IEEE Computational Intelligence Society
Transactions on Evolutionary Computation

A New Taxonomy and Systematic Comparison of Surrogate Assisted Evolutionary Algorithms for Single-Objective Optimization

Journal:	*Transactions on Evolutionary Computation*
Manuscript ID	TEVC-00061-2020
Manuscript Type:	Survey
Date Submitted by the Author:	20-Feb-2020
Complete List of Authors:	Tong, Hao; University of Birmingham, School of Computer Science HUANG, Changwu; Southern University of Science and Technology, Department of Computer Science and Engineering Minku, Leandro; University of Birmingham, School of Computer Science Yao, Xin; Southern University of Science and Technology, Department of Computer Science and Engineering
Keywords:	Evolutionary algorithms, surrogate models, absolute fitness models, relative fitness models, computationally expensive problems

A New Taxonomy and Systematic Comparison of Surrogate Assisted Evolutionary Algorithms for Single-Objective Optimization

Hao Tong, Changwu Huang, Leandro L. Minku, *Member, IEEE*, and Xin Yao, *Fellow, IEEE*

***Abstract*—Surrogate assisted evolutionary algorithms (SAEAs), which use efficient surrogate models or meta-models to approximate the fitness function in evolutionary algorithms (EAs), are effective and popular methods for solving computationally expensive optimization problems. During past decades, a number of SAEAs have been proposed by combining different surrogate models and EAs. This paper is dedicated to providing a comprehensive review of surrogate models used in single-objective SAEAs. A new taxonomy of surrogate models in SAEAs for single-objective optimization is introduced in this paper. Surrogate models are classified into two major categories: absolute fitness models, which directly approximate the fitness function values of candidate solutions, and relative fitness models, which estimates the relative rank or preference of candidates rather than their fitness values. Then, these two categories of surrogate models are reviewed in sequence. Comprehensive experimental studies are conducted to compare different surrogate models in terms of time complexity, model accuracy, parameter influence, and the overall performance. This is helpful for researchers to select suitable surrogate models when designing SAEAs. Open research questions and future work are discussed at the end of the paper.**

***Index Terms*—Evolutionary algorithms, surrogate models, absolute fitness models, relative fitness models, computationally expensive problems.**

I. Introduction

Evolutionary Algorithms (EAs) require a so-called fitness function to evaluate the quality of candidate solutions when solving a problem, and a large number of fitness evaluations may be required in order to obtain a satisfying solution. However, in many real world problems, the fitness function is very complex and expensive to compute. For example, a single Navier-Stokes fitness evaluation will take several hours [1]. Moreover, in some problems, like aeroelastic response simulations in reducing helicopter vibration [2], the solution can only be evaluated by expensive simulation or experiments. Such problems are referred to as computationally expensive problems (CEPs). Computational cost has become a crucial challenge in applying EAs to solve CEPs.

To reduce the computational cost of solving CEPs, surrogate-assisted evolutionary algorithms (SAEAs) were developed. In SAEAs, surrogate models or meta-models are approximate models that can simulate the behavior of the original fitness function, but can be evaluated much faster and more cheaply. The easy-to-compute surrogates are used to evaluate some candidate solutions instead of evaluating them by real fitness function, so that the computational cost of the optimization procedure is reduced.

The use of surrogate models in EAs has attracted much attention over past decades, and a large number of SAEAs have been introduced since mid-1980s [3]. SAEAs have been applied to computationally expensive single-objective optimization [4][5][6], multi-objective optimization [7][8][9][10] and combinational optimization [11] problems. Additionally, SAEAs have been applied to problems other than CEPs, including interactive optimization problems, dynamic optimization [6], etc. SAEAs have also been successfully adopted to solve real-world expensive optimization problems from different domains, including engineering design [12][13], environment protection [14], resource processing [15], health industry [16] and interactive design [17][18].

Many surrogate modeling approaches have been introduced to SAEAs over the past few years. Usually, surrogate models are used to predict fitness values of new candidate solutions by approximating the fitness function. For this purpose, regression models, such as polynomial regression, Gaussian Process regression (a.k.a. Kriging model), and Radial basis functions, are commonly used in SAEAs [19]. This kind of surrogate models which directly predict the fitness values for candidates are referred to as absolute fitness models in this paper. In recent years, some new surrogate models were introduced to SAEAs by providing the relative rank or preference of candidate solutions rather than predict their fitness values [20][21][22][23]. In this paper, this new kind of surrogate models are referred to as relative fitness models. The modeling techniques and usages of relative fitness models are different from commonly-used absolute fitness models. However, the existing surveys mostly focus on absolute fitness models that provide absolute fitness value predictions, and, to the best of our knowledge, there is no comprehensive review systematically discussing the absolute and relative fitness models. Additionally, there is still no guideline for choosing suitable surrogate models when designing SAEAs.

This motivates us to give a more comprehensive review that contains both absolute and relative fitness models used in SAEAs for single-objective optimization problems. The remainder of this paper is organized as follows. Section II

Hao Tong and L.L.Minku are with the School of Computer Science, University of Birmingham, Edgbaston, Birmingham B15 2TT, UK. (e-mail: hxt922@cs.bham.ac.uk, L.L.Minku@cs.bham.ac.uk.)

Changwu Huang and Xin Yao (*the corresponding author*) are with the Shenzhen Key Laboratory of Computational Intelligence, University Key Laboratory of Evolving Intelligent Systems of Guangdong Province, Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen 518055, China. (email: huangcw3@sustech.edu.cn, xiny@sustech.edu.cn)

gives a brief introduction to the general SAEA framework and presents our new taxonomy for surrogate models in SAEAs. Sections III and IV review absolute fitness models and relative fitness models, respectively. Section V compares different surrogate models from several perspectives, including time complexity, model accuracy, parameter influence, and overall performance, through experimental studies. The paper closes with a brief conclusion and a discussion of future works in Section VI.

II. General framework of single-objective SAEA and Categorisation of surrogate models

A. Overview

The general framework of SAEAs is for single-objective optimization illustrated in Fig. 1. On the basis of an EA, an SAEA introduces new steps or issues, including the construction of surrogate, interaction between the surrogate and the EA, and re-evaluation of some candidate solutions. The interaction and re-evaluation steps together are also known as model management or evolution control in the literature [4]. Specifically, the construction step involves building or updating surrogate model(s), the interaction concerns the mechanism of incorporating surrogate model(s) into the EA, and the re-evaluation refers to identifying some individuals (candidate solutions) to be evaluated by the original fitness function in the optimization process.

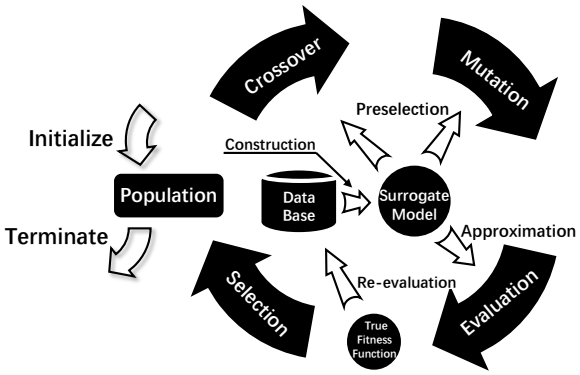


Fig. 1: The general framework of SAEAs. The crossover, mutation, evaluation and selection are common operators of a basic EA. An SAEA integrates a surrogate model into the basic EA.

a) Model Construction: Before using surrogate(s) in the evolutionary search, a SAEA needs to construct surrogate model(s) based on a set of candidate solutions evaluated by the true fitness function. The initial sample data set, which is used to train the initial surrogate, can be collected by running the EA for several generations without a surrogate or by some sampling strategies (also known as design of experiments, DOE). Commonly, a sampling strategy or DOE, which aims at maximizing the amount of information gained from an experimental study while minimizing the amount of data to be collected [24], is used to collect the initial data set owing to its efficiency in data collection. There mainly exists two kinds of sampling strategies: static and adaptive methods. Static sampling methods put emphasis on space-filling, that is, locating samples (or design points) as uniformly as possible across the space. Latin hypercube sampling (LHS) [25], which is the most commonly used DOE technique, is a typical static sampling method. Adaptive sampling strategy generates samples sequentially (iteratively) with the assistance of some system information [26] obtained from previously collected data. More about adaptive sampling can be found in [27], which summarized the existing adaptive sampling approaches. The trade-off between local exploitation and global exploration needs to be considered when building an accurate surrogate model.

b) Interaction (pre-selection and approximation): The constructed surrogate can be used in the individual creation and/or evaluation steps of an EA, as shown in Figure 1. Using the surrogate model in the creation of individuals, i.e., in crossover or mutation, is known as pre-selection. Generally, in pre-selection, several new individuals (offspring) are generated by variation operators and then the surrogate model is used to pre-select the most promising offspring. After the pre-selection, the exact fitness values of the selected individuals are still computed in the evaluation step, ensuring that the optimization process can converge to the right solutions. This method reduces the computational cost by improving the quality of individuals evaluated by the true function [28].

When using surrogate models in the evaluation step of an EA, a portion of the fitness evaluations is provided by the surrogate model rather than by the true fitness function. This is to reduce the high computational cost related to the evaluation of all individuals, which can be very expensive when solving CEPs. By replacing expensive fitness evaluations with efficient surrogate model predictions, the computational cost can be reduced significantly [4]. However, due to the uncertainty or inaccuracy of surrogate models, if all the individuals are evaluated by surrogates, the SAEA may converge to a false optimum introduced by surrogates. Thus, surrogate models should be used together with the true fitness function in fitness evaluation to prevent the optimization process from being misled by false optima. This is the motivation for re-evaluation.

c) Re-evaluation: This is a key step that affects the convergence of SAEAs. There are several strategies for selecting individuals to be re-evaluated by the true fitness function. The most intuitive strategy is to select individuals which are the fittest according to the fitness predictions from the surrogate model. For example, in Particle Swarm Optimization (PSO) [29], if the personal best and global best individuals are challenged, the challengers have to be re-evaluated to determine whether the best position should really be updated or not. Another different approach is based on clustering techniques [30] [31]. The population are clustered into several groups, and then, in each group, the individual closest to the center is re-evaluated.

The above two re-evaluation strategies only consider the predictive fitness without any information from the model itself. There are some criteria for re-evaluation which combine fitness prediction and uncertainty information of models to balance exploitation and exploration [32][33]. Examples of these strategies includes the Expected Improvements (EI) [34],

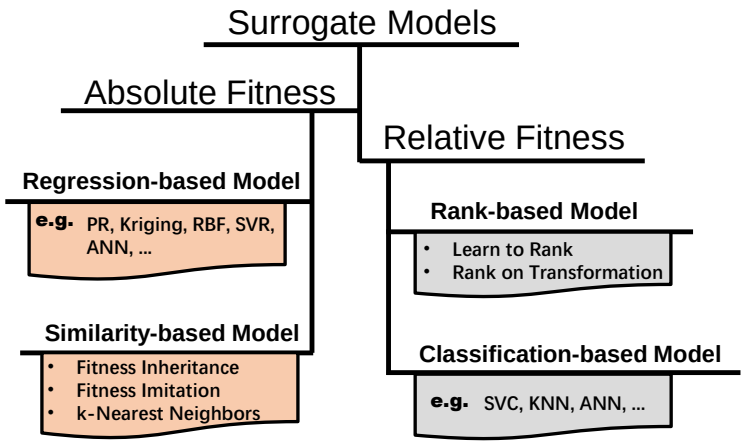


Fig. 2: Categorisation of surrogate models in SAEAs

Lower Confidence Bound (LCB) [35], and Probability of Improvements (POI) [36]. However, such strategies are limited to surrogate models that can provide uncertainty information, such as the Kriging model [37]. Recent re-evaluation strategies [38][39] try to apply novel methods to determine the re-evaluation, whose main targets are to achieve the trade-off between exploitation and exploration. For example, Wang et al. [38] proposed a committee-based active learning assisted PSO to solve CEPs.

