An Adaptive Model Selection Strategy for Surrogate-Assisted Particle Swarm Optimization Algorithm

Haibo Yu
School of Mechanical Engineering
Taiyuan University of Science and Technology
Taiyuan, 030024 China
tyustyuhaibo@126.com

Chaoli Sun
Department of Computer Science and Technology
Taiyuan University of Science and Technology
Taiyuan, 030024 China
chaoli.sun.cn@gmail.com

Yin Tan
Division of Industrial and System Engineering
Taiyuan University of Science and Technology
Taiyuan, 030024 China
tanying1965@gmail.com

Jianchao Zeng
School of Computer Science and Control Engineering
North University of China
Taiyuan, 030051 China
zengjianchao@263.net

Yaochu Jin
[1] Department of Computer Science
University of Surrey
Guildford, GU2 7XH, UK
[2] Department of Computer Science and Technology
Taiyuan University of Science and Technology
Taiyuan, 030024 China
yaochu.jin@surrey.ac.uk

***Abstract*—Computationally expensive problems pose a serious challenge to the successful application of evolutionary algorithms to complex engineering optimization. To address this challenge, surrogate models, also known as metamodels, are commonly used in lieu of the expensive fitness function for the computational cost of optimization. However, it is nontrivial to choose an appropriate metamodel for properly replacing the expensive fitness function. In this paper, an adaptive model selection strategy based on fitness landscape analysis is proposed for a surrogate-assisted particle swarm optimization algorithm. The structure of the sampling space is learned, based on which the more suited surrogate model, either a polynomial regression model or a radial basis function network, will be chosen to estimate the fitness value. Simulation results on seven widely used benchmark functions demonstrate the efficacy of the proposed algorithm.**

Keywords—Surrogate models; Particle swarm optimization; Fitness landscape analysis; Fitness distance correlation; Model selection

I. INTRODUCTION

Evolutionary algorithms are a class of powerful global optimizers. They operate with a population and require the computation of the fitness function for each individual in the population at every generation. Compared with conventional optimization algorithms, evolutionary algorithms are more efficient for problems which are discontinuous, non-differential, multi-modal, noisy and not well-defined. Due to these merits, evolutionary algorithms have attracted increasing interest from industry. In many practical applications, however, like streamline optimization of vehicles and aerodynamic analysis of aircraft wings, the computation time for each function evaluation may take a few hours or even days. As a result, application of evolutionary algorithms to such problems will be prohibitive due to the large number of expensive fitness evaluations. Therefore, methods to reduce the number of expensive fitness evaluation are sought to widen the application range of evolutionary algorithms. In the past decades, surrogate models, also known as metamodels, were proposed as a computationally efficient means to replace the expensive function to save the total optimization time [1].

The most prominent and commonly used surrogates in engineering design optimization include polynomial regression (PR) [2, 3], Kriging or Gaussian process (KG) [4-8], artificial neural networks (ANN) [9], radial basis function networks [10, 11], radial basis functions (RBF) [12, 13], and support vector machines (SVM) [14, 15].

However, no single surrogate model is best suited for all types of problems and which surrogate should be used for a given problem largely depends on the nature of the fitness landscape. Comparative studies have been carried out for assessing the properties of various surrogate models. Simpson et al [16] made a comparative study on the approximation

This work was sponsored in part by National Natural Science Foundation of China (Grant Nos. 61472269, 61403272, and 61403271) and the State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University, China.

accuracy and prediction performance of the surrogate models between Kriging and PR on a real-world engineering design problem, in which similar performance for PR and Kriging was observed based on the empirical results. Giunta et al [17] discussed PR and DACE interpolating models through calculating the modeling accuracy on several test problems, and their results indicated that PR is more accurate than DACE in terms of several metrics of modeling error. Different from the above comparisons, Jin et al [18] compared four commonly used metamodeling methods based on multiple performance measures using 14 test problems with different features. Their conclusion was that the RBF performs best among the four metamodeling methods in terms of both average accuracy and robustness. In addition, the RBF performs best for small training data and discrete sample sets, as well as for test problems with different degrees of nonlinearity. It is worth noting that PR is simpler in the form and more convenient for algebraic analysis compared with other metamodeling methods. Moreover, Zhao et al [19] proposed a quantitative method to describe the characteristics of the sample data for the comparative study of the relationship between sample quality merits and performance measures of several metamodeling methods.

