness function $f_h(\mathbf{x})$.
- **Source task 2**: Find the optimum of the low-fidelity fitness function $f_l(\mathbf{x})$.
- **Target task**: Find the optimum of the high-fidelity fitness function $f_h(\mathbf{x})$.

It is clear that both source tasks can be easily achieved but the target task is hard. However, due to the correlation among the above tasks, the expensive target task can be approximated from both cheap source tasks, where the transfer stacking algorithm [50] can be applied. Therefore, in the proposed algorithm, TSA-BFEA, we employ a new surrogate model $\hat{f}_T(\mathbf{x})$ (termed transfer surrogate model in the following text) to replace the high-fidelity fitness function in the optimization process as shown in Fig. 2.

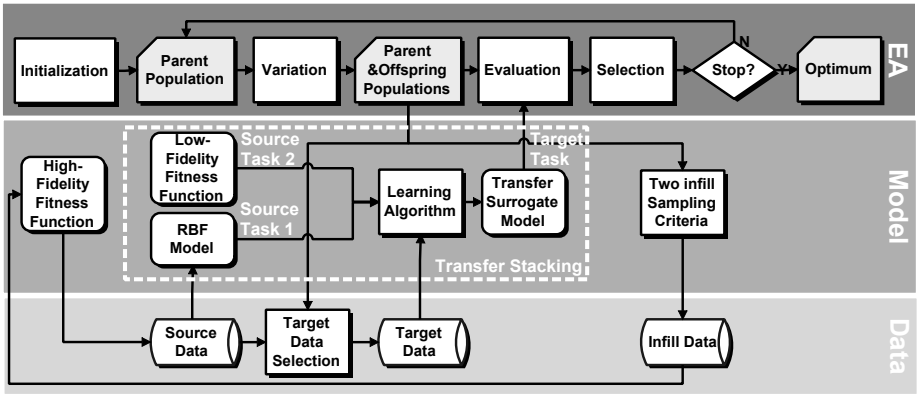

Figure 2: A diagram of the proposed framework.

As mentioned in [51], three components in simulation-driven EAs should be carefully considered: an EA as the optimizer, surrogate models as cheap evaluations, and data from simulations as the training data for building surrogate models. Fig. 2 represents the proposed algorithm into three different levels (EA, model, and data).

The main optimization procedure described in Fig. 2 is a flow-chart of EAs with initialization, variation (crossover, mutation, local search, etc.), evaluation, and selection. However, TSA-BFEA differs from the existing surrogate-assisted EAs in the model level, where a transfer surrogate model learned from both low- and high-fidelity fitness functions is used as the fitness function. To build such a transfer surrogate model for evaluations, two source tasks, an RBF model $\hat{f}_{RBF}(\mathbf{x})$ based on the source data D_S and the low-fidelity fitness function $f_l(\mathbf{x})$, are combined to locally approximate to the high-fidelity fitness function $f_h(\mathbf{x})$ using the transfer stacking algorithm, where the target data D_T are the subset of D_S close to the current population. Also, in each generation of TSA-BFEA, the population is evaluated by two infill sampling criteria to select new source data for re-evaluating via the high-fidelity fitness function.

3.2. Surrogate Model Construction Using Transfer Stacking

As shown in Fig. 2, the target task in the proposed algorithm is to locally approximate the high-fidelity fitness function $f_h(\mathbf{x})$ by transferring two different source tasks. One source task is an RBF model $\hat{f}_{RBF}(\mathbf{x})$ trained to approximate $f_h(\mathbf{x})$ using the source data D_S, the other source task is the original low-fidelity fitness function $f_l(\mathbf{x})$.

For the first source task, the RBF model is built based on d Gaussian radial basis functions. To calculate the hyper-parameters of the RBF model using the source data D_S with N data points, the k-means clustering algorithm is adopted to obtain those d centers of radial basis functions, the maximum distance between the centers are set as the widths, and the pseudo-inverse method [52] is used to assign the weights of d Gaussian radial basis functions. For the second source task, the evaluations using $f_l(\mathbf{x})$ are easily obtained, which can be directly used.

In each generation, the target task in the proposed algorithm aims to enhance the accuracy of the surrogate model in the area of the current population. Therefore, the target data D_T is the subset of training data in that local area. Using D_T, a transfer surrogate model $\hat{f}_T(\mathbf{x})$ from $f_l(\mathbf{x})$ and $\hat{f}_{RBF}(\mathbf{x})$ can be

built via the transfer stacking algorithm [50] as shown in Equation (4):

$$\hat{f}_T(\mathbf{x}) = a_1 f_l(\mathbf{x}) + a_2 \hat{f}_{RBF}(\mathbf{x}), \tag{4}$$

which is different from Equation (2). In the late stage of the optimization process, the target data D_T will concentrate in a small local area. In that case, if we use Equation (2) to build the transfer surrogate model $\hat{f}_T(\mathbf{x})$, it is easy to obtain a model highly correlated to a_{B+1}. Such model is almost a constant and cannot guide the search. That is the reason why the bias (a_{B+1}) is deleted from Equation (4). The target data D_T representing the local area that the evolutionary search focuses on is employed to learn the weight $\{a_1, a_2\}$. As shown in Algorithm 1, D_T is collected from D_S. The target data selection criterion is that one data point in D_S is similar to any solution in the current population P^t. Therefore, in the t-th generation, D_T^t is composed of the data points in D_S which are the nearest neighbors to P^t. Thus, D_T^t with n_T data points (denoted as $\{(\mathbf{x}_T^1, \mathbf{y}_T^1), ..., (\mathbf{x}_T^{n_T}, \mathbf{y}_T^{n_T})\}$) is close to the current population. The least square method is employed to obtain a weight $\{a_1, a_2\}$ by minimizing MSE of $\hat{f}_T(\mathbf{x})$ for D_T^t, as shown in Equation (5). Thus, the transfer surrogate model $\hat{f}_T(\mathbf{x})$ can adaptively refine the approximation accuracy in the area of the current population.

$$MSE(a_1, a_2) = \frac{1}{n_T} \sum_{j}^{n_T} \left(a_1 f_l(\mathbf{x}_T^j) + a_2 \hat{f}_{RBF}(\mathbf{x}_T^j) - \mathbf{y}_T^j \right)^2 \tag{5}$$

3.3. Two Infill Sampling Criteria

The approximation error of the transfer surrogate model $\hat{f}_T(\mathbf{x})$ is inevitable, which might misguide the search of the proposed algorithm. Infilling a few new samples to the training data is a common way to efficiently improve the quality of surrogate models [18]. As suggested in [53, 54], promising and uncertain solutions, as infilling samples, are beneficial to the optimization process in surrogate-assisted evolutionary algorithms (SAEAs). Promising solutions can enhance the accuracy of the surrogate model in the promising area, and uncertain solutions can help the algorithm to explore the area with insufficient

Algorithm 1 Pseudo code of data selection for target data D_T^t in the t-th generation.