B. Categorisation of Surrogate Models in SAEAs

Obviously, a surrogate model is a key element in an SAEA. The construction, interaction, and re-evaluation are all related to the surrogate. In the literature, plenty of surrogate models have been adopted to assist EAs. In existing reviews, surrogate models are divided into different categories based on different criteria. For example, [19][4] categorized SAEAs into individual-based, generation-based and population-based evolution control assisted algorithms. [6][11] discussed single- and multi-model(s) in SAEAs according to the number of models used for fitness approximation. [5] researched the generalized evolutionary framework focusing on *curse of uncertainty* and *bless of uncertainty*. [40] discussed models used in evolutionary algorithm in the literature, including estimating the distribution, building the inverse model and surrogate model. Additionally, there also exist many surveys about multi-objective SAEA. [8] reviewed computationally expensive MOEA based on what kind of an approximation used in the algorithm. In [10], six metamodeling frameworks are summarised based on the cardinality of meta models for objectives and constraints in MOEA. [9][41] introduced several practical cases in expensive multi-objective optimization area.

It is no doubt that, in SAEAs, surrogate models are used to make predictions, whether to predict the fitness values or preferences of candidate solutions generated in evolutionary search. Based on this principle, this paper classifies the surrogate models in SAEAs into two main categories:

- Absolute Fitness Models, which directly predict the fitness function values of candidate solutions in the optimization process.
- Relative Fitness Models, which provide estimation of rank or preference of candidates rather than the absolute fitness values.

Relative fitness models define a new type of surrogates which can assist EAs in solving expensive problems, and it also introduces a series of new model management strategies which are different from those used for absolute fitness models, as will be explained in Section IV. In some practical problems, such as interactive optimization where human feedback can be hard to estimate, but can be captured by relative ranking. Therefore, it is necessary to give a systematic review and comparison of different surrogate models including both absolute and relative fitness models so that we can understand better the strength and weakness of different methods.

Based on the above discussion, the existing surrogate models in SAEAs are classified into two main categories, i.e., absolute and relative fitness models, and each main category is further divided into two subcategories, as shown in Fig. 2. Absolute fitness models are divided into regression-based and similarity-based models. Relative fitness models are divided into rank-based and classification-based models. The following section will give a brief review of both absolute and relative fitness models.

III. Absolute Fitness Models

Absolute fitness models are commonly used in SAEAs. They aim at predicting the fitness values of individuals. Let F denote the true fitness function. The goal of constructing an absolute fitness model is to learn a model $\hat{F}$ from some evaluated samples (called training data) to approximate the input-output relationship of function F, so that the model can predict the fitness value of any new individual. An illustrative flow diagram of SAEAs using absolute fitness models is given in Fig. 3. According to modeling approaches, absolute fitness models can be divided into two subcategories: regression-based and similarity-based models.

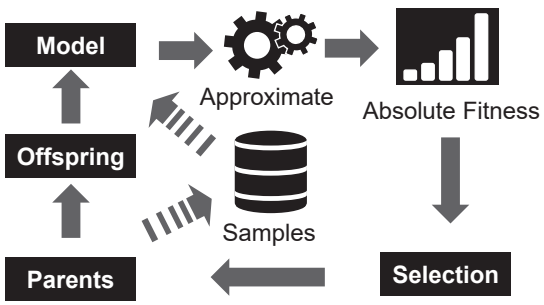


Fig. 3: Illustration of SAEA using absolute fitness model.

A. Regression-based Models

Regression-based models are widely used to emulate the relationship between the input and output of a system [33][19]. In SAEAs, regression-based models are the approximate model $\hat{F}$, which maps a solution vector (candidate solution or individual) $\mathbf{x} \in \mathbb{R}^D$ to its fitness value (output) $y = F(\mathbf{x}) \in \mathbb{R}$. Given the training data set $\mathcal{D} = [(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), ..., (\mathbf{x}_n, y_n)]$, the objective of training or constructing a regression model $\hat{F}$ is to minimize the error function (also called the loss function) $E_{\hat{F}}$ which represents the discrepancy between real fitness values and model predictions on the training data set. The most commonly adopted error function is the mean squared error (MSE) defined as

$$E_{\hat{F}} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{F}(\mathbf{x}_i))^2$$

where n is the number of samples in training data set $\mathcal{D}$, y_i and $\hat{F}(\mathbf{x}_i)$ denote the true and predicted fitness values of individual $\mathbf{x}_i \in \mathcal{D}$, respectively.

Based on regression techniques, many surrogate models are used to assist evolutionary algorithms in the literature, including Polynomial Regression (PR), Support Vector Regression (SVR), Kriging Model, Radial Basis Function (RBF), etc [33]. Thanks to the use of these models, the computational resources are significantly reduced. Bernardino et al. [42] introduced a simple weighted linear regression to enhance the performance of a clonal selection algorithm when solving CEPs. SVR model is also effective in some practical applications, such as the optimization of railway wind barriers [43], although the training process will be very hard when problems' dimension is very high. Kriging model is popular in evolutionary algorithms because it can provide the uncertainty as well as predictive fitness [44]. Additionally, RBF models have also been applied to many real world problems [13][45].

However, there is no clear conclusion on which regression model is the best because each model has its own strengths and weaknesses according to experimental studies [46]. Jin et al. [47] analyzed PR, Kriging method and RBF on fourteen test problems. They showed that RBF is the best for high order non-linear problems both for small and large scales. For low-order non-linear problems, PR is the best for local modeling, while the Kriging method is the most suitable for global modeling. The results also showed that the Kriging method is the most sensitive to noisy data. Nik et al. [48] and Diaz et al. [49] showed similar results that the Kriging method is the relatively better method for low dimensional problems, while RBF is more suitable for high-dimension problems. Additionally, RBF has shown excellent capability of being both local and global surrogates [50].

B. Similarity-based Models

Similarity-based surrogate models are another sub-category of absolute fitness models [51]. Different from regression-based models, they approximate fitness values based on the correlation between non-evaluated and evaluated individuals. The competitive performance of similarity-based models assisted EAs has been shown in [52] and [53], where similarity-based models were combined with state-of-art EAs to address real world problems. Similar to regression-based models, similarity-based models also involve a function $\hat{F}$ which provides an approximate fitness value $\hat{F}(\mathbf{x}) \in \mathbb{R}$ to a solutions $\mathbf{x} \in \mathbb{R}^D$. The main basis of similarity-based models is the correlation between individuals. Therefore, we can provide a general formulation of similarity-based models as follows:

$$\hat{F}(\mathbf{x}) = \sum_{i=1}^{N_c} Cor(\mathbf{x}, \mathbf{x}_i)F(\mathbf{x}_i) \quad (1)$$

where N_c denotes the number of individuals used to derive the fitness of a new individual $\mathbf{x}$, $Cor(\mathbf{x}, \mathbf{x}_i)$ represents the correlation between two individuals, and $F(\mathbf{x}_i)$ denotes the true fitness of related individual.

In the literature, there are three kinds of surrogate models based on the similarity idea, including *Fitness Inheritance*, *K Nearest Neighbors* and *Fitness Imitation*. Fitness inheritance, which estimates the fitness of a new solution based on the fitness of its parents, was fist proposed in 1995 [54]. Many SAEAs using fitness inheritance have been proposed during the last decades. For example, Sun et al [55] proposed an efficient algorithm which combined fitness inheritance and a PSO algorithm to solve expensive large scale problems. For K nearest neighbors, the fitness of a new individual is determined by its K nearest neighbor individuals, which can be seen as a generalized form of fitness inheritance [51]. In contrast, the fitness imitation clusters the population into several groups and one individual is selected as the representative of each group. The representative individuals are evaluated using the actual fitness function, whereas the remainder of each group is estimated by similarity measurements [4].

Similarity-based models are easy-to-construct since they are essentially based on the similarities between individuals. They do not have to define a specific formulus or to optimize the model parameters with training data, but only need to calculate the similarities. Thanks to the limited size of a sample database in expensive problems, the time for finding nearest neighbours is not particularly costly, and thus, the computational cost in creating and using similarity-based models is much cheaper than regression-based models in most SAEAs. The similarity measurement is a key issue for similarity-based fitness approximation, which directly affects the performance of surrogates.

Nonetheless, the accuracy of similarity-based models deteriorates significantly when the problem is highly nonlinear and/or the search space is large. As a result, it is better to employ similarity based models as local surrogates. Due to model's structural characteristics, each similarity-based model has its range of applications. Taking Artificial Immune System as a typical example, fitness inheritance is not appropriate because each offspring only has one parent available [56].

C. Discussion

Absolute fitness models are straightforward methods for fitness approximation and play an important role in SAEAs. Both regression-based and similarity-based models provide approximate fitness values for individuals (solutions), and then the predicted fitness values are used in population creation or individual selection in EAs.

Compared with regression-based models, similarity-based models are simpler and easier to compute and use, since only the similarity between individuals is considered for predictions. On the contrary, regression-based models are more accurate than similarity-based models to some extent, because they optimize parameters in the surrogate model to capture the potential patterns of true fitness functions. Thus, regression-based models are suitable for non-linear, multi-dimensional response spaces, while similarity-based models are commonly used for local approximation in small regions.

Although absolute fitness models are powerful methods to approximate the fitness functions for CEPs, it is not straightforward to use them in SAEAs, since some other aspects like fidelity and/or complexity of the model also need to be taken into account. Therefore, in terms of model management strategies, it is important to balance the accuracy and efficiency of absolute fitness models in SAEAs [6].

IV. Relative Fitness Models

Relative fitness models focus on the relative preference between individuals rather than their absolute fitness values. In EAs, the fitness of individuals, which represents the quality of candidate solutions, is the basis for determining whether an individual is selected or eliminated by selection operators. Generally, fitter individuals, i.e., individuals with higher fitness values, are selected to enter the next generation. For some EAs, such as Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [57] and Self-Learning Particle Swarm Optimizer (SLPSO) [58], selection depends on the ranking of individuals. In this case, the result of selection is only related to the rank of individuals. In other words, if the absolute fitness values of individuals changed, but their rank remains the same, the result of selection does not change. Relative fitness methods, which focus on the comparative result of the current population, are suitable in this case, and thus have been proposed recently. A general framework of relative fitness model assisted EAs is shown in Figure 4.

Different from absolute fitness approximation, a relative fitness model $\hat{F}$ provides the preference $\widehat{y}$ of a new individual in a specific group rather than its absolute fitness value. In the existing literature, the relative fitness approximation $\widehat{y}$ is

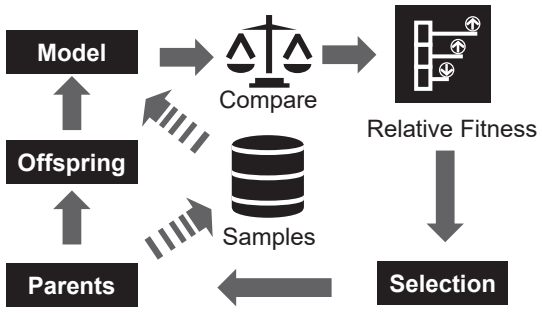


Fig. 4: Illustration of SAEA using relative fitness model.

given as a rank label $\{r_1, r_2, ..., r_N\}$ or a classification label $\{+1, -1\}$. According to this, relative fitness models are further categorized into rank-based and classification-based models.