According to the valuable empirical results discussed in [18, 19], we can see that an appropriate metamodel may be achieved by characterizing the problem fitness landscape and the distribution of the sample data in the fitness landscape. In other words, some prior information of fitness landscape the problem can be used to guide model selection. Yang et al [20] applied a convolution smoothing mechanism to the fitness landscape analysis, and proposed a coarse to fine function smoothing method by tracking the global optimum over different smoothing levels. Pei et al [21, 22] employed discrete Fourier transform to extract important frequency information of the fitness landscape by resampling the search space, and constructed an trigonometric approximation model for fitness estimation using inverse discrete Fourier transform. However, most existing methods are either limited to simplifying the landscapes through suppressing the local optima or only suitable for some particular function optimizations. When it comes to expensive and complex optimization problems with unknown internal structures, such as expensive black-box optimization, the above methods may be inefficient or infeasible. Thus in this paper, an adaptive model selection strategy based on fitness landscape analysis is proposed, in which fitness distance correlation (FDC) is utilized to gain insight into the features of the local fitness landscape and then one of polynomial regression and radial basis function will adaptively be chosen to replace the real time-consuming fitness evaluations.

The rest of the paper is organized as follows. Section II briefly reviews existing fitness landscape analysis techniques and the canonical particle swarm optimization algorithm. Our proposed method is given in Section III. Section IV gives the numerical results and analysis. Finally, Section V summarizes the main contributions of our paper and some suggestions for future work.

II. Background

A. Fitness Landscape Analysis Methods

A large number of fitness landscape analysis techniques for quantitatively measuring the difficulty of optimization problems have been proposed in the past two decades. Popular methods include auto-correlation and correlation length [23], information and partial information content [24-26], fitness cloud [27, 28], information landscape [29, 30], and fitness distance correlation [31]. Comprehensive surveys of the fitness landscape analysis methods were presented in [32-34].

Fitness distance correlation (FDC) was first proposed by Jones et al [31] as a measure of problem difficulty to study the performance of genetic algorithms. FDC coefficient determines the degree of correlation between the distances of samples and their corresponding fitness values to the global optimum. It was often used to reflect the global feature of fitness landscape. FDC metric has received wide attention since it was put forward. Tomassini et al [35] discussed the performance of genetic programing through quantifying the problem hardness with FDC. Interesting results were obtained by exploiting FDC analysis in set covering problems [36]. In [37], effectiveness of the evolutionary meta-search was investigated assisted with FDC analysis and FDC scatter plots. Müller et al [38] transferred the concept of FDC analysis from theoretical biology and discrete combinatorial optimization to continuous optimization, and classified the CEC 2005 benchmark functions based on their FDC coefficients, the robust and accuracy of the landscape characterization were also tested.

Formally, given a set $\boldsymbol{F} = \{f_1, f_2, ..., f_n\}$ of n individual fitness values and a corresponding set $\boldsymbol{D} = \{d_1, d_2, ..., d_n\}$ of the n distances to the nearest global optimum, FDC is defined as:

$$FDC = \frac{\frac{1}{n}\sum_{i=1}^{n}(f_i - \bar{f})(d_i - \bar{d})}{\sigma(\boldsymbol{F})\sigma(\boldsymbol{D})}, \quad (1)$$

where $\bar{f}$, $\bar{d}$, $\sigma(\boldsymbol{F})$ and $\sigma(\boldsymbol{D})$ are the means and standard deviations of $\boldsymbol{F}$ and $\boldsymbol{D}$. Obviously, $FDC \in [-1,1]$. Without loss of generality, for minimization problems, $FDC > 0$ indicates that the fitness value will decrease when the distance to the global optimum decreases, meaning a positive correlation, while $FDC < 0$ shows that the fitness value will increase when the distance to the global optimum decreases, representing a negative correlation. Similar to the correlation coefficient in statistics, FDC gives a description of a linear relationship between variables of distance and fitness value. Global characterization of the fitness function can then be correspondingly reflected by FDC analysis. However, the implementation of original FDC needs to know the global optimum, which is obviously unknown in real-world problems. Therefore, the best sample with best fitness value in the current sample set is often used as a substitute for the global optimum for FDC analysis. Malan [39] introduced a modified version of FDC and denoted FDC_s :