Input: D_S^t-the source data in the t-th generation, P^t-the current population.

Set the target data D_T^t empty.
for i=1:$|P^t|$ **do**
  Find the nearest neighbor of P_i^t (NB) in D_S^t.
  if $NB \notin D_T^t$ **then**
    Add NB to D_T^t.
  end
end

Output: D_T^t.

training data. In each generation of the proposed algorithm, after obtaining the transfer surrogate model $\hat{f}_T(\mathbf{x})$, one promising and one uncertain solution ($\mathbf{x}^p$ and $\mathbf{x}^u$) from the current population P are evaluated using the high-fidelity fitness, which are selected according to the following criteria:

$$\mathbf{x}^p = \arg\min_{\mathbf{x} \in P \setminus D_S} \hat{f}_T(\mathbf{x}) \tag{6}$$

$$\mathbf{x}^u = \arg\max_{\mathbf{x} \in P \setminus D_S} \left| f_l(\mathbf{x}) - \hat{f}_{RBF}(\mathbf{x}) \right| \tag{7}$$

The promising solution $\mathbf{x}^p$ is the solution with the best predicted fitness value, and the uncertain solution $\mathbf{x}^u$ is the solution with the largest difference between those two low-fidelity fitness values. To avoid repeated solutions for the high-fidelity fitness evaluations, the selection is performed inside $P \setminus D_S$. Samples $\mathbf{x}^p$ and $\mathbf{x}^u$ are added to D_S to replace its two oldest members, then the RBF model and transfer surrogate model are retrained in the beginning of the next generation. The pseudo code of TSA-BFEA is summarized in Algorithm 2.

Algorithm 2 Pseudo code of TSA-BFEA.

Input: A bi-fidelity optimization problem with two fidelity fitness functions $f_l(\mathbf{x})$ and $f_h(\mathbf{x})$.

Initialize population P_0 with N solutions.
Sample N solutions to build D_S using Latin hypercube sampling.
Calculate the fitness of D_S using $f_h(\mathbf{x})$.
$t = 1$
while the cost budget is not exhausted **do**
    Generate offspring population O_t using variation operators for the parent population P_{t-1}.
    $P_t = P_{t-1} \cup O_t$.
    Train $\hat{f}_{RBF}(\mathbf{x})$ using D_S.
    Select D_T^t from D_S according to P_t using Algorithm 1.
    Train $\hat{f}_T(\mathbf{x})$ from $f_l(\mathbf{x})$ and $\hat{f}_{RBF}(\mathbf{x})$ using D_T^t.
    Evaluate P_t using $\hat{f}_T(\mathbf{x})$.
    Select P_t with N top $\hat{f}_T(\mathbf{x})$ for the next generation.
    Calculate $\mathbf{x}^p$ and $\mathbf{x}^u$ as Equations (6) and (7).
    Evaluate $\mathbf{x}^p$ and $\mathbf{x}^u$ using $f_h(\mathbf{x})$.
    Replace two oldest members in D_S with $\mathbf{x}^p$ and $\mathbf{x}^u$.
    $t = t + 1$
end

Output: Predicted optimum in P_t.

4. Experimental Study

In this section, we analyze the behavior of TSA-BFEA on the multi-fidelity benchmark MFB problems and their variants [5]. To discuss the effectiveness of the transfer surrogate model, infill sampling criteria, and transfer stacking, we employed the following algorithms to compare with TSA-BFEA:

- **EA-HF**: an EA using the high-fidelity fitness function only.
- **EA-AFAg**: an EA using the generation-based fidelity adjustment strat-

egy in [5].

- **EA-RBF**: an EA using the RBF model to approximate the high-fidelity fitness function, and the infill sampling criteria defined in Equations (6) and (7), which are same with the proposed algorithm.

- **RBF-cSAEA**: an EA using the RBF model to approximate the difference between the low- and high-fidelity fitness function [55], where the population is re-evaluated using the high-fidelity fitness function every 10 generations to enrich the training data.

- **EA-CK**: a co-Kriging-based EA, where the lower confidence bounding (LCB) is employed as its infill sampling criterion. The co-Kriging model adopts a simple Kriging model [56] with a Gaussian kernel, whose hyper-parameters are optimized using the method in [57].

- **MCEEA**: a multi-tasking EA which considers the optimization of the low-fidelity fitness function as a cheap task and the optimization of the high-fidelity fitness function as an expensive task [41], where the knowledge transfer from the low- to high-fidelity fitness function is performed in every five generations and the population size for the expensive task is 25% of that of the cheap task.

Since bi-fidelity optimization problems are concerned in this work, we employ MFB1, MFB8, and MFB12 with two discrete fidelity levels ($\phi = 1000$ and $\phi = 10000$) as the test problems, where a high-fidelity evaluation needs 10 times of cost of a low-fidelity evaluation [5].

The EA involved in this work employs the real coded genetic algorithm [58] with simulated binary crossover (SBX) with $\eta = 15$, polynomial mutation with $\eta = 15$, and tournament selection. For the SAEAs under comparison, their training datasets start with 100 samples, and their infilling samples replace the oldest samples in their training datasets. All compared algorithms on each test problem stop by 1000 high-fidelity fitness evaluations (equivalent to 10000 low-fidelity fitness evaluations due to the cost differences in [5]).