A. Rank-based Models

Rank-based models predict the relative rank of a new population based on evaluated samples and then the individuals with a higher rank survives to the next generation [59]. Generally, a population of size N is ranked by a surrogate model $R(\cdot)$ from best to worst as presented in Equation (2),

$$R(\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N) = (\mathbf{x}_{1:N}, \ldots, \mathbf{x}_{\mu:N}, \ldots, \mathbf{x}_{N:N}) \quad (2)$$

where $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N$ represent a new population (without true fitness) and $\mathbf{x}_{\mu:N}$ represents the μ_{th} best individual in the population ranked by the model.

For a rank-based model, the predicted order or rank is more important than the fitness value, so that a reasonable error function named as order-error (OE) [60] expressed in Equation (3) can be used as the loss function.

$$Eor_R = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} \theta(y_i - y_j, R(\mathbf{x}_i) - R(\mathbf{x}_j)) \quad (3)$$

where $\theta(a, b) = (a \leq 0)$ xor $(b \leq 0)$, y_i, y_j are true fitness and $R(\mathbf{x}_i), R(\mathbf{x}_j)$ are ranks predicted by the surrogate model.

There are two different rank-based models motivated from different motivations in the literature. The first idea is to transfer the original objective space into another equivalent but different space, which we named it *Rank on Transformation*. In the new space, the rank of individuals is maintained but it is much easier to be approximated. Gong et al [61] transferred the expensive objective function into a probability density function and then built a cheap rank-based model by approximating the probability density. On the contrary, another kind of rank-based models directly predict the rank of individuals by using learning to rank algorithms [62]. The first work was in 2006 when Runarsson [20] proposed a rank-based surrogate model (by using ordinal regression) to assist an evolution strategy. After that, more work follows, e.g., authors in [63] adopted RankSVM to enhance CMA-ES for solving CEPs and Lu et al. [23] employed RankSVM to pre-select the high quality individuals in differential evolution algorithms. Huang et al. [64] adopted the approximate ranking procedure based on a Kriging model to assist the search process of CMA-ES.

In rank-based models, it is particularly important to select appropriate training samples. If the training samples are too close to each other, the discrepancy between them is hard to distinguish. If samples are too far away from each other, the precision of the rank model may significantly deteriorate. As a result, it is hard to guarantee the accuracy of rank-based surrogates, which in turn affects the convergence of SAEAs. Therefore, rank-based models are usually applied for pre-selection in the literature [23].

B. Classification-based Models

Classification-based models were first mentioned in 2003 [18] and have developed significantly in recent years. Different from rank-based models, classification-based models just consider the comparison result between individuals and a reference solution. The mechanism of classification-based models is shown in Fig. 5. A classifier is built based on a single reference individual and its neighbors. If the neighbor's fitness is greater than the reference, it is assigned with label +1, otherwise label -1 is assigned to it. A different classifier is built for each reference point. New individuals are then assigned label +1 or -1 by each classifier.

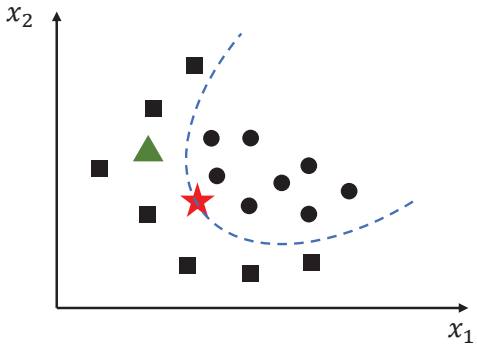


Fig. 5: Classification-based model. Star denotes the reference, triangle denotes the new individual, squares represent the samples with label -1, circles represent the samples with label +1

The task of building a classification-based model can be considered as: train a model for determining whether the new individual is better than the reference solution or not. The classifier $C : \mathbb{R}^n \rightarrow \{1, -1\}$ provides a label $+1$ or -1 for any new individual $\mathbf{x} \in \mathbb{R}^n$, where label $+1$ and -1 represents the new individual is better or worse than the reference, respectively [18]. A very simple error function to measure the quality of the classification model is presented as:

$$Eor_C = \frac{1}{N}\sum_{n=1}^{N}\mathbb{I}(L_n, C_n) \tag{4}$$

where $\mathbb{I}(a, b)$ is an indicator function which has value 1 if a equals to b and otherwise has value 0, L_n is the predefined label for training data, C_n is the predicted label. Other loss functions, like Hinge loss, can also be employed [65].

The commonly used classification models in the literature including Support Vector Classification [21], K-Nearest Neighbors [66], etc. Lu et al [21] proposed a classical classification-based SAEA for single-objective optimization, Classification-based Differential Evolution (CADE). It applied soft-margin support vector classification(SVC) as the classifier and the parent is taken as the reference individual for each offspring. The CADE mainly contains three steps including training set choosing, classifier training and exact re-evaluation. If the predictive result for an individual is worse than its parent, this individual will be re-revaluated by the actual function, otherwise it will be deleted directly.

Classification-based relative fitness models predict whether the new individual is better than the reference solution or not, and this prediction is then used to select promising individuals. However, at present, they are commonly used for pre-selection in SAEAs due to their low accuracy with limited training data. New model management strategies are still desirable for classification-based model assisted EAs.

In classification-based SAEAs, the problem of imbalanced training data is often encountered when creating classifiers. The samples for constructing a classification-based model are selected from the neighbors of a reference solution. As the evaluated points are distributed sparsely in the search space, the situation where all the selected samples are worse or better than the reference point may happen, which forms an imbalanced dataset. Lu et al. [21] employed a threshold-moving [67] method to address the imbalance dataset problem.

C. Discussion

We have reviewed the relative fitness approximate models in SAEAs in this section. Compared with regression-based absolute fitness models, relative fitness models are much simpler because they only focus on the rank or ordinal relation of individuals rather than their absolute fitness values. They are usually used as local models in SAEAs for pre-selection. In contrast, regression-based models, in general, are more elaborate and accurate than relative fitness models. Thus, the global models used in SAEAs are usually regression models.

The application of relative fitness models so far has been limited to some EAs. Rank-based model was mainly used in CMA-ES and the main achievements of classification-based models are still limited to DE and some special MOEAs.

It is necessary to mention that relative fitness models do not have the ability of describing the global trend and providing information of the whole fitness landscape. They are only used as a local model around the querying individuals or solutions in SAEAs.

In summary, relative fitness approximate methods have some advantages and are suitable for some practical problems. For example, the human interactive problems, in which the objective fitness strongly depends on the human sense and is difficult to quantify [18]. And expensive constrained optimization problem is also suitable to use relative fitness models, in which the feasiblity of solutions can be regarded as a classification problem and the vilation degree can be modelled to a rank model.

However, there are still many issues worth exploring in relative fitness models for SAEAs. For example, the model management strategies for classification-based and rank-based

models are very limited currently, which is a potential direction to do more research in the future. Moreover, it is also valuable to research relative fitness model in other optimization problems, not only in CEPs, such as constrained optimization problems, dynamical optimization problems, etc.

V. Experimental Studies and Analysis

In this section, we conduct a series of experiments to analyse the performance of absolute and relative fitness models. SAEAs with different models and strategies are compared from various aspects on single-objective optimization benchmark functions.

A. Experimental setting

1) Surrogate models with different strategies: In order to compare different surrogate models, we select one representative model from each category in our experiments. According to a recent paper comparing metamodeling techniques in evolutionary algorithms [49], RBF is the most robust and scalable regression models. So, we select the RBF model as the representive for regression model. But there's no paper comparing other kinds of surrogate models. Therefore, we choose some commonly used and typical models, that is, KNN, RankSVM, and SVC as the representative models of similarity-based models, rank-based models, and classification-based models, respectively.

Algorithm 1: Individual-based evolution control: SAEA

```
Initialize the first population with N individuals.
Evaluate N individuals.
Set FE = N.
while FE < maxFE do
    • Perform Crossover and Mutation operators to generate N offspring.
    / *** Absolute Fitness Surrogate Model *** /
    if absolute fitness model then
        Predict the fitness of each offspring.
        Re-evaluate p_sm * N best individuals.
        FE = FE + p_sm * N
    / *** Rank-based Model *** /
    if rank-based model then
        Kendall's τ = 0.
        while τ ≤ τ_th do
            Controlled individual ← the highest rank individual.
            Re-evaluate the controlled individual.
            FE = FE + 1.
            Caculate the Kendall's τ.
    • Select N best individuals from parents and offspring to form the population for next generation.
```

Algorithm 2: Pre-selection: SAEA

```
Initialize the first population with N individuals.
Evaluate N individuals.
Set FE = N.
while FE < maxFE do
    / *** Absolute Fitness Surrogate Model *** /
    / *** or Rank-based Model ***/
    if absolute fitness model or rank-based model then
        for each individual do
            Generate λ_pre offspring candidates.
            Build the surrogate model.
            if absolute fitness model then
                Predict the fitness of all candidates.
                Select the best candidate as the offspring.
            if rank-based model then
                Caculate the Kendall's τ.
                if τ > 0.5 then
                    Select the best candidate as the offspring.
                else
                    Randomly select a candidate as the offspring.
            Re-evaluate the offspring.
            FE = FE + 1.
        • Select N best individuals from parents and offspring to form the population for next generation.
    / *** Classification-based Model *** /
    if classification-based model then
        • Perform Crossover and Mutation operators to generate N offspring.
        for each individual do
            Find its parent as the reference individual.
            Build classification-based model.
            Predict the label of offspring.
            if predictive label is positive then
                Re-evaluate the offspring.
                FE = FE + 1.
                Select best individual from parent and offspring to enter new generation.
            else
                Select parent into new generation.
        • N individuals enter into new generation.
```

In addition to modeling techniques, there also exists several model management strategies in SAEAs. In our experiments, we choose two typical strategies: *individual-based evolution control* and *pre-selection*. In individual-based evolution control, the number of individuals being re-evaluated

by actual fitness functions are determined by parameter p_{sm} for an absolute fitness model, and τ_{th} for a rank-based model, in which τ is the Kendall's coefficient [20]. For pre-selection strategy, the parameter generator factor, i.e. λ_{pre}, determines how many offsprings are generated by mutation and crossover operators for an absolute fitness surrogate model and a rank-based model. The classification-based model is different. Only one mechanism is used to assist evolutionary algorithms [21] which can be regarded as the pre-selection strategy but without any generator factor.

Finally, a genetic algorithm (GA) is selected as the basic EA to be combined with a surrogate model and model management strategies. The pseudocodes of GA assisted with different models and model management strategies are presented in Algorithms 1 and 2.

2) Benchmark functions and running condition: The test functions used in the experiments are selected from CEC2015 expensive optimization test problems [68]. We only choose several typical functions from the benchmark, and the details of them are shown in Table I.

In all SAEA instances, the surrogate model is used as a local model, that is, a model is constructed for each new individual. In our experiments, we execute each algorithm instance with 25 independent runs and the number of fitness evaluations for each run are set as $200d$, where d denotes the dimensionality of the test function. A LHS with $5d$ samples is used for initialization to collect the training data set for each SAEA in the experiments. Due to the time limitation, we only compared four selected models only in the cases of $d = 10$ and $d = 20$. The population size N is set to 40 in all experiments. The percentage of crossover pc is 0.7, and mutation rate pm is 0.3.

TABLE I: The details of benchmark functions [68]. F^* denotes the optimal fitness of the test function.