$$FDC_s = \frac{\sum_{i=1}^{n}(f_i - \bar{f})(d_i^* - \bar{d}^*)}{\sqrt{\sum_{i=1}^{n}(f_i - \bar{f})^2}\sqrt{\sum_{i=1}^{n}(d_i^* - \bar{d}^*)^2}} \qquad (2)$$

where $d_i^* \in \boldsymbol{D}^* = \{d_1^*, d_2^*, ..., d_n^*\}$, is the distance between the i^{th} individual and the best position in the current sample set, $\bar{d}^*$ is the mean of $\boldsymbol{D}^*$.

As discussed above, FDC coefficient gives useful information on the global structure of fitness landscape. In addition, the FDC value shows the complexity of the problem landscape and linearity or ruggedness.

B. Particle Swarm Optimization

The canonical particle swarm optimization (PSO) is a class of heuristic swarm intelligence algorithms. The initial idea of PSO originated from the studies on the foraging behavior of birds and fish in nature. Kennedy and Eberhart [40] developed the PSO by modifying the simulation model of birds flocking to make the particles fly towards the best solution in the solution space [41]. Because of the simplicity in its principle, easiness to implement, and no requirement on gradient information, PSO has been widely used in engineering design and computational science [42-44].

In the canonical PSO, an initial population is randomly generated, and each member in the population is assigned with a certain speed. In each generation, each particle updates its position and velocity by taking into account the best position it has found so far and the best position of all particles in the population up to current generation. The equations for updating the velocity and position of each particle are given as follows:

$$\begin{aligned} \boldsymbol{v}_i^{(t+1)} &= \omega \boldsymbol{v}_i^{(t)} + c_1 r_1 (\boldsymbol{pbest}_i^{(t)} - \boldsymbol{x}_i^{(t)}) + c_2 r_2 (\boldsymbol{gbest}^{(t)} - \boldsymbol{x}_i^{(t)}) \\ \boldsymbol{x}_i^{(t+1)} &= \boldsymbol{x}_i^{(t)} + \boldsymbol{v}_i^{(t+1)} \end{aligned} \qquad (3)$$

where $\boldsymbol{x}_i^{(t)}$, $\boldsymbol{v}_i^{(t)}$, $\boldsymbol{x}_i^{(t+1)}$ and $\boldsymbol{v}_i^{(t+1)}$ are the position and velocity of i^{th} particle at time t and $t+1$, ω is the inertia weight, c_1 and c_2 are acceleration coefficients, r_1 and r_2 are random numbers uniformly distributed in the interval $[0,1]$, $\boldsymbol{pbest}_i^{(t)}$ is i^{th} particle's best position found up to time t, and $\boldsymbol{gbest}^{(t)}$ is the best position found by the swarm so far.

III. Model Selection Strategy Based on Fitness Landscape Analysis

Whether a surrogate model is able to enhance the search efficiency of an optimization algorithm heavily depends on whether the surrogate is able to properly describe the local fitness landscape the population is currently searching. Therefore, model selection strategy is very important in surrogate-assisted evolutionary algorithms. Techniques like split sample, cross validation and bootstrapping are commonly used for the model selection [3]. Metamodel which yields the minimum error is often taken as the optimal one in surrogate-based evolutionary algorithm [45-47]. As the purpose of evolutionary search is to find improved solutions, Le et al [48] suggested to use an evolvability measurement to assess the evolvability of each surrogate to select the optimal one under the assumption that the surrogate utilized are precise enough to fit the problem landscape. Different to the method proposed in [48], in this paper, we propose to use fitness distance correlation to select the surrogate model which is able to best capture the real fitness function.

A. Relationship Between Landscape Characters and Metamodels

The problem structure is not only one of the important factors that influence the efficiency of evolutionary algorithms, but also an important criterion to choose a surrogate model for the objective function. Although they have been widely applied as quantitative measures related to the features of the fitness landscape, to the best of our knowledge, fitness landscape analysis methods have not yet been utilized in the metamodel selection strategies. In this paper, we present preliminary work that explores the fitness landscape analysis metric to measure some important features of problem, such as neutrality, linearity and ruggedness, in order to assist the selection of the surrogate in solving computationally expensive problems.