4.1. Accuracy of the Transfer Surrogate Model

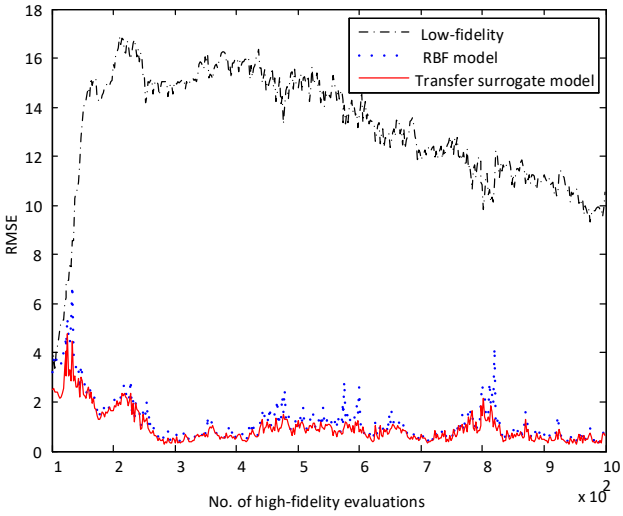

Figure 3: The average RMSE of the RBF model and the transfer surrogate model of the population during the optimization process in TSA-BFEA on bi-fidelity MFB1.

The transfer surrogate model (refer to Fig. 2) estimates the fitness for selection in the proposed algorithm, which makes its accuracy essential for the optimization performance. To compare the approximation error of the RBF model, the low-fidelity fitness function, and the transfer surrogate model, we use the population in each generation to calculate the root mean square error (RMSE). After repeating the proposed algorithm on MFB1 for 30 independent runs, the average RMSE during the optimization process is shown in Fig. 3.

The low-fidelity fitness function and RBF model have different RMSE during the optimization process since they are different approximations to the high-fidelity fitness function. As the evolutionary optimization proceeds, the population concentrates in a small region, which is the reason why the RMSE of the low-fidelity fitness function and the RBF model changes. However, the RMSE of the transfer surrogate model is smaller than that of the low-fidelity fitness function and RBF model regardless of the optimization stage. It confirms that the transfer surrogate model is able to enhance the accuracy of the RBF

model with the help of knowledge transfer from the low-fidelity fitness function, thereby improving the optimization results.

4.2. Comparisons of Infill Sampling Criteria

In TSA-BFEA, both one promising solution ($\mathbf{x}^p$) and one uncertain solution ($\mathbf{x}^u$) are chosen based on the criteria in Equations (6) and (7) to re-evaluate their fitness using the high-fidelity fitness function. To analyze the influence of both criteria, we compare the following TSA-BFEA variants on a representative problem (bi-fidelity MFB1):

- TSA-BFEA: the proposed algorithm using both criteria in Equations (6) and (7).
- TSA-BFEA-P: the proposed algorithm using the criterion in Equation (6) only.
- TSA-BFEA-U: the proposed algorithm using the criterion in Equation (7) only.
- TSA-BFEA-WOI: the proposed algorithm without using any criterion in Equations (6) or (7).

We repeat those four variants for 30 independent times. The optimal solutions obtained by TSA-BFEA variants on bi-fidelity MFB1 are shown in Fig. 4. TSA-BFEA outperforms TSA-BFEA-P, TSA-BFEA-U and TSA-BFEA-WOI. TSA-BFEA-WOI has the worst average fitness, which implies the benefits of infilling samples to the proposed algorithm. As discussed in [53], sampling $\mathbf{x}^p$ can improve the accuracy of the transfer surrogate model in the area around the predicted local optimum, sampling $\mathbf{x}^u$ can improve the accuracy of the transfer surrogate model in the uncertain area, and simultaneously sampling both $\mathbf{x}^p$ and $\mathbf{x}^u$ strikes a balance between exploitation and exploration for the optimization search. That is the reason why both criteria are important for enhancing the quality of the transfer surrogate model in TSA-BFEA.

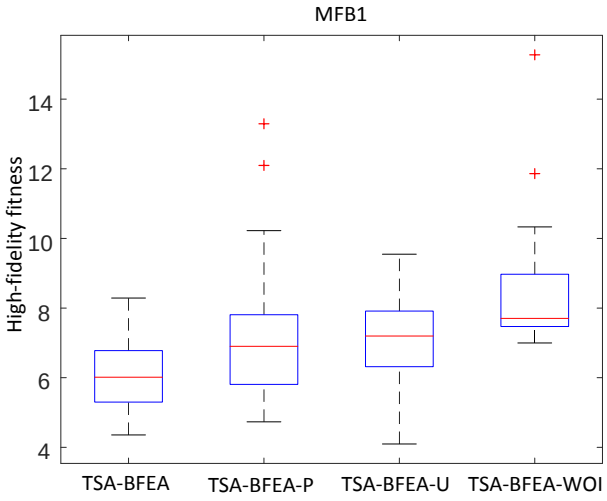

Figure 4: Solutions obtained by TSA-BFEA variants on bi-fidelity MFB1 for 30 independent times.

4.3. Comparative Experiments

To further discuss the behavior TSA-BFEA, we compare it with EA-HF, EA-AFAg, EA-RBF, RBF-cSAEA, MCEEA, and EA-CK on bi-fidelity MFB1, MFB8, and MFB12, where $\phi = 1000$ is set as the low-fidelity level and $\phi = 10000$ is the high-fidelity level. The characteristics (surrogate assists, number of the used fidelity levels, fidelity adjustment strategies, and transfer learning techniques) of those compared algorithms are shown in Table 1.

Table 1: Characteristics of compared algorithms.

Algorithm	Surrogate assists	No. of fidelity	Fidelity adjustment	Transfer learning
EA-HF	No	Single-fidelity	No	No
EA-AFAg	No	Bi-fidelity	Yes	No
EA-RBF	Yes	Single-fidelity	No	No
EA-CK	Yes	Bi-fidelity	No	No
RBF-cSAEA	Yes	Bi-fidelity	No	No
MCEEA	No	Bi-fidelity	No	Multi-tasking
TSA-BFEA	Yes	Bi-fidelity	No	Transfer stacking

The solutions obtained by EA-HF, EA-AFAg, EA-RBF, RBF-cSAEA, EA-

Table 2: Optimal solutions obtained by EA-HF, EA-AFAg, EA-RBF, RBF-cSAEA, EA-CK, MCEEA, and TSA-BFEA on bi-fidelity MFB1, MFB8, and MFB12. The results are shown in the form of mean ± standard deviation. The best fitness values among all the compared algorithms for each problem are highlighted.