Problem	Modality	Dimensionality	F^*
f_1	Unimodal	10,20	100
f_2			200
f_4	Simple Multimodal		400
f_8			800
f_{13}	Composition Function		1300
f_{15}			1500

B. Analysis 1: Computation Time

Firstly, we analyse the time complexity of each investigated surrogate model. We recorded the execution time from choosing the training data to predicting fitness values or ranking all offsprings. The experiments are executed in a computer of Intel Core i5 CPU with 3.2 GHz and 8 GB RAM. And the sizes of training data are set to $5d$ for all models. The average running times for training and using the surrogate model in different problems and dimensionalities are presented in Table II

The KNN model is the most efficient surrogate model. It is because that KNN is a lazy model which doesn't learn a discriminative function from the training data but "memorizes" the training data set instead. The only computational burden is to calculate the similarity between a new individual with training data and predict the fitness using the similarities.

TABLE II: The average running time for training and using the surrogate model in $d = 10, 20$. (Unit: *ms*)

D	Problem	RBF	KNN	RankSVM	SVC
10	f_1	0.957	**0.280**	7.852	4.436
	f_2	0.974	**0.280**	6.319	3.165
	f_4	0.966	**0.279**	6.825	3.341
	f_8	0.952	**0.277**	7.526	4.039
	f_{13}	0.957	**0.279**	7.754	4.146
	f_{15}	0.953	**0.278**	7.644	4.323
20	f_1	2.265	**0.383**	65.923	5.184
	f_2	2.279	**0.394**	92.022	3.478
	f_4	2.305	**0.400**	90.002	4.881
	f_8	2.254	**0.384**	81.653	5.006
	f_{13}	2.253	**0.386**	79.875	5.231
	f_{15}	2.192	**0.383**	77.685	5.247

The RBF model is also very efficient. In the procedure of training a RBF model, the distance matrix between training data points and its inversion need to be computed, which brings computational cost. The SVC model is a little worse than the RBF model. The main procedure in the SVC model is the quadratic programming optimization process, which is time-consuming.

RankSVM is the most expensive surrogate model among these four models, because of the pairwise operation both in training and testing procedures. The most important point is the sharply increasing of training data to the SVM after the pairwise operation, which directly causes the increase of training time. As a result, the computational time for RankSVM is much more than other three models.

C. Analysis 2: Model Accuracy

The model accuracy usually determines the performance of surrogate models in SAEA. Therefore, we analyze the accuracy of different surrogate models in this subsection. Existing surveys compared the coefficient of determination (R^2) for different kinds of surrogate models [46][49]. However, the rank correlation is much more important than the regression accuracy in evolutionary algorithms. So, we use rank correlation as the accuracy metric for comparing four surrogate models.

In each generation, N parents generates N offsprings by variation operators, and the whole population is denoted as $P = \{Ind_i | i = 1, 2, ..., 2N\}$. The new population selected based on true fitness function evaluation is denoted as $P_f = \{Ind_i | i = \lambda_1, \lambda_2, ..., \lambda_N\}$. And the new population selected with the use of surrogate model is denoted as $P_m = \{Ind_i | i = \mu_1, \mu_2, ..., \mu_N\}$. Then, the rank correlation of the model is calculated by Eq. (5).

$$Corr = \frac{|P_m \cap P_f|}{N} \tag{5}$$

The metric $Corr$ is calculated for each surrogate model in each generation and the results are showed in Table III. The value in each cell is the average value of all generations.

TABLE III: The accuracy (rank correlation) of surrogate model in $d = 10, 20$.

D	Problem	RBF	KNN	RankSVM	SVC
10	f_1	0.836	0.676	**0.937**	0.489
	f_2	**0.921**	0.915	0.585	0.466
	f_4	**0.863**	0.843	0.574	0.501
	f_8	**0.820**	0.758	0.709	0.518
	f_{13}	**0.801**	0.751	0.690	0.539
	f_{15}	**0.826**	0.737	0.760	0.544
20	f_1	0.908	0.600	**0.938**	0.472
	f_2	**0.953**	0.950	0.550	0.460
	f_4	**0.895**	0.829	0.621	0.614
	f_8	**0.852**	0.786	0.707	0.573
	f_{13}	**0.830**	0.655	0.829	0.532
	f_{15}	**0.875**	0.738	0.740	0.575

From Table III, we can easily find that the ranking of model accuracy of the investigated models is:

$$RBF \succ KNN \approx RankSVM \succ SVC$$

Overall, the absolute fitness model is better than the relative fitness model with respect to the rank correlation, although the relative fitness model intuitively should be better than the absolute fitness model. It is probably because the absolute fitness model aims at approximating the fitness landscape, and if the approximation is accurate enough, the ranking correlation is also pretty accurate. From this aspect, we can claim that the RBF model is better than the KNN model with respect to R^2. The RankSVM model performed relatively well in general, comparable to the KNN model in the sense that sometimes it is better and sometimes worse over different benchmark functions. In our experiments, the SVC model has the lowest rank correlation.

The accuracy of a surrogate model directly influences the performance of SAEA. However, the impact of model accuracy on SAEAs' performance varies with different model management strategies, that is much greater on the individual-based evolution control than a pre-selection strategy. This is because a low model accuracy will cause many low-quality individuals to survive to the new generation and high-quality individuals being obsoleted in the individual-based control strategy. For pre-selection strategy, the low-quality will not lead SAEA into a false optimum but only deteriorate the convergence speed of SAEA, because all individuals entering into the selection operator have actual fitness. Therefore, even though the SVC's accuracy is not outstanding, the SAEA assisted by SVC can obtain an acceptable performance. However, its performance will be worse than other models with pre-selection strategy.

D. Analysis 3: Parameter Influence

In SAEAs, the use of surrogate models brings in some new parameters, which have effect on the performance of SAEAs. How to determine these newly introduced parameters is an important issue in designing SAEAs. However, there is almost no comprehensive study on this issue in the literature. In [51], authors only analyzed the effect of database size and neighbourhood size in the kNN model. In [49], authors studied the influence of training samples' size for local and global models, but it only focused on regression-based models. In order to have a comprehensive understanding about the parameters' influence of different kinds of surrogate models on SAEAs' performance, we conducted a series of experiments in this subsection.

We studied two important parameters for each surrogate model. For the individual-based evolution control, one is the size of training data K, and another is the parameter with respect to the re-evaluation ratio p_{sm}. In the rank-based model, the threshold of rank correlation, τ_{th}, determines the re-evaluation ratio, which re-evaluates the individual until updated model accuracy satisfies a threshold. For the pre-selection strategy, one parameter is the size of training data K, and another is the generator factor λ_{pre} that determines how many offsprings are generated for one pair of parents. We employed the grid search method to study the parameters' influence. SAEAs and parameter's settings we studied in our experiments are listed in Table IV.

TABLE IV: The details of SAEA and their parameters.

Model Management Strategy	Surrogate Model	SAEA	Parameters
Individual-based Evolution Control	RBF	IB-RBF	K, p_{sm}
	KNN	IB-KNN	K, p_{sm}
	RankSVM	IB-Rank	K, τ_{th}
Pre-Selection Strategy	RBF	PS-RBF	K, λ_{pre}
	KNN	PS-KNN	K, λ_{pre}
	RankSVM	PS-Rrank	K, λ_{pre}
	SVC	PS-SVC	K
K	$\{i \times d \mid i = 1, 2, \ldots, 10\}$		
p_{sm}, τ_{th}	$\{0.05(2i-1) \mid i = 1, 2, \ldots, 10\}$		
λ_{pre}	$\{i + 1 \mid i = 1, 2, \ldots, 10\}$		

The experiments are executed on six different functions with the dimension equals to 10 and 20. In order to study the overall performance and to avoid the difference of problems' scale, we employ another performance metric from literature [69]. For each SAEA, denoted as A, we have test functions $\mathcal{F} = \{f_k | k = 1, 2, ..., n\}$ and parameter settings $\mathcal{C} = \{c_j | j = 1, 2, ..., m\}$. The performance of each parameter setting $PM(c_i)$ in test functions can be calculated by Eq. (6):

$$PM(c_i) = \frac{1}{m-1} \sum_{j=1, j \neq i}^{m} P(A_{c_i} > A_{c_j}) \tag{6}$$

The $P(A_{c_i} > A_{c_j})$ represents the probability of A_{c_i} outperforming A_{c_j}, which can be calculated by Eq. (7):

$$P(A_{c_i} > A_{c_j}) = \frac{1}{n} \sum_{k=1}^{n} P(q_{i,k} < q_{j,k} | f_k) \tag{7}$$

where $q_{i,k}$ denotes the quality of solution obtained by A_{c_i} on f_k. And $P(q_{i,k} < q_{j,k} | f_k)$ can be estimated by the following

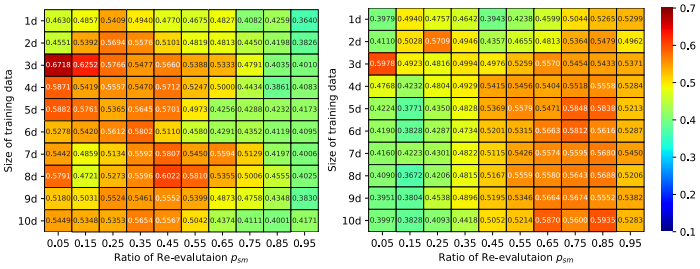
equation:

$$P(q_{i,k} < q_{j,k}|f_k) = \frac{\sum_{s=1}^{s_i}\sum_{t=1}^{s_j}\mathbb{I}(y_{i,k,s} < y_{j,k,t})}{s_i \times s_j} \quad (8)$$

where $y_{i,k,s}$ represents the fitness of solution obtained by A_{c_i} in s_{th} trial for f_k problem. s_i, s_j represent the number of trails of each algorithm for one problem. $\mathbb{I}(\cdot)$ denotes the indicator function.

The experimental results of different SAEAs using different surrogate models and parameter settings are presented in Figures 6 $\sim$ 12. The number in each cell of heatmap represents the performance value $PM(c_i)$ of each parameter configuration.

1) IB-RBF: For the RBF model with individual-based model control (IB-RBF) in Figure 6, a large p_{sm} value gets a worse performance when $d = 10$, but large p_{sm} for $d = 20$ makes the performance much better. It might be because the quality of RBF model is higher in the case of $d = 10$. A small p_{sm} means more generations. And when the surrogate model is accurate enough, more generations make the SAEA obtain better convergence. On the contrary, when the surrogate model is not accurate, more generations probably would lead SAEA to a false optimum so that the re-evaluation is much more important. Therefore, a high p_{sm} can get a better performance in $d = 20$ of IB-RBF. The size of training data K by contrast makes little impact on IB-RBF's performance. According to the result, $3d$ to $4d$ would be the most suitable value for IB-RBF.



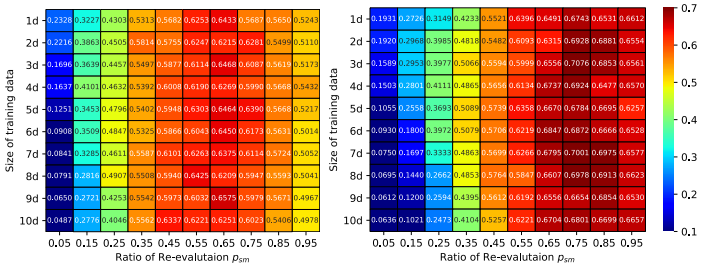
a) $d = 10$ b) $d = 20$

Fig. 6: Parameter influence in IB-RBF with $d = 10, 20$.