Generally, the inherent structure of metamodel can indirectly reflect its approximation ability, such as the combinations of the basis functions. Taking the PR as an example, different orders of PR have different landscape characterizations. For single peak problems, effective approximation using zero and first order PR are hard to achieve, while high approximation accuracy can be obtained by second order PR. Equally, model structural analysis is also needed for other surrogate models [49-51]. Furthermore, uncertainty introduced by different metamodels should also be considered when the problem landscape is approximated [52, 53]. In other words, the purpose for conducting fitness landscape analysis is to reduce the impact of uncertainty to make a better choice from the candidate metamodels. Therefore, if we can successfully estimate features of the fitness landscape, we can develop an effective model selection strategy.

Moreover, we can use fitness landscape analysis methods to acquire information about the degree of nonlinearity of the problem landscape for selecting a surrogate model. As discussed in [18], PR is the best candidate for problems with low-order nonlinearity, while RBF is believed to perform best for problems with high-order nonlinearity. Consider the FDC metric described in Section II, for minimization problems, the degree of nonlinearity can be obtained through FDC coefficient. Low-order nonlinearity of fitness landscape will be shown if FDC coefficient is close to the value of 1 or -1, which indicates that the problem landscape is smooth, while high-order nonlinearity can be found if the FDC coefficient is close to zero, which implies that the problem is very rugged. Therefore, based on the FDC coefficient, we propose to adaptively make selection between PR and RBF for efficiently assisting the evolutionary algorithms. Similar to the discussions on the approximation quality assessment of PR and RBF in [54], we firstly compare three mathematical functions with different fitness landscape characters (as shown in Table I) to study the performance of PR and RBF, and the results are shown in Figs. 1-3.

From Figs.1-3, we can conclude that RBF performs better than PR in multi-modal landscapes with high-order nonlinearity by capturing more local information, whereas PR and RBF have similar accuracy in the unimodal landscape with low-order nonlinearity.

TABLE I. TEST FUNCTIONS

Test Functions	Feature	Nonlinearity
$f_1(x_1,x_2)=x_1^2+x_2^2$, $[-10,10]$	Unimodal	Low-order
$f_2(x_1,x_2)=\sin(x_1)+\sin(x_2)$, $[0,2\pi]$	Multimodal	High-order
$f_3(x_1,x_2)=\sin\sqrt{x_1^2+x_2^2}\,/\sqrt{x_1^2+x_2^2}$, $[-3\pi,3\pi]$	Multimodal	High-order

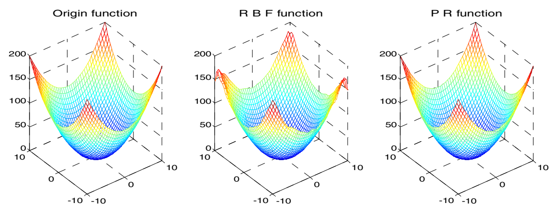


Fig. 1. Metamodel surface plots of f_1 .

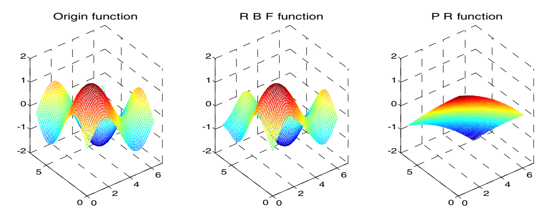


Fig. 2. Metamodel surface plots of f_2 .

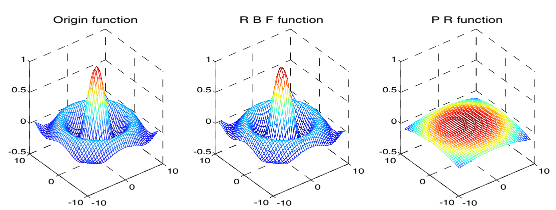


Fig. 3. Metamodel surface plots of f_3 .