Problem	EA-HF	EA-AFAg	EA-RBF	RBF-cSAEA	EA-CK	MCEEA	TSA-BFEA
MFB1	18.7±1.3	11.1±3.2	9.9±2.2	10.4±1.4	6.6±1.1	7.6±0.9	**6.1±1.0**
MFB8	18.3±1.5	11.6±2.6	9.8±1.9	9.1±1.4	7.3±1.3	9.3±1.4	**5.3±1.0**
MFB12	18.3±1.5	5.2±2.9	10.2±2.3	14.1±2.6	**3.9±1.1**	5.6±0.8	18.9±3.3

CK, MCEEA, and TSA-BFEA on bi-fidelity MFB1, MFB8, and MFB12 averaged over 30 independent runs are presented in Table 2. According to the average performance of the compared algorithms, TSA-BFEA and EA-CK are two best-performing algorithms on MFB1 and MFB8, EA-CK is the best-performing algorithm on MFB12. For all the problems, EA-RBF finds worse solutions than those two best-performing algorithms, EA-HF has the worst performance on all three problems. It is not surprising that EA-HF without any surrogate assistance performs the worst. Driven by the generation-based fidelity adjustment taking advantage of both fidelity levels, EA-AFAg performs better than EA-HF.

In addition, we compare the computational complexity of two best SAEAs (TSA-BFEA and EA-CK) on three test problems given in Table 3. We use the Wilcoxon signed-rank test (significance level=0.05)[59, 60] to compare the algorithms. The results in Table 3 indicate that TSA-BFEA needs much less computation time than EA-CK due to the high complexity of the co-Kriging model.

Table 3: Execution time (s) of TSA-BFEA and EA-CK on bi-fidelity MFB1, MFB8, and MFB12. The results are shown in the form of mean ± standard deviation. The results are analyzed by the Wilcoxon signed-rank test (significance level=0.05). The significant algorithms for each problem are highlighted in bold face.

Problem	EA-CK	TSA-BFEA
MFB1	12331.0±606.7	**22.8±3.9**
MFB8	18793.3±647.8	**24.2±2.0**
MFB12	15119.0±563.1	**24.4±3.3**

To further discuss the behavior of the compared algorithms, we replace their

high fitness functions with Ackley, Rastrigin, Rosenbrock, Griewank, and Ellipsoid functions [61], whose feasible space is scaled to $[-1, 1]^d$ to be same as that of the MFB problems. Thus, those new bi-fidelity optimization problems are named by their error types and high fitness functions. For example, a bi-fidelity Ackley function with resolution errors is named as MFB1-Ackley. We compared EA-HF, EA-AFAg, EA-RBF, RBF-cSAEA, EA-CK, MCEEA, and TSA-BFEA using the computational budget of 1000 high-fidelity fitness evaluations (equivalent to 10000 low-fidelity fitness evaluations due to the cost differences in [5]) on those 15 new problems. All compared algorithms are run independently for 30 times.

Table 4: Optimal solutions obtained by EA-HF, EA-AFAg, EA-RBF, RBF-cSAEA, EA-CK, MCEEA, and TSA-BFEA on bi-fidelity Ackley, Rastrigin, Rosenbrock, Griewank, and Ellipsoid with resolution, stochastic, and instability errors. The results are shown in the form of mean ± standard deviation. The results are analyzed by the Friedman test with the Bergmann-Hommel post-hoc test (TSA-BFEA is the control method and the significance level is 0.05). The best fitness values among all the compared algorithms for each problem are highlighted.

Problem	EA-HF	EA-AFAg	EA-RBF	RBF-cSAEA	EA-CK	MCEEA	TSA-BFEA
MFB1-Ackley	18.3±0.4	18.8±0.5	10.6±1.3	20.0±0.6	19.1±1.3	19.2±0.6	**7.7±0.8**
MFB1-Rastrigin	259.4±21.9	107.6±41.0	106.7±20.9	113.6±17.2	85.6±28.7	84.1±9.4	**63.1±9.1**
MFB1-Rosenbrock	1584.7±244.9	333.9±113.0	198.0±42.1	389.4±70.2	310.5±76.1	308.7±47.7	**57.2±9.1**
MFB1-Griewank	188.8±31.7	5.3±1.3	5.7±1.1	10.0±2.5	8.7±7.6	8.7±1.9	**4.0±0.8**
MFB1-Ellipsoid	773.5±106.9	8.9±7.3	15.2±6.2	30.1±6.5	32.9±27.3	26.5±7.1	**8.2±1.7**
MFB8-Ackley	18.5±0.5	18.4±0.5	11.1±1.1	20.1±0.4	20.0±0.3	19.2±0.4	**7.6±0.9**
MFB8-Rastrigin	252.8±16.9	120.8±30.2	97.7±15.5	97.3±8.7	77.9±17.9	98.4±16.4	**54.1±10.3**
MFB8-Rosenbrock	1622.7±345.8	321.7±76.6	164.1±27.7	334.0±72.2	338.1±66.4	352.8±56.6	**60.1±11.1**
MFB8-Griewank	192.9±30.6	7.5±2.2	4.7±0.8	9.3±1.7	9.7±6.6	9.1±1.7	**3.9±0.7**
MFB8-Ellipsoid	750.4±105.1	12.6±15.4	15.6±4.4	30.7±6.6	42.8±33.7	23.8±5.6	**6.4±1.3**
MFB12-Ackley	18.5±0.4	**3.4±1.4**	7.6±0.7	19.4±1.3	4.4±0.4	7.6±0.8	12.3±0.9
MFB12-Rastrigin	251.9±17.0	**30.4±20.2**	104.9±16.2	193.5±35.3	43.0±11.0	56.5±6.6	192.6±19.3
MFB12-Rosenbrock	1566.0±308.2	135.5±174.5	**61.1±11.6**	699.9±432.2	352.2±271.0	161.0±36.0	123.2±21.9
MFB12-Griewank	196.0±22.3	**1.8±1.6**	3.9±1.0	19.0±7.2	17.9±14.1	9.0±2.0	12.0±2.7
MFB12-Ellipsoid	756.7±120.3	**2.5±2.0**	15.9±4.8	37.4±9.5	46.8±38.9	29.0±7.9	30.8±8.9
Average rank	6.5	2.8	2.6	5.4	4.1	4.2	2.3
Adjusted p-value	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	NA