2) IB-KNN: For the KNN model with individual-based model control (IB-KNN), we can easily find that the size of training data almost has no influence on SAEA's performance for $d = 10$ and $d = 20$, as shown in Figure 7. We can analyze the reason from the predictive model of KNN as Eq. (9):

$$\hat{F}(\mathbf{x}) = \frac{\sum_{j=1}^{k} s\left(\mathbf{x}, \mathbf{x}_j\right)^2 F\left(\mathbf{x}_j\right)}{\sum_{j=1}^{k} s\left(\mathbf{x}, \mathbf{x}_j\right)^2} \quad (9)$$

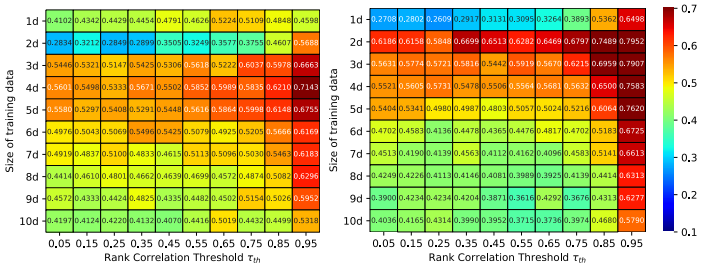
where $s\left(\mathbf{x}, \mathbf{x}^{p_i}\right) = 1 - \frac{d(\mathbf{x}, \mathbf{x}^{p_i})}{d(\mathbf{x}^U, \mathbf{x}^L)}$ and $d\left(\mathbf{x}, \mathbf{y}\right)$ is the distance between $\mathbf{x}$ and $\mathbf{y}$. For a large K value of KNN model, the $s\left(\mathbf{x}, \mathbf{x}^{p_i}\right)$ between the new solution and the farthest point is very small and close to zero. Therefore, the points far away from the new solution can not make much difference to the predictive result. As a result, SAEAs with the small and large k value in the KNN model obtain a similar performance. The p_{sm} setting as 0.55 $\sim$ 0.75 is the most suitable for SAEA's final performance for the KNN model both for $d = 10$ and $d = 20$. This is because a small p_{sm} is likely to lead the SAEA to the false optima and a large p_{sm} causes the decrease of generation. $p_{sm} = 0.55 \sim 0.75$ makes balance between the model accuracy (avoid to leading to the false optimum) and the number of evolution generations.



a) $d = 10$ b) $d = 20$

Fig. 7: Parameter influence in IB-KNN with $d = 10, 20$.

3) IB-Rank: For the RankSVM model with individual-based model control (IB-Rank) in Figure 8, the influence of training data size K on SAEA's performance is more significant than that of rank correlation threshold τ_{th}. From the result, the K setting as $3d \sim 4d$ is the best choice. When K is small, the training data is not sufficient for training an accurate enough rank model. On the other hand, the Kendall's τ is computed using the relative ordering of the ranks of all $K(K-1)/2$ possible pairs [20]. Therefore, when training data is too much, the Kendall's τ will be small if the rank model is not accurate enough so that the generation of evolutionary will decrease due to the more re-evaluations in each iteration. As a result, a moderate value of K is most suitable for the rank-based model. For rank correlation threshold τ_{th}, a large value is better because it means a more accurate rank-based model. From the results, τ_{th} setting as bigger than 0.95 might be a suitable choice both for $d = 10$ and $d = 20$.

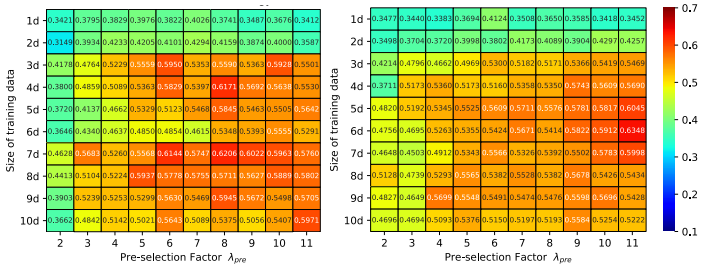


a) $d = 10$ b) $d = 20$

Fig. 8: Parameter influence in IB-Rank with $d = 10, 20$.

4) PS-RBF: For the RBF model with pre-selection strategy (PS-RBF) in Figure 9, the upper part of the heatmap is shallower than the lower part in $d = 10$ as well as $d = 20$. It indicates a bigger training data size brings on a better performance of SAEA with PS-RBF. It is because more data can make the RBF model more accurate and the pre-selection can select potentially better solutions. A K of $3d \sim 4d$ is enough because more data than $4d$ may not increase the quality of RBF any more. For pre-selection factor λ_{pre}, a bigger value

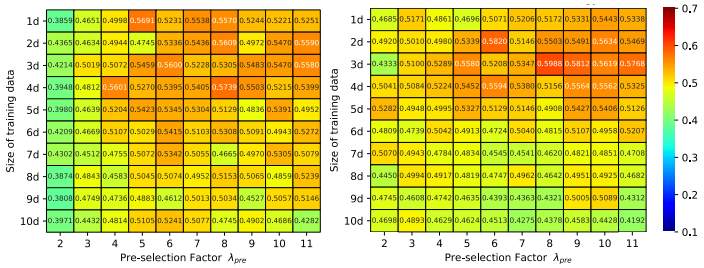
brings on better performance, but an increased λ_{pre} doesn't obtain obvious performance increase. From the results, the pre-selection factor setting as 8 is enough for PS-RBF.



a) $d = 10$ b) $d = 20$

Fig. 9: Parameter influence in PS-RBF with $d = 10, 20$.

5) PS-KNN: For the KNN model with pre-selection strategy (PS-KNN), the difference between different parameter settings is very small as shown in Figure 10. In the case of $d = 10$, the performance of SAEA is slightly improved when λ_{pre} increases. It is probably because the number of offspring generated by mutation or crossover operators increases with the increase of λ_{pre} and it makes the search diversity increase. And when $d = 20$, the recommended parameter setting for PS-KNN would be a small K and a large λ_{pre}, even though the performance has no big difference.

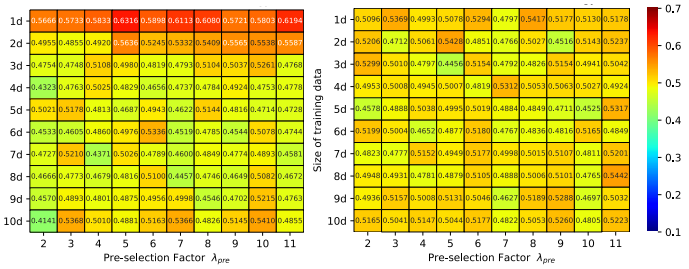


a) $d = 10$ b) $d = 20$

Fig. 10: Parameter influence in PS-KNN with $d = 10, 20$.

6) PS-Rank: For the RankSVM model with pre-selection strategy (PS-Rank) in Figure 11, the performance of different parameters is also very similar to each other, especially in the case of $d = 20$. For $d = 10$, a small K is more suitable than others for SAEA's performance. The main reason for the similar performance is the small value of Kendall's τ. The training data for each rank-based model will be $K \cdot \lambda_{pre}$, and the τ will be computed using the relative ordering of the ranks of all $K \cdot \lambda_{pre}(K \cdot \lambda_{pre} - 1)/2$ possible pair, so that the τ will be very small in most cases. Therefore, the PS-Rank will randomly select the candidate as the offspring in most cases. The individuals to be ranked in PS-Rank are generated from the same parents so that these individuals might be close to each other. Therefore, a small number of training data can distinguish these individuals more precisely. For λ_{pre}, it almost has no influence on the performance of PS-Rank.

7) PS-SVC: For the SVC model with pre-selection strategy (PS-SVC), due to its special algorithm structure, only one



a) $d = 10$ b) $d = 20$

Fig. 11: Parameter influence in PS-Rank with $d = 10, 20$.

hyper-parameter has an effect on SAEA's performance. Therefore, we only analyzed the effect of training data size in Figure 12. It is very clear that the training data size setting as $6d$ is enough in the case of $d = 10$. SAEA's performance will not be improved a lot even though K continues to increase. The training data size almost has no influence on SAEA's performance when $d = 20$ if we consider the algorithm randomness. The likely reason is that the limited and imbalanced training data cannot satisfy the requirement for training an efficient SVC model in the case of $d = 20$.

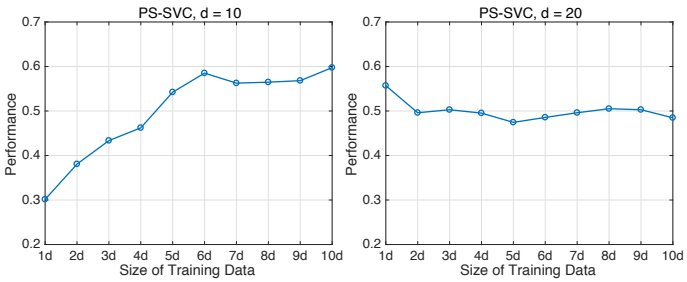


Fig. 12: Parameter influence in PS-SVC with $d = 10, 20$.

E. Analysis 4: Performance Comparison

In this subsection, we will analyze the efficacy of different kind of surrogate models by comparing SAEA's performance on test functions. For fair comparison, we compare surrogate models' performance in SAEA with its best parameter configuration obtained from the previous experiments.

1) Overall performance: The results are presented in Table V. Numbers in each cell denote averaged best fitness and standard deviation. The bold ones are the best results among all models on one problem. The ranking of each algorithm instance is calculated on the basis of statistical hypothesis tests (i.e. Wilcoxon test with a 0.05 significance level), in which if two results have no significant difference according to the hypothesis test, they will be applied to the same ranking. The number in brackets of each cell represents the algorithm instance's ranking. Besides, the average ranking of algorithm instances for $d = 10$ and $d = 20$ are also provided in the table, respectively.

First of all, it is obvious that PS-RBF and IB-RBF assisted SAEAs perform best among all algorithm instances on test functions according to the average ranking. It indicates the RBF model is still the best model in the absolute and relative

TABLE V: The results of comparing algorithm instances on test functions over 25 runs including the average best fitness and standard deviation shown as AVR ± STD. The boldface figures are the best fitness among all algorithm instances for each test function according to the Wilcoxon test with a 0.05 significance level.