B. *Some Practical Issues*

As we use FDC for the fitness landscape analysis, some practical issues regarding the effectiveness of FDC are considered. Among others, sample uniformity, sample size and distance metrics are fundamental to determine the accuracy of FDC metric. Generally, if sufficient sample points are uniformly distributed across the problem landscape, accurate FDC metric can be obtained. However, since the individuals will gradually aggregate in the optimal region in the process of evolutionary search, sample points collected from different regions of the fitness landscape are of different characteristics, resulting in different accuracy of FDC coefficients. Moreover, there exist numerous distance metrics, such as the Hamming distance and Euclidean distance. In discrete optimization, the Hamming distance is often used for FDC, whereas Euclidean distance is used in the continuous case. In this paper, since we focus on continuous optimization, Euclidean distance is thus chosen for the calculation of FDC. Nevertheless, it is worth noting that the FDC of individual's neighborhood landscape is only estimated by a small number of neighborhood samples. We aim to provide a preliminary exploitation of the relationship between features of fitness landscape and best suited surrogate.

C. *The Proposed Algorithm*

Our motivation comes from two aspects. First, based on the fitness landscape analysis, we determine whether the fitness of an individual is to be evaluated using the real fitness function or estimated using a surrogate. Second, a surrogate will be automatically chosen to be utilized for fitness approximation based on the characters of its neighborhood landscape, such as neutrality, linearity, and ruggedness.

The main steps of the proposed method are given below:

Step 1: Randomly initialize the positions and velocities of particles, $\boldsymbol{x}_i^{(0)}$ and $\boldsymbol{v}_i^{(0)}$, $i=1,2,...,N$, and evaluate the fitness value of each particle $f(\boldsymbol{x}_i^{(0)})$.

Step 2: Build the historical database using all the particles produced in the first j_0 generations whose fitness are evaluated using the real objective function.

Step 3: Update the velocity and position using (3) for each particle.

Step 4: For each particle $\boldsymbol{x}_i^{(j_0+1)}$, $i=1,2,...,N$, find its neighbors in the database $(\boldsymbol{n}_{ij}^{(j_0+1)}, f(\boldsymbol{n}_{ij}^{(j_0+1)}))$, $j=1,2,...,n$, where n is the number of samples in its neighborhood. If the maximum distance between particle $\boldsymbol{x}_i^{(j_0+1)}$ and its neighbors is less than a predefined threshold, then the fitness of the nearest sample is assigned to $\boldsymbol{x}_i^{(j_0+1)}$, go to Step 8; otherwise go to Step 5.

Step 5: If the number of samples in the neighborhood $n<(D+1)(D+2)/2$, then evaluate the fitness using the real fitness function; otherwise, choose $(D+1)(D+2)/2$ nearest samples in the database to $\boldsymbol{x}_i^{(j_0+1)}$.

Step 6: Calculate the maximum difference of fitness values in neighbors: $Fd=\max\left\{f_{norm}(\boldsymbol{n}_{ij}^{(j_0+1)})\right\}_1^n-\min\left\{f_{norm}(\boldsymbol{n}_{ij}^{(j_0+1)})\right\}_1^n$, where $f_{norm}(\boldsymbol{n}_{ij}^{(j_0+1)})$ is the normalized fitness value of j^{th} neighborhood sample $\boldsymbol{n}_{ij}^{(j_0+1)}$. If $Fd\le eps$, estimate the fitness value of particle $\boldsymbol{x}_i^{(j_0+1)}$ by averaging the fitness values of all neighborhood samples, go to Step 8; otherwise, go to Step 7.

Step 7: Estimate the FDC coefficient of neighborhood landscape of particle $\boldsymbol{x}_i^{(j_0+1)}$. If $|FDC|\ge\varepsilon_{FDC}$, select PR as the metamodel for estimating the fitness of particle

$\boldsymbol{x}_i^{(j_0+1)}$, otherwise construct an RBF and approximate the fitness of particle $\boldsymbol{x}_i^{(j_0+1)}$.

Step 8: Update the best position $\boldsymbol{pbest}_i^{(j_0+1)}$ for each particle and overall best position $\boldsymbol{gbest}^{(j_0+1)}$ for the population. If the improved position is estimated by metamodel, then re-evaluate it using the real fitness function, and archive this position and its fitness value into the database.

Step 9: Terminate if the stopping criterion is satisfied, otherwise, go to step 3.