The results of EA-HF, EA-AFAg, EA-RBF, RBF-cSAEA, EA-CK, MCEEA,

and TSA-BFEA on bi-fidelity Ackley, Rastrigin, Rosenbrock, Griewank, and Ellipsoid with resolution, stochastic, and instability errors are given in Table 4. These results are analyzed by the Friedman test with the Bergmann-Hommel post-hoc test, where TSA-BFEA is the control method and the significance level is 0.05. The comparative results are similar to those in Table 2. Overall, TSA-BFEA using transfer learning and surrogate techniques exhibits the best performance, and EA-HF without any surrogate assistance or fidelity adjustment strategy performs the worst.

The test problems in Table 4 and in Table 2 are different, the behaviors of most compared algorithms to different types of errors are similar except for EA-RBF, EA-CK, and MCEEA. EA-CK and MCEEA degenerate their performance and are worse than EA-RBF, where the knowledge transfer between two fidelity levels does not benefit the optimization. In fact, a low-fidelity fitness evaluation needs 10% computation cost of a high-fidelity fitness evaluation, which is more expensive than surrogate models. If the knowledge transfer is not effective enough, then transfer learning based algorithms cannot deliver better performance than surrogate-assisted algorithms using the high-fidelity level only. Therefore, EA-CK and MCEEA are outperformed by EA-RBF on the new bi-fidelity optimization problems with complex exact objective functions. In addition, the level of the instability errors of MFB12 problems is large. EA-AFAg changes the low-fidelity level to the high-fidelity level at very beginning of the search. The co-Kriging model can deal with such a large constant difference between the low and high fidelity fitness functions. MCEEA keeps the elitist in a small separate population for the expensive task (optimizing the high-fidelity fitness function), which makes it less vulnerable to the large instability errors. EA-AFAg rather than EA-CK performs the best on the problems with instability errors in Table 2. Compared with EA-CK, EA-AFAg uses the high-fidelity fitness evaluations in late searching state to accurately refine the optimum. The transfer surrogate model in TSA-BFEA combines the low-fidelity fitness function and RBF model. However, the low-fidelity fitness function has a instability error which is a large constant value or zero according to the probability. Thus,

it is hard for the combined transfer surrogate model in TSA-BFEA to benefit from the low-fidelity fitness function, which is the reason why it degrades the performance on MFB12 problems. From the above results, we can make the following observations:

- Although the low-fidelity fitness function is of low accuracy, it still can help the optimizer to find good solutions within a limited computational budget.
- Surrogate models using cheaper costs than the low- and high-fidelity fitness evaluations guide the search, which are helpful for bi-fidelity optimization problems.
- The good performance of TSA-BFEA and EA-CK implies that the surrogate model taking both fidelity levels into account is effective on bi-fidelity optimization problems. However, the knowledge transfer still requires high cost-effectiveness, because it uses a number of extra low-fidelity fitness evaluations, which costs a non-ignorable part of computation budget. Even if that used budget results in an equal performance improvement of surrogate-assisted EAs, transfer learning-based algorithms have no advantages over surrogate-assisted EAs.
- Although TSA-BFEA and EA-CK achieve similar performance on MFB1 and MFB8 problems, TSA-BFEA is computationally more efficient than EA-CK in building surrogates.

5. Application to Benficiation Processes

Beneficiation processes are production processes that concentrate valuable minerals from raw ores via separating tailings and waste rock [62]. The process includes a series of production units as illustrated in Fig. 5. In the process, the raw ore is firstly separated into two types in the raw ore screening according to its size. Then, the separated large and small sized ore are delivered to low-intensity magnetic production line (LMPL) and high-intensity magnetic production line

(HMPL) separately. The larger sized ore will be roasted in the roasting unit to separate the waste rock before they are conveyed to the grinding unit. By contrast, the small sized ore containing higher magnetic can be grounded directly as shown in HMPL. The semi-finished products from grinding units in LMPL and HMPL are sent to the magnetic separation processes separately, where the tailing grade and concentrate grade can be recognized and separated. Then, the tailing and concentrate grade from the two lines are mixed together and delivered to the dewatering units. After dewatered, the concentrate ore, which is the final product, will be stored in the storeroom and tailing ore in tailing dam.

Operational indices optimization in beneficiation processes aims to improve the concentrate grade (G), concentrate yield (Y), as well as decrease the energy consumption (E) including the costs of roasting unit, grinding units via properly coordinating operating state of each unit. In this work, we consider the following 15 operational indices, namely, particle sizes of the raw ore entered in LMPL and HMPL (pl, ph) and grade of the raw ore entered in LMPL and HMPL (gl, gh), capacity and run time of shaft furnace roasting (sc, st), grade of waste ore (gw), grade of feed ore of grindings in LMPL and HMPL (gfl, gfh), capacity of grindings in LMPL and HMPL (gcl, gch) and running time of grindings in LMPL and HMPL (gtl, gth), grade of tailings from LMPL and HMPL (tl, th). These 15 operational indices are taken as decision variables and denoted as $X = (pl, ph, gl, gh, sc, st, gw, gfl, gfh, gcl, gch, gtl, gth, tl, th)$. The optimization problem can be formulated as follows:

$$\begin{aligned} &\min f_{BP}(X) = 0.15E - G - 0.0078Y \\ &s.t. G = \Phi_1(X), Y = \Phi_2(X) \\ &E = sc + 0.3st + gcl + gch + gtl + gth \end{aligned} \tag{8}$$

where Φ_1 and Φ_2 represent the unknown correlations between objectives and decision variables in the operational indices optimization problem. Unfortunately, the exact evaluation of $f_{BP}(X)$ is not available, only 450 historical data are available for optimizers. According to the prediction results in [63], a multilayer