D	Problem	PS-RBF	PS-KNN	PS-Rank	PS-SVC
10	f1	**6.3189e+03 ± 2.7100e+03 (1)**	**6.2222e+03 ± 3.0985e+03 (1)**	1.3269e+05 ± 1.0201e+05 (4)	5.4414e+05 ± 1.3998e+06 (4)
	f2	**1.9534e+04 ± 7.8264e+03 (1)**	2.1284e+04 ± 8.9185e+03 (3)	2.2319e+04 ± 7.4356e+03 (3)	2.5320e+04 ± 8.5496e+03 (5)
	f4	**6.7013e+02 ± 2.0807e+02 (1)**	6.6177e+02 ± 1.5887e+02 (2)	8.3013e+02 ± 2.2133e+02 (5)	9.5162e+02 ± 1.9861e+02 (6)
	f8	8.0299e+02 ± 8.5518e-01 (4)	8.0247e+02 ± 8.2028e-01 (3)	8.0352e+02 ± 6.9876e-01 (5)	8.0415e+02 ± 8.2992e-01 (6)
	f13	**1.6178e+03 ± 1.3178e+00 (1)**	1.6197e+03 ± 3.0017e+00 (5)	1.6189e+03 ± 1.6564e+00 (3)	1.6235e+03 ± 4.5621e+00 (8)
	f15	**1.6750e+03 ± 1.7918e+02 (1)**	1.8295e+03 ± 7.0078e+01 (3)	1.6913e+03 ± 1.8190e+02 (2)	1.8804e+03 ± 9.5120e+01 (6)
Average-Ranking ($d = 10$)		1.5	2.83	3.67	5.83
20	f1	**3.3580e+07 ± 2.7307e+07 (1)**	4.1484e+07 ± 2.5689e+07 (2)	6.3646e+07 ± 4.5747e+07 (3)	1.3027e+08 ± 1.1491e+08 (6)
	f2	3.2654e+04 ± 7.0460e+03 (2)	3.5837e+04 ± 8.0113e+03 (4)	3.5188e+04 ± 8.6006e+03 (3)	3.8394e+04 ± 8.9980e+03 (5)
	f4	1.2969e+03 ± 3.7046e+02 (2)	1.3536e+03 ± 3.4626e+02 (3)	1.6349e+03 ± 4.9787e+02 (4)	2.1249e+03 ± 5.7958e+02 (6)
	f8	**8.0852e+02 ± 3.2504e+00 (1)**	8.0886e+02 ± 3.4550e+00 (2)	8.1621e+02 ± 8.5271e+00 (4)	8.1727e+02 ± 4.5242e+00 (6)
	f13	**1.5958e+03 ± 5.9549e+00 (1)**	1.5998e+03 ± 9.1526e+00 (4)	1.5990e+03 ± 6.9405e+00 (3)	1.6070e+03 ± 1.5230e+01 (6)
	f15	**1.9238e+03 ± 3.5793e+01 (1)**	1.9449e+03 ± 4.1802e+01 (2)	1.9652e+03 ± 3.7121e+01 (4)	2.0253e+03 ± 4.0137e+01 (6)
Average-Ranking ($d = 20$)		1.33	2.83	3.5	5.83
Average-Ranking		1.42	2.83	3.58	5.83
D	Problem	IB-RBF	IB-KNN	IB-Rank	NS
10	f1	7.8707e+03 ± 5.3762e+03 (2)	3.5295e+04 ± 1.0491e+05 (3)	9.4191e+03 ± 1.0561e+04 (2)	8.8084e+05 ± 8.8456e+05 (5)
	f2	**1.9742e+04 ± 6.3255e+03 (1)**	2.1131e+04 ± 8.1986e+03 (2)	2.5569e+04 ± 1.0216e+04 (4)	3.0945e+04 ± 1.1908e+04 (6)
	f4	**5.7690e+02 ± 1.0756e+02 (1)**	6.4679e+02 ± 1.3257e+02 (2)	7.3147e+02 ± 1.6015e+02 (3)	7.5388e+02 ± 1.4102e+02 (4)
	f8	**8.0180e+02 ± 9.8822e-01 (1)**	**8.0190e+02 ± 1.0296e+00 (1)**	8.0210e+02 ± 1.0909e+00 (2)	8.0403e+02 ± 8.4194e-01 (6)
	f13	1.6184e+03 ± 1.3885e+00 (2)	1.6189e+03 ± 1.7946e+00 (4)	1.6202e+03 ± 2.2969e+00 (7)	1.6204e+03 ± 2.9060e+00 (6)
	f15	**1.6409e+03 ± 1.6444e+02 (1)**	1.8809e+03 ± 4.6203e+01 (5)	1.9048e+03 ± 8.3818e+01 (7)	1.7931e+03 ± 1.9014e+02 (4)
Average-Ranking ($d = 10$)		1.33	2.83	4.17	5.17
20	f1	6.1769e+07 ± 5.3520e+07 (3)	8.9753e+07 ± 1.2423e+08 (4)	1.3174e+08 ± 1.0504e+08 (6)	1.0890e+08 ± 9.5198e+07 (5)
	f2	**3.1528e+04 ± 7.6819e+03 (1)**	3.5311e+04 ± 1.2032e+04 (3)	3.9266e+04 ± 1.3019e+04 (5)	3.8990e+04 ± 8.1845e+03 (5)
	f4	**1.1292e+03 ± 3.8573e+02 (1)**	1.3336e+03 ± 4.6006e+02 (2)	1.8292e+03 ± 5.0367e+02 (5)	1.9917e+03 ± 4.2165e+02 (6)
	f8	8.0917e+02 ± 3.9245e+00 (2)	8.1215e+02 ± 5.9572e+00 (3)	8.6065e+02 ± 5.8219e+01 (7)	8.1685e+02 ± 4.9504e+00 (5)
	f13	1.5972e+03 ± 6.1122e+00 (2)	1.5991e+03 ± 8.8844e+00 (3)	1.6113e+03 ± 2.1264e+01 (4)	1.6041e+03 ± 1.5099e+01 (5)
	f15	1.9368e+03 ± 2.2315e+01 (2)	1.9600e+03 ± 4.8161e+01 (3)	2.0581e+03 ± 7.5934e+01 (6)	1.9873e+03 ± 4.1149e+01 (5)
Average-Ranking ($d = 20$)		1.83	3	5.5	5.17
Average-Ranking		1.58	2.92	4.83	5.17

fitness model. Then the KNN model is the second best surrogate model in the table, the ranking of PS-KNN and IB-KNN follows the RBF model. The RandkSVM and SVC models perform much worse than the previous two kinds of models, where the SVC model is the worst in our experiments. Overall, we can get a basic conclusion:

$$\mathrm{RBF} \succ \mathrm{KNN} \succ \mathrm{RankSVM} \succ \mathrm{SVC}$$

where $M_1 \succ M_2$ represents M_1 performs better than M_2.

For one model with two different model management strategies, we can find that the pre-selection strategy is slightly better than the individual-based evolution control. As discussed in Section 2, a pre-selection strategy reduces the computational cost by improving the quality of individuals and it can ensure the optimization converge to the right solution. The individual-based model cannot guarantee the right direction of optimization process. However, the number of generations of individual-based evolution control strategy is much more than the pre-selection strategy. Therefore, in most cases, a pre-selection strategy is better than individual-based evolution control while in some cases, the evolution control with a high quality surrogate model can obtain much better performance thanks to more generations.

The relative performance for each model in $d = 10$ and $d = 20$ has no big difference. RBF is the best model in the case of $d = 10$ as well as $d = 20$ and SVC is the worst model both in $d = 10$ and $d = 20$.

2) Discussion for each model: We plot evolutionary curves of each algorithm instance on all test functions as shown in Figure 14. Together with the numerical results and evolutionary curves, we will discuss the characteristic of each model from the SAEA's performance in this subsection.

As analyzed before, the RBF model is the best model among four kinds of surrogate models. On almost all test functions, RBF obtains the best result. From the evolutionary curves, we can find that PS-RBF and IB-RBF show a steady convergence. In most cases, PS-RBF and IB-RBF converge faster than others during the whole optimization process. We can conclude that the RBF model is a good surrogate model for SAEA not only because of its good final results but also the outstanding evolutionary performance. The main reason for RBF model's excellent performance is its high accuracy.

The KNN model also obtains excellent performance after RBF model. However, we can find in the result table that KNN model's performance deteriorates for composite functions, i.e. f_{13} and f_{15}. It is because the composite function is highly non-linear so that the accuracy of KNN model decreased, which can be found in Table III. Composite functions have many local optimas, where the algorithm is easily trapped.

For the RankSVM surrogate model, its performance is better

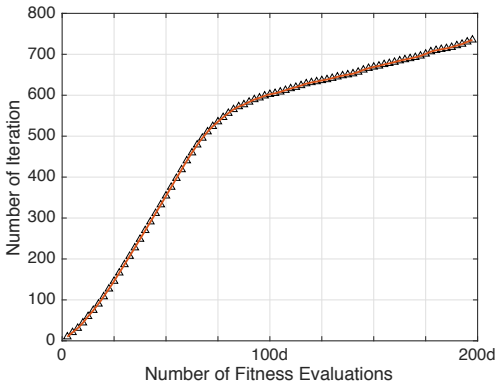


Fig. 13: The increase of number of generations during the optimization process in f_1 with $d = 20$.

than the evolutionary algorithm without any surrogate model in our experiments, but its performance is still limited and inferior to RBF and KNN models obviously. Different from previous two models, the PS-Rank and IB-Rank have a big difference that PS-Rank is much better than IB-Rank. It is probably because the accuracy of obtained RankSVM model is low and it influences individual-based evolution control much more than the pre-selection strategy. As a result, the performance of RankSVM is not good aalthough PS-Rank is better than IB-Rank. However, we find that IB-Rank converges very fast in the early stage of optimization but the final results are not good on almost all functions according to Figure 14. We record the number of generations every 25 fitness evaluations during the whole optimization process, which is plotted in Figure 13. We can find that the same number of fitness evaluations feed fewer iterations in the later stage of optimization process. As a result, the performance of IB-Rank is decreased a lot because of the lack of generations.

The SVC model is the worst model among four models and its performance is even worse than evolutionary algorithms without any surrogate models in our experiments. Compared with other models with a pre-selection strategy, PS-SVC doesn't generate additional offsprings during mutation and crossover so that the search diversity is much poorer than other models with pre-selection strategies. From evolutionary curves, the SVC model assisted SAEA also obtain the good convergence at the early stage but its speed slows down very quickly. This is probably because of the decrease of SVC model's accuracy in the later stage of optimization as showed in Figure 15, in which we plot the change of accuracy in the optimization process.

It is worth mentioning that the performance of SAEA with PS-SVC is even worse than evolutionary algorithms without any surrogate models. The problem of poor diversity is one reason, but the most important is that the SVC model is not suitable in our evolutionary algorithm framework. In the first work of classification-based model [21], authors used SVC in a DE algorithm, in which one parent generates one offspring and the new parent is selected between these two individuals, which we named it as parent offspring tournament selection (POTS). But in our experiment, the basic framework for selection is to choose the better individuals in the individual pool which includes parent and offspring population. Furthermore, we compared the performance of SVC assisted SAEA with POTS based none-surrogate EA (POTS-NSEA) in $d = 10$. The results are showed in Figure 16, where the blue box are SVC-SAEA and orange box are POTS-NSEA. In this case, the SVC-SAEA performs better than EA without any surrogate models in almost all functions. Therefore, we can conclude that the classification model is more suitable in POTS based evolutionary algorithms.

F. Summary

From the above analyses, we can obtain a comprehensive conclusion about four surrogate models. The RBF model is the most effective surrogate model which has low computation time, high accuracy and the excellent performance in SAEA. Thus, RBF is recommended for SAEAs to solve single-objective CEPs. KNN model is the most efficient model with respect to computation time and the corresponding SAEA also obtain good performance. RankSVM model and SVC model are not very effective compared with previous two models. These two models are more effective in the specific algorithm structure. It is the main reason that the SAEA with a relative fitness model is very limited in the literature for single-objective optimization.

VI. Conclusion and Future work

In this paper, we give a comprehensive taxonomy and comparison of surrogate models used in SAEAs for single-objective optimization problems.. Different from existing surveys, we propose a new and more systematic taxonomy that divides existing surrogate models in SAEAs into two main classes: absolute and relative fitness approximate models.

Absolute fitness models provide approximate fitness values of new individuals to replace the actual expensive fitness evaluation. According to modeling approaches, absolute fitness models can be further divided into regression-based and similarity-based models. Regression-based models construct a mathematical function to model the true fitness function. It has a higher accuracy in approximating the fitness function. Besides, regression-based models could provide overall information about the actual fitness function, which is not only valuable for optimization but also for problem analysis. The similarity-based model derives the fitness of un-evaluated new individual by measuring the correlation between the new individual and its neighbors. It is the most convenient among the four kinds of models, but their accuracy cannot be guaranteed, especially when the fitness landscape is complex.