As we can see that the algorithm begins with a randomly initialized population. All positions and corresponding fitness at the first j_0 generations are saved in the database. Because of the random nature of the PSO search, the positions of the particles may overlap in the evolutionary search. In our method, only non-duplicated particles will be archived in the initialization stage of database. It is worth noting that the nearest samples in the neighborhood are utilized for fitness landscape analysis. In this paper, a hypersphere is used as the neighborhood of particle $\boldsymbol{x}_i^{(j_0+1)}$, and its radius is determined by the distance between particle $\boldsymbol{x}_i^{(j_0+1)}$ and the global best position $\boldsymbol{gbest}^{(j_0)}$ in previous generation, and an upper bound is set $\left\|\boldsymbol{v}_i^{(t+1)}\right\|_2$. The purpose is to detect the particle which has insufficient near-samples for constructing a metamodel. To avoid overfitting, the maximum number of samples in the neighborhood is set to $(D+1)(D+2)/2$, which is the least number required for constructing the PR. On the other hand, a lower bound of neighborhood is set by a predefined threshold. Moreover, given a group of neighborhood samples, their fitness values are first normalized using the following equation:

$$f_{norm}(\boldsymbol{n}_{ij}) \triangleq f(\boldsymbol{n}_{ij}) / \sum_1^n f(\boldsymbol{n}_{ij}) \qquad (4)$$

where $f_{norm}(\boldsymbol{n}_{ij})$ is the normalized fitness value of j^{th} neighborhood sample $\boldsymbol{n}_{ij}$. The purpose of step 6 is to recognize a near-neutral area by determining the degree of ruggedness of neighborhood fitness landscape. Additionally, according to the FDC coefficients, the features of neighborhood fitness landscapes can be divided into two categories when it is close to 1 or -1, namely, uphill or downhill section and single peak or basin section, as shown in Fig. 4, where the solid and dashed lines denote the original one dimensional function and its neighborhood fitness landscape, respectively, assuming only five sample points in each neighborhood. We can find that the neighbor fitness landscapes in these two categories both have low-order nonlinearity, and then we use the absolute value of FDC coefficient as the model selection criterion in step 7. Note that at the late stage of evolutionary search, when a large number of particles locate in the optimal region (indicating a smooth basin), the FDC coefficients of sample sets in this region may be close to 1.

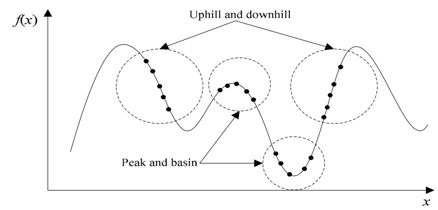


Fig. 4. Explanation to the absolute value of FDC.

IV. Empirical Study

In the empirical study, seven commonly used 10-dimensional benchmark functions (refer to Table II in the Appendix) are chosen to evaluate the performance of the proposed algorithm, a surrogate-based particle swarm optimization with fitness distance correlation (SPSOFDC). Three algorithms are selected for a comparative study, namely, a standard PSO (SPSO) as described in Section II, a PR-assisted PSO (SPSOPR), and a RBF-assisted PSO (SPSORBF). For all the benchmark problems, the parameters of the above four algorithms are set up as follows. The population size is set to 50, which can guarantee sufficient samples for FDC analysis. The maximum number of iteration is set to 500, and the inertial weighting factor ω is decreased linearly from 0.9 to 0.4. The cognition and social coefficients are set to $c_1 = c_2 = 2.05$. The maximum velocity is set to $v_{\max} = x_{\max}$, where $x_{\max}$ is the upper bound of the benchmark function. Set $j_0 = 10$ for the database building phase which includes 500 exact evaluations. The computational budget is 6000 exact fitness evaluations and the accuracy is $eps = 10^{-8}$. The threshold for determining the lower bound of neighborhood is 10^{-7}. In addition, 20 independent runs are conducted for statistical analysis. To study the effect of the threshold parameter ε_{FDC} on the performance of SPSOFDC, we set $\varepsilon_{FDC} = 0.75$ as a reference value. We also run experiments using $\varepsilon_{FDC} = 0.65$ and 0.85 for fair comparisons. The convergence profiles of the four algorithms on the benchmark functions are shown in Fig. 5.