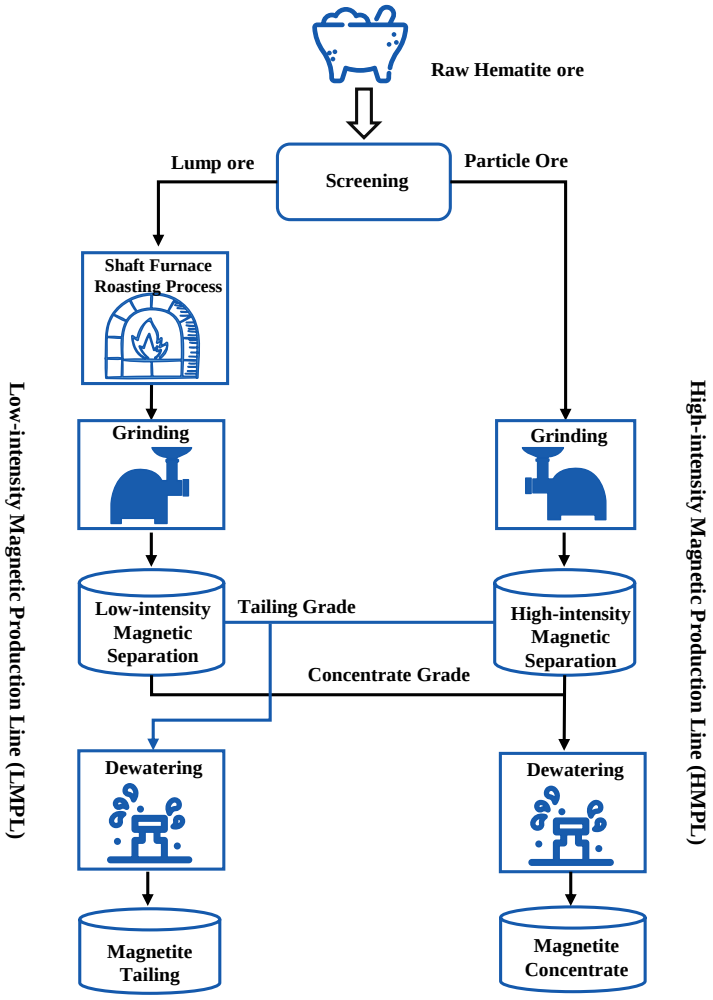

Figure 5: An illustrative diagram of the beneficiation process.

perception neural network (MLP) [64] and the first-order polynomial regression model which are trained from those historical data can be used as the high- and low-fidelity fitness functions in the optimization process. In this section, we compare EA-HF, EA-AFAg, EA-RBF, EA-CK, MCEEA, and TSA-BFEA (with 500 high-fidelity fitness evaluations as the stopping criterion) to the new formulated bi-fidelity beneficiation process optimization problem. The optimal solutions obtained by the compared algorithms in 30 independent runs are sum-

marized in Table 5. From these results, we can see that TSA-BFEA performs the best and EA-HF performs the worst on the beneficiation process problem. EA-RBF and RBF-cSAEA obtain better solution than EA-HF, which is due to their employed RBF models. However, EA-AFAg using the fidelity adjustment strategy outperforms EA-RBF and RBF-cSAEA. EA-CK and MCEEA perform better than other compared algorithms, except for TSA-BFEA. As the error type of the beneficiation process problem is not the instability error, TSA-BFEA outperforms the compared algorithms, which is consistent with the finding on the test problems.

Table 5: Optimal solutions obtained by EA-HF, EA-AFAg, EA-RBF, RBF-cSAEA, EA-CK, MCEEA, and TSA-BFEA on the beneficiation process optimization application.

Algorithm	Obtained solution
EA-HF	-130.8±5.8
EA-AFAg	-134.9±5.8
EA-RBF	-131.8±6.0
RBF-cSAEA	-131.7±5.5
EA-CK	-136.1±6.3
MCEEA	-142.9±4.2
TSA-BFEA	**-149.7±3.3**

6. Conclusions

In this work, we propose a transfer surrogate-assisted evolutionary algorithm to address bi-fidelity optimization problems, where the high-fidelity evaluations are expensive and the low-fidelity evaluations are cheap. Using the transfer stacking technique, the proposed algorithm transfers the low-fidelity fitness function and an RBF model to approximate the high-fidelity fitness function. The experimental results on the benchmark problems and beneficiation process optimization application demonstrate that the proposed algorithm can effectively solve bi-fidelity optimization problems.

Although this work confirms the effectiveness of transfer learning on bi-fidelity optimization problems, there are still some remaining challenges to be addressed in the future. Firstly, the proposed algorithm is only applicable to

the multi-fidelity optimization problems with two fidelity levels. A more generic multi-fidelity optimization algorithm based on transfer learning needs to be developed. Secondly, the proposed transfer surrogate model is ineffective in dealing with instability errors in the low-fidelity function. Other transfer learning techniques will be considered to overcome the failure of the proposed algorithm on the MFB problems with instability errors. Thirdly, this work only uses one RBF model as an additional low-fidelity fitness function. In the future, heterogeneous surrogate models can be used for transfer stacking to further improve the accuracy of the transfer surrogate model.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China (Nos. 61976165 and 61590922).

References

[1] P. J. Fleming, R. C. Purshouse, Evolutionary algorithms in control systems engineering: a survey, Control engineering practice 10 (11) (2002) 1223–1241.

[2] D. Dasgupta, Z. Michalewicz, Evolutionary algorithms in engineering applications, Springer Science & Business Media, 2013.

[3] S. Koziel, L. Leifsson, Surrogate-based modeling and optimization, Applications in Engineering.

[4] J. Branke, M. Asafuddoula, K. S. Bhattacharjee, T. Ray, Efficient use of partially converged simulations in evolutionary optimization, IEEE Transactions on Evolutionary Computation 21 (1) (2017) 52–64.

[5] H. Wang, Y. Jin, J. Doherty, A generic test suite for evolutionary multifidelity optimization, IEEE Transactions on Evolutionary Computation 22 (6) (2018) 836–850.

[6] Y. Jin, B. Sendhoff, A systems approach to evolutionary multiobjective structural optimization and beyond, IEEE Computational Intelligence Magazine 4 (3) (2009) 62–76.