Relative fitness approximate models have recently been introduced to SAEAs. This relatively new kind of methods focuses on the preference or rank of individuals rather than their absolute fitness values. Relative fitness models could be categorized into two sub-classes: rank-based and classification-based surrogate models. Rank-based models provide the rank information of the current population for selection, and classification-based methods compare individuals with one reference solution. These two models are both specifically

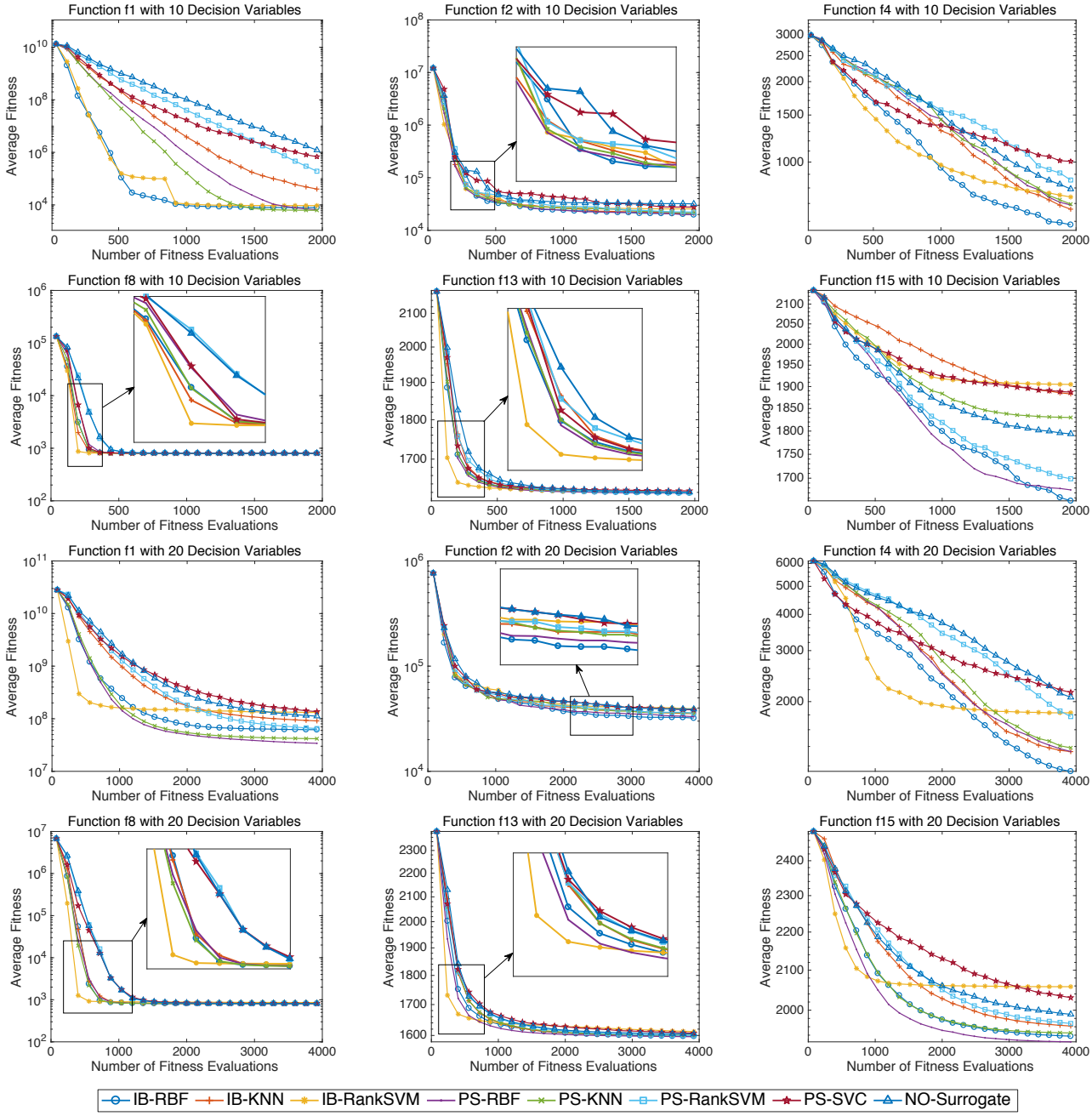


Fig. 14: The convergence curves of algorithm instances on test functions with $d = 10, 20$.

designed for EAs that only use the rank information in selection. Due to the special mechanism, they are only used in limited works.

Comparison studies of different surrogate models are conducted from different perspectives, including computation time, model accuracy, parameter influence, and the performance of SAEAs. We selected RBF, KNN, RankSVM and SVC as the representative surrogate models and embedded them into the general evolutionary algorithm, i.e. genetic algorithm. Firstly, the compuation time is compared, KNN and RBF models are more efficient than RankSVM and SVC model while RankSVM is the most time-consuming model. Then, the model accuracy are compared by using rank correlation as the metric. We find that the RBF model is more accurate than other three surrogate models on most test functions. Finally, we compared their efficacy by running the SAEA assisted by different models and model management strategies. The results indicate that the pre-selection strategy is better than individual-based evolution control given the same surrogate model, and RBF model is still the most effective surrogate model among absolute and relative fitness models. Additionally, we also find that the SVC model is not suitable for all evolutionary algorithms.

For each surrogate model we used in our experiment, we explore the influence of some important model parameters. The size of training data K does not make a big difference

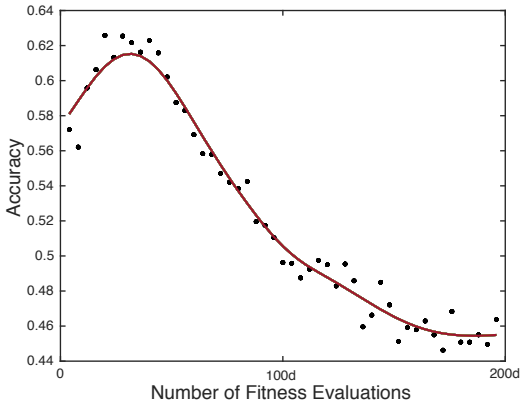


Fig. 15: The dynamical change of average accuracy for SVC model.

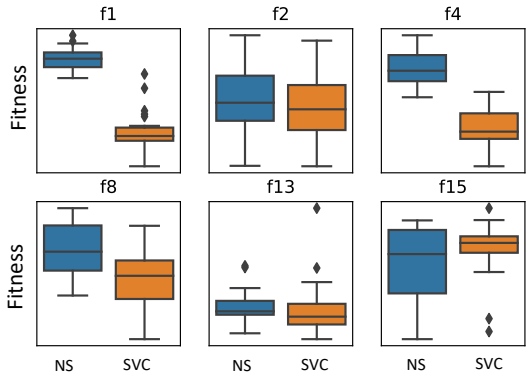


Fig. 16: The comparison between SVC-SAEA and POTS-NSEA in $d = 10$

to the SAEA's performance of RBF and KNN models, but it is very important to the RankSVM and SVC models. For individual-based evolution control, the parameter determining the ratio of re-evaluation is important to all surrogate models, that low p_{sm} is suitable for the RBF model, moderate p_{sm} is suitable for the KNN model and large τ_{th} is suitable for the RankSVM model. For the pre-selection strategy, the preselection factor's influence is not very obvious, and the results suggest a moderate λ_{pre} is enough for all surrogate models.

The surrogate models used in single-objective evolutionary algorithm are reviewed and compared in this paper. For future work, it is well worth to review the surrogate models applied in multi-objective optimization, especially for the relative fitness surrogate model. Although there have been some reviews about metamodels in multi-objective evolutionary algorithm [8][9][10], it still has no work to compare different kind of models by experimental analysis, and it also has no research to study the parameter influence for different models in multi-objective evolutionary algorithms. Therefore, it is valuable to do some analysis in this topic.

Finally, we also find, in the process of literature review, most work in SAEAs is concerned with unconstrained single objective optimization. Yet many real world problems have constraints and multiple objectives. They may be dynamic and contain uncertainty. There has been very few work in SAEAs for constrained optimization, multi-objective optimization and dynamic optimization. This is clearly an interesting direction for future research.

ACKNOWLEDGEMENT

This work was supported by the Program for Guangdong Introducing Innovative and Enterpreneurial Teams (Grant No. 2017ZT07X386), Shenzhen Science and Technology Program (Grant No. KQTD2016112514355531), and the Program for University Key Laboratory of Guangdong Province (Grant No. 2017KSYS008).

REFERENCES

[1] Y.S. Ong, P.B. Nair, and A.J. Keane, "Evolutionary optimization of computationally expensive problems via surrogate modeling," *AIAA Journal*, vol. 41, no. 4, pp. 687–696, 2003.

[2] B. Glaz, T. Goel, L. Liu, P.P. Friedmann, and R.T. Haftka, "Multiple-surrogate approach to helicopter rotor blade vibration reduction," *AIAA Journal*, vol. 47, no. 1, pp. 271–282, 2009.

[3] J.J. Grefenstette and J.M. Fitzpatrick, "Genetic search with approximate function evaluations," in *Proceedings of an International Conference on Genetic Algorithms and Their Applications*, 1985, pp. 112–120.

[4] Y. Jin, "A comprehensive survey of fitness approximation in evolutionary computation," *Soft Computing*, vol. 9, no. 1, pp. 3–12, 2005.

[5] D. Lim, Y. Jin, Y.S. Ong, and B. Sendhoff, "Generalizing surrogate-assisted evolutionary computation," *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 3, pp. 329–355, 2010.

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

[7] A. Díaz-Manríquez, G. Toscano, J.H. Barron-Zambrano, and E. Tello-Leal, "A review of surrogate assisted multiobjective evolutionary algorithms," *Computational Intelligence and Neuroscience*, vol. 2016, pp. 9 420 460:1–9 420 460:14, 2016.

[8] T. Chugh, K. Sindhya, J. Hakanen, and K. Miettinen, "A survey on handling computationally expensive multiobjective optimization problems with evolutionary algorithms," *Soft Computing*, vol. 23, no. 9, pp. 3137–3166, 2019.

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

[10] K. Deb, R. Hussein, P.C. Roy, and G. Toscanopulido, "A taxonomy for metamodeling frameworks for evolutionary multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 1, pp. 104–116, 2019.

[11] T. Bartz-Beielstein and M. Zaefferer, "Model-based methods for continuous and discrete global optimization," *Applied Soft Computing*, vol. 55, pp. 154–167, 2017.

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

[13] X. Cai, L. Gao, and F. Li, "Sequential approximation optimization assisted particle swarm optimization for expensive problems," *Applied Soft Computing*, vol. 83, p. 105659, 2019.

[14] V. Christelis and A. Mantoglou, "Pumping optimization of coastal aquifers assisted by adaptive metamodelling methods and radial basis functions," *Water Resources Management*, vol. 30, no. 15, pp. 5845–5859, 2016.

[15] R.G. Regis, "Particle swarm with radial basis function surrogates for expensive black-box optimization," *Journal of Computational Science*, vol. 5, no. 1, pp. 12–23, 2014.

[16] H. Wang, Y. Jin, and J.O. Jansen, "Data-driven surrogate-assisted multiobjective evolutionary optimization of a trauma system," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 6, pp. 939–952, 2016.