[7] J. Ren, A. Thelen, A. Amrit, X. Du, L. Leifsson, Y. Tesfahunegn, S. Koziel, Application of multi-fidelity optimization techniques to benchmark aerodynamic design problems, in: 54th AIAA Aerospace Sciences Meeting, 2016, p. 1542.

[8] S. Atamturktur, Z. Liu, S. Cogan, H. Juang, Calibration of imprecise and inaccurate numerical models considering fidelity and robustness: a multiobjective optimization-based approach, Structural and Multidisciplinary Optimization 51 (3) (2015) 659–671.

[9] L. Leifsson, S. Koziel, Variable-fidelity aerodynamic shape optimization, in: Computational Optimization, Methods and Algorithms, Springer, 2011, pp. 179–210.

[10] A. Mehmani, Uncertainty-integrated surrogate modeling for complex system optimization, Ph.D. thesis, Syracuse University (2015).

[11] D. Lim, Y.-S. Ong, Y. Jin, B. Sendhoff, Evolutionary optimization with dynamic fidelity computational models, in: International Conference on Intelligent Computing, Springer, 2008, pp. 235–242.

[12] S. Koziel, Multi-fidelity multi-grid design optimization of planar microwave structures with sonnet, in: International Review of Progress in Applied Computational Electromagnetics, 2010, pp. 26–29.

[13] A. I. Forrester, A. Sóbester, A. J. Keane, Multi-fidelity optimization via surrogate modelling, in: Proceedings of the royal society of london a: mathematical, physical and engineering sciences, Vol. 463, The Royal Society, 2007, pp. 3251–3269.

[14] M. K. Zahir, Z. Gao, An improved parallel optimization framework for transonic airfoil design, Reseach Journal of Applied Sciences, Engineering and Technology 5 (22) (2013) 5209–5216.

[15] G. Sun, G. Li, M. Stone, Q. Li, A two-stage multi-fidelity optimization procedure for honeycomb-type cellular materials, Computational Materials Science 49 (3) (2010) 500–511.

[16] G. Sun, G. Li, S. Zhou, W. Xu, X. Yang, Q. Li, Multi-fidelity optimization for sheet metal forming process, Structural and Multidisciplinary Optimization 44 (1) (2011) 111–124.

[17] S. Koziel, S. Ogurtsov, S. Szczepanski, Local response surface approximations and variable-fidelity electromagnetic simulations for computationally efficient microwave design optimisation, IET microwaves, antennas & propagation 6 (9) (2012) 1056–1062.

[18] Y. Jin, Surrogate-assisted evolutionary computation: Recent advances and future challenges, Swarm and Evolutionary Computation 1 (2) (2011) 61–70.

[19] C. Smith, J. Doherty, Y. Jin, Recurrent neural network ensembles for convergence prediction in surrogate-assisted evolutionary optimization, in: Computational Intelligence in Dynamic and Uncertain Environments (CIDUE), 2013 IEEE Symposium on, IEEE, 2013, pp. 9–16.

[20] C. Smith, J. Doherty, Y. Jin, Multi-objective evolutionary recurrent neural network ensemble for prediction of computational fluid dynamic simulations, in: IEEE Congress on Evolutionary Computation, 2014.

[21] S. Ulaganathan, I. Couckuyt, F. Ferranti, T. Dhaene, E. Laermans, Variable-fidelity surrogate modelling with kriging, in: 17th International Conference on Chemistry and Chemical Engineering, World Academy of Science, Engineering and Technology, International Science Index, Computer and Information Engineering (ICCCE 2015), 2015, pp. 514–518.

[22] D. Peri, E. F. Campana, High-fidelity models and multiobjective global optimization algorithms in simulation-based design, Journal of Ship Research 49 (3) (2005) 159–175.

[23] Q. Zhou, X. Shao, P. Jiang, H. Zhou, L. Shu, An adaptive global variable fidelity metamodeling strategy using a support vector regression based scaling function, Simulation Modelling Practice and Theory 59 (2015) 18–35.

[24] D. E. Myers, Co-kriging-new developments, in: Geostatistics for natural resources characterization, Springer, 1984, pp. 295–305.

[25] L. Huang, Z. Gao, D. Zhang, Research on multi-fidelity aerodynamic optimization methods, Chinese Journal of Aeronautics 26 (2) (2013) 279–286.

[26] Z.-H. Han, R. Zimmermann, S. Görtz, A new cokriging method for variable-fidelity surrogate modeling of aerodynamic data, AIAA Paper 1225 (2010) (2010) 1–22.

[27] P. Perdikaris, D. Venturi, J. Royset, G. Karniadakis, Multi-fidelity modelling via recursive co-kriging and gaussian–markov random fields, Proc. R. Soc. A 471 (2179) (2015) 20150018.

[28] L. Le Gratiet, J. Garnier, Recursive co-kriging model for design of computer experiments with multiple levels of fidelity, International Journal for Uncertainty Quantification 4 (5).

[29] S. Koziel, S. Ogurtsov, I. Couckuyt, T. Dhaene, Variable-fidelity electromagnetic simulations and co-kriging for accurate modeling of antennas, IEEE Transactions on Antennas and Propagation 61 (3) (2013) 1301–1308.

[30] T. Chugh, Y. Jin, K. Miettinen, J. Hakanen, K. Sindhya, A surrogate-assisted reference vector guided evolutionary algorithm for computationally expensive many-objective optimization, IEEE Transactions on Evolutionary Computation 22 (1) (2018) 129 – 142.

[31] Z. Guo, L. Song, C. Park, J. Li, R. T. Haftka, Analysis of dataset selection for multi-fidelity surrogates for a turbine problem, Structural and Multidisciplinary Optimization (2018) 1–16.

[32] S. J. Pan, Q. Yang, A survey on transfer learning, IEEE Transactions on Knowledge and Data Engineering 22 (10) (2010) 1345–1359.

[33] Y.-Q. Hu, Y. Yu, W.-W. Tu, Q. Yang, Y. Chen, W. Dai, Multi-fidelity automatic hyper-parameter tuning via transfer series expansion, in: AAAI 2019, 2019.

[34] A. Gupta, Y.-S. Ong, L. Feng, Insights on transfer optimization: Because experience is the best teacher, IEEE Transactions on Emerging Topics in Computational Intelligence 2 (1) (2018) 51–64.