[17] X. Sun, D. Gong, Y. Jin, and S. Chen, "A new surrogate-assisted interactive genetic algorithm with weighted semisupervised learning," *IEEE Transactions on Cybernetics*, vol. 43, no. 2, pp. 685–698, 2013.

[18] J. Ziegler and W. Banzhaf, "Decreasing the number of evaluations in evolutionary algorithms by using a meta-model of the fitness function," in *European Conference on Genetic Programming*. Springer, 2003, pp. 264–275.

[19] Y. Jin and B. Sendhoff, "Fitness approximation in evolutionary computation-a survey," in *Proceedings of the 4th Annual Conference on Genetic and Evolutionary Computation*. Morgan Kaufmann Publishers Inc., 2002, pp. 1105–1112.

[20] T.P. Runarsson, "Ordinal regression in evolutionary computation," in *Parallel Problem Solving from Nature-PPSN IX*. Springer, 2006, pp. 1048–1057.

[21] X. Lu, K. Tang, and X. Yao, "Classification-assisted differential evolution for computationally expensive problems," in *IEEE Congress on Evolutionary Computation (CEC), 2011*. IEEE, 2011, pp. 1986–1993.

[22] X.F. Lu and K. Tang, "Classification-and regression-assisted differential evolution for computationally expensive problems," *Journal of Computer Science and Technology*, vol. 27, no. 5, pp. 1024–1034, 2012.

[23] X. Lu, K. Tang, B. Sendhoff, and X. Yao, "A new self-adaptation scheme for differential evolution," *Neurocomputing*, vol. 146, pp. 2–16, 2014.

[24] S.S. Garud, I.A. Karimi, and M. Kraft, "Design of computer experiments: A review," *Computers & Chemical Engineering*, vol. 106, pp. 71–95, 2017.

[25] F.A. Viana, "A tutorial on latin hypercube design of experiments," *Quality and Reliability Engineering International*, vol. 32, no. 5, pp. 1975–1985, 2016.

[26] K. Crombecq, E. Laermans, and T. Dhaene, "Efficient space-filling and non-collapsing sequential design strategies for simulation-based modeling," *European Journal of Operational Research*, vol. 214, no. 3, pp. 683–696, 2011.

[27] H. Liu, Y.S. Ong, and J. Cai, "A survey of adaptive sampling for global metamodeling in support of simulation-based complex engineering design," *Structural and Multidisciplinary Optimization*, pp. 1–24, 2017.

[28] L. Gräning, Y. Jin, and B. Sendhoff, "Efficient evolutionary optimization using individual-based evolution control and neural networks: A comparative study." in *European Symposium on Artificial Neural Networks*, 2005, pp. 273–278.

[29] C. Sun, Y. Jin, J. Zeng, and Y. Yu, "A two-layer surrogate-assisted particle swarm optimization algorithm," *Soft Computing*, vol. 19, no. 6, pp. 1461–1475, 2015.

[30] H.S. Kim and S.B. Cho, "An efficient genetic algorithm with less fitness evaluation by clustering," in *IEEE Congress on Evolutionary Computation (CEC), 2001*, vol. 2. IEEE, 2001, pp. 887–894.

[31] Y. Jin and B. Sendhoff, "Reducing fitness evaluations using clustering techniques and neural network ensembles," in *Genetic and Evolutionary Computation Conference*. Springer, 2004, pp. 688–699.

[32] D.R. Jones, "A taxonomy of global optimization methods based on response surfaces," *Journal of Global Optimization*, vol. 21, no. 4, pp. 345–383, 2001.

[33] A.I. Forrester and A.J. Keane, "Recent advances in surrogate-based optimization," *Progress in Aerospace Sciences*, vol. 45, no. 1-3, pp. 50–79, 2009.

[34] M. Sasena, P. Papalambros, and P. Goovaerts, "Metamodeling sampling criteria in a global optimization framework," in *8th Symposium on Multidisciplinary Analysis and Optimization*, 2000, p. 4921.

[35] D.D. Cox and S. John, "Sdo: A statistical method for global optimization," in *in Multidisciplinary Design Optimization: State-of-the-Art*, 1997, pp. 315–329.

[36] H.J. Kushner, "A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise," *Journal of Basic Engineering*, vol. 86, no. 1, pp. 97–106, 1964.

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

[38] H. Wang, Y. Jin, and J. Doherty, "Committee-based active learning for surrogate-assisted particle swarm optimization of expensive problems," *IEEE Transactions on Cybernetics*, vol. 47, no. 9, pp. 2664–2677, 2017.

[39] H. Tong, C. Huang, J. Liu, and X. Yao, "Voronoi-based efficient surrogate-assisted evolutionary algorithm for very expensive problems," in *IEEE Congress on Evolutionary Computation, CEC 2019, Wellington, New Zealand, June 10-13, 2019*. IEEE, 2019, pp. 1996–2003.

[40] R. Cheng, C. He, Y. Jin, and X. Yao, "Model-based evolutionary algorithms: a short survey," *Complex & Intelligent Systems*, vol. 4, no. 4, pp. 283–292, 2018.

[41] Y. Jin, H. Wang, T. Chugh, D. Guo, and K. Miettinen, "Data-driven evolutionary optimization: an overview and case studies," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 3, pp. 442–458, 2018.

[42] H.S. Bernardino, H.J. Barbosa, and L.G. Fonseca, "Surrogate-assisted clonal selection algorithms for expensive optimization problems," *Evolutionary Intelligence*, vol. 4, no. 2, pp. 81–97, 2011.

[43] X. Llorà, K. Sastry, D.E. Goldberg, A. Gupta, and L. Lakshmi, "Combating user fatigue in igas: partial ordering, support vector machines, and synthetic fitness," in *Genetic and Evolutionary Computation Conference, GECCO 2005, Proceedings, Washington DC, USA, June 25-29, 2005*, 2005, pp. 1363–1370.

[44] A. Ratle, "Kriging as a surrogate fitness landscape in evolutionary optimization," *Artificial Intelligence for Engineering Design and Manufacturing*, vol. 15, no. 1, pp. 37–49, 2001.

[45] R.G. Regis, "Evolutionary programming for high-dimensional constrained expensive black-box optimization using radial basis functions," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 3, pp. 326–347, 2014.

[46] T. Østergård, R.L. Jensen, and S.E. Maagaard, "A comparison of six metamodeling techniques applied to building performance simulations," *Applied Energy*, vol. 211, pp. 89–103, 2018.

[47] R. Jin, W. Chen, and T.W. Simpson, "Comparative studies of metamodelling techniques under multiple modelling criteria," *Structural and Multidisciplinary Optimization*, vol. 23, no. 1, pp. 1–13, 2001.

[48] M.A. Nik, K. Fayazbakhsh, D. Pasini, and L. Lessard, "A comparative study of metamodeling methods for the design optimization of variable stiffness composites," *Composite Structures*, vol. 107, pp. 494–501, 2014.

[49] A. Díaz-Manríquez, G. Toscano, and C.A.C. Coello, "Comparison of metamodeling techniques in evolutionary algorithms," *Soft Computing*, vol. 21, no. 19, pp. 5647–5663, 2017.

[50] L.J. Herrera, H. Pomares, I. Rojas, A. Guillén, G. Rubio, and J. Urquiza, "Global and local modelling in rbf networks," *Neurocomputing*, vol. 74, no. 16, pp. 2594–2602, 2011.

[51] L. Fonseca, H. Barbosa, and A. Lemonge, "On similarity-based surrogate models for expensive single-and multi-objective evolutionary optimization," in *Computational Intelligence in Expensive Optimization Problems*. Springer, 2010, pp. 219–248.

[52] A. Hameed, A. Kononova, and D. Corne, "Engineering fitness inheritance and co-operative evolution into state-of-the-art optimizers," in *IEEE Symposium Series on Computational Intelligence, 2015*. IEEE, 2015, pp. 1695–1702.

[53] J.H. Chen, D.E. Goldberg, S.Y. Ho, and K. Sastry, "Fitness inheritance in multi-objective optimization," in *Proceedings of the 4th Annual Conference on Genetic and Evolutionary Computation*. Morgan Kaufmann Publishers Inc., 2002, pp. 319–326.

[54] R.E. Smith, B.A. Dike, and S.A. Stegmann, "Fitness inheritance in genetic algorithms," in *Proceedings of the 1995 ACM Symposium on Applied Computing*, ser. SAC '95, no. 6. ACM, 1995, pp. 345–350.

[55] C. Sun, J. Zeng, J. Pan, S. Xue, and Y. Jin, "A new fitness estimation strategy for particle swarm optimization," *Information Sciences*, vol. 221, pp. 355–370, 2013.

[56] T. Hildebrandt and J. Branke, "On using surrogates with genetic programming," *Evolutionary Computation*, vol. 23, no. 3, pp. 343–367, 2015.

[57] N. Hansen, "The cma evolution strategy: a comparing review," in *Towards a new evolutionary computation*. Springer, 2006, pp. 75–102.

[58] C. Li, S. Yang, and T.T. Nguyen, "A self-learning particle swarm optimizer for global optimization problems," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 42, no. 3, pp. 627–646, 2012.

[59] T.P. Runarsson, "Approximate evolution strategy using stochastic ranking," in *Evolutionary Computation, 2006. CEC 2006. IEEE Congress on*. IEEE, 2006, pp. 745–752.

[60] C. Audet, M. Kokkolaras, S. Le Digabel, and B. Talgorn, "Order-based error for managing ensembles of surrogates in mesh adaptive direct search," *Journal of Global Optimization*, vol. 70, no. 3, pp. 645–675, 2018.

[61] W. Gong, A. Zhou, and Z. Cai, "A multioperator search strategy based on cheap surrogate models for evolutionary optimization," *IEEE Transactions Evolutionary Computation*, vol. 19, no. 5, pp. 746–758, 2015.

[62] C. He, C. Wang, Y.X. Zhong, and R.F. Li, "A survey on learning to rank," in *International Conference on Machine Learning and Cybernetics*, vol. 3. Ieee, 2008, pp. 1734–1739.

[63] I. Loshchilov, M. Schoenauer, and M. Sebag, "Self-adaptive surrogate-assisted covariance matrix adaptation evolution strategy," in *Proceedings of the 14th Annual Conference on Genetic and Evolutionary Computation*, no. 8. ACM, 2012, pp. 321–328.

[64] C. Huang, B. Radi, A.E. Hami, and H. Bai, "Cma evolution strategy assisted by kriging model and approximate ranking," *Applied Intelligence*, vol. 48, no. 11, pp. 4288–4304, 2018.

[65] M. Lapin, M. Hein, and B. Schiele, "Analysis and optimization of loss functions for multiclass, top-k, and multilabel classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[66] J. Zhang, A. Zhou, K. Tang, and G. Zhang, "Preselection via classification: A case study on evolutionary multiobjective optimization," *Information Sciences*, vol. 465, pp. 388–403, 2018.

[67] Z.H. Zhou and X.Y. Liu, "Training cost-sensitive neural networks with methods addressing the class imbalance problem," *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 1, pp. 63–77, 2006.

[68] Q. Chen, B. Liu, Q. Zhang, J. Liang, P. Suganthan, and B. Qu, "Problem definitions and evaluation criteria for cec 2015 special session on bound constrained single-objective computationally expensive numerical optimization," *Technical Report, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou, China and Technical Report, Nanyang Technological University*, 2014.

[69] F. Peng, K. Tang, G. Chen, and X. Yao, "Population-based algorithm portfolios for numerical optimization," *IEEE Transactions on evolutionary computation*, vol. 14, no. 5, pp. 782–800, 2010.