[35] A. Gupta, Y.-S. Ong, L. Feng, Multifactorial evolution: toward evolutionary multitasking, IEEE Transactions on Evolutionary Computation 20 (3) (2016) 343–357.

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

[37] M. Iqbal, B. Xue, H. Al-Sahaf, M. Zhang, Cross-domain reuse of extracted knowledge in genetic programming for image classification, IEEE Transactions on Evolutionary Computation 21 (4) (2017) 569–587.

[38] M. Iqbal, W. N. Browne, M. Zhang, Reusing building blocks of extracted knowledge to solve complex, large-scale boolean problems, IEEE Transactions on Evolutionary Computation 18 (4) (2014) 465–480.

[39] M. Hauschild, M. Pelikan, K. Sastry, C. Lima, Analyzing probabilistic models in hierarchical BOA, IEEE Transactions on Evolutionary Computation 13 (6) (2009) 1199–1217.

[40] S. J. Louis, J. McDonnell, Learning with case-injected genetic algorithms, IEEE Transactions on Evolutionary Computation 8 (4) (2004) 316–328.

[41] J. Ding, C. Yang, Y. Jin, T. Chai, Generalized multitasking for evolutionary optimization of expensive problems, IEEE Transactions on Evolutionary Computation 23 (1) (2019) 44–58.

[42] D. Lim, Y.-S. Ong, A. Gupta, C. K. Goh, P. S. Dutta, Towards a new praxis in optinformatics targeting knowledge re-use in evolutionary computation: simultaneous problem learning and optimization, Evolutionary Intelligence 9 (4) (2016) 203–220.

[43] A. Gupta, Y.-S. Ong, L. Feng, K. C. Tan, Multiobjective multifactorial optimization in evolutionary multitasking, IEEE Transactions on Cybernetics 47 (7) (2017) 1652–1665.

[44] J. Luo, A. Gupta, Y.-S. Ong, Z. Wang, Evolutionary optimization of expensive multiobjective problems with co-sub-pareto front gaussian process surrogates, IEEE Transactions on Cybernetics 49 (5) (2018) 1708–1721.

[45] A. T. W. Min, Y.-S. Ong, A. Gupta, C.-K. Goh, Multiproblem surrogates: transfer evolutionary multiobjective optimization of computationally expensive problems, IEEE Transactions on Evolutionary Computation 23 (1) (2019) 15–28.

[46] R. Meuth, M.-H. Lim, Y.-S. Ong, D. C. Wunsch, A proposition on memes and meta-memes in computing for higher-order learning, Memetic Computing 1 (2) (2009) 85–100.

[47] L. Feng, Y.-S. Ong, A.-H. Tan, I. W. Tsang, Memes as building blocks: a case study on evolutionary optimization+ transfer learning for routing problems, Memetic Computing 7 (3) (2015) 159–180.

[48] M. Jiang, Z. Huang, L. Qiu, W. Huang, G. G. Yen, Transfer learning-based dynamic multiobjective optimization algorithms, IEEE Transactions on Evolutionary Computation 22 (4) (2018) 501–514.

[49] B. Liu, S. Koziel, Q. Zhang, A multi-fidelity surrogate-model-assisted evolutionary algorithm for computationally expensive optimization problems, Journal of Computational Science 12 (2016) 28–37.

[50] D. Pardoe, P. Stone, Boosting for regression transfer, in: Proceedings of the 27th International Conference on International Conference on Machine Learning, Omnipress, 2010, pp. 863–870.

[51] Y. Jin, H. Wang, T. Chugh, D. Guo, K. Miettinen, Data-driven evolutionary optimization: an overview and case studies, IEEE Transactions on Evolutionary Computation 23 (3) (2019) 442–458.

[52] K.-L. Du, M. Swamy, Radial basis function networks, in: Neural Networks and Statistical Learning, Springer, 2014, pp. 299–335.

[53] H. Wang, Y. Jin, J. Doherty, Committee-based active learning for surrogate-assisted particle swarm optimization of expensive problems, IEEE Transactions on Cybernetics 47 (9) (2017) 2664–2677.

[54] J. Tian, Y. Tan, J. Zeng, C. Sun, Y. Jin, Multiobjective infill criterion driven gaussian process-assisted particle swarm optimization of high-dimensional expensive problems, IEEE Transactions on Evolutionary Computation 23 (3) (2018) 459–472.

[55] S. van Rijn, S. Schmitt, M. Olhofer, M. van Leeuwen, T. Bäck, Multi-fidelity surrogate model approach to optimization, in: Proceeding of the 5th Annual Conference on Genetic and Evolutionary Computation Conference, 2018.

[56] R. A. Olea, Geostatistics for engineers and earth scientists, Technometrics 42 (4) (2000) 444–445.

[57] R. P. Brent, Algorithms for minimization without derivatives, Courier Corporation, 2013.

[58] L. J. Eshelman, J. D. Schaffer, Real-coded genetic algorithms and interval-schemata, in: Foundations of genetic algorithms, Vol. 2, Elsevier, 1993, pp. 187–202.

[59] J. Derrac, S. García, D. Molina, F. Herrera, A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, Swarm and Evolutionary Computation 1 (1) (2011) 3–18.

[60] J. Carrasco, S. García, M. Rueda, S. Das, F. Herrera, Recent trends in the use of statistical tests for comparing swarm and evolutionary computing algorithms: Practical guidelines and a critical review, Swarm and Evolutionary Computation 54 (2020) 100665.

[61] H. Wang, Y. Jin, C. Sun, J. Doherty, Offline data-driven evolutionary optimization using selective surrogate ensembles, IEEE Transactions on Evolutionary Computation.

[62] J. Ding, T. Chai, W. Cheng, X. Zheng, Data-based multiple-model prediction of the production rate for hematite ore beneficiation process, Control Engineering Practice 45 (2015) 219–229.

[63] C. Yang, J. Ding, Y. Jin, C. Wang, T. Chai, Multitasking multiobjective evolutionary operational indices optimization of beneficiation processes, IEEE Transactions on Automation Science and Engineering 16 (3) (2019) 1046–1057.

[64] G. Cybenko, Approximation by superpositions of a sigmoidal function, Mathematics of control, signals and systems 2 (4) (1989) 303–314.