Obviously, the performance of the standard PSO is the worst when compared with alternative methods. In terms of F1, F2, F4, and F7, one can observe from the results shown in Fig. 5 that the performance of SPSOFDC is comparable to SPSOPR or SPSORBF by increasing or decreasing the threshold parameter. For functions F3 and F4, the convergence performances of SPSOFDC become worse when $\varepsilon_{FDC} = 0.65$ or $\varepsilon_{FDC} = 0.85$. However, we can find that the property of SPSOFDC is improved before 4500 exact evaluations when setting $\varepsilon_{FDC} = 0.65$. In contrast, improvement is achieved when adjusting the threshold parameter in SPSOFDC for F1, F5, F6 and F7. Moreover, SPSOFDC is better than SPSORBF when we set 0.65 and 0.85 for F7 and F6, respectively. The results in Fig. 5 indicate that the threshold parameter ε_{FDC} directly determine the convergence property of SPSOFDC. However, the results also indicate that, for different problems, the selection of an optimal threshold parameter is nontrivial and is problem dependent. The statistical results after 6000 exact evaluations are summarized in Fig. 6 and the frequencies of surrogate usage in SPSOFDC search with different ε_{FDC} are also shown in Fig. 7. The results are consistent with the convergence profiles in Fig. 6. For functions F1, F2, and F7, there exists a high frequency of usage for PR in SPSOFDC. On the contrary, a high frequency of usage for RBF is exhibited in SPSOFDC on functions F3, F4, F5 and F6. Note that the number of calls for PR is extremely larger than RBF for F1 and F2, which are multimodal functions with a global single funnel topology in low dimensions, while it is nearly zero compared to RBF for functions F3 and F4. Overall, the numerical results confirm the efficiency of the proposed algorithm SPSOFDC on optimizing the selected benchmark problems.

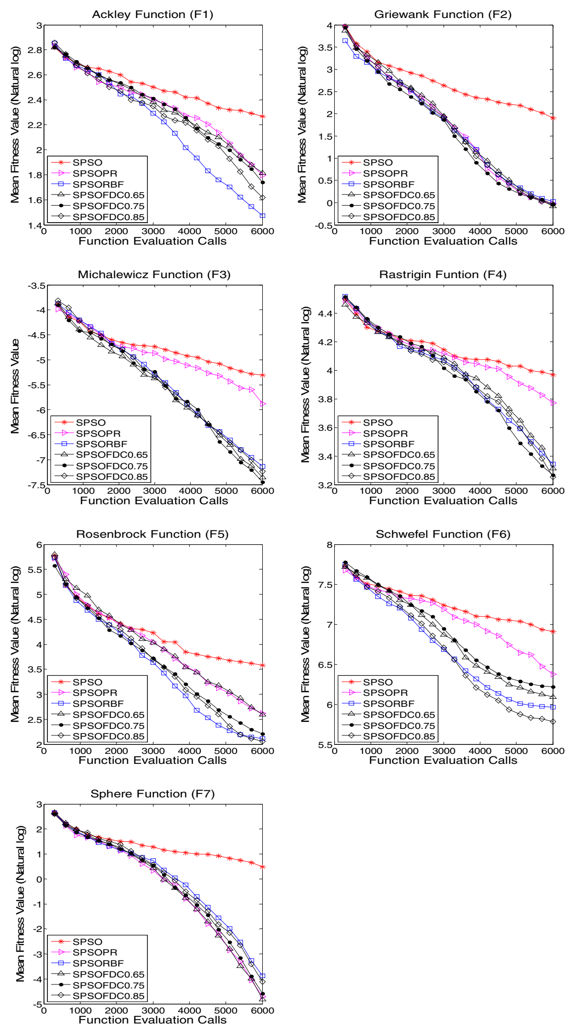
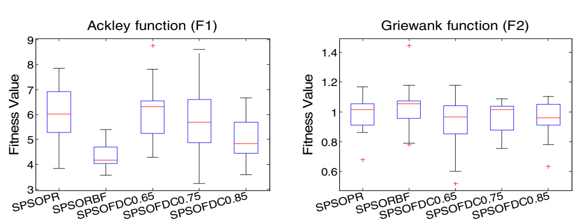


Fig. 5. Performances of SPSOPR, SPSORBF, and SPSOFDC with different ε_{FDC} on the benchmark functions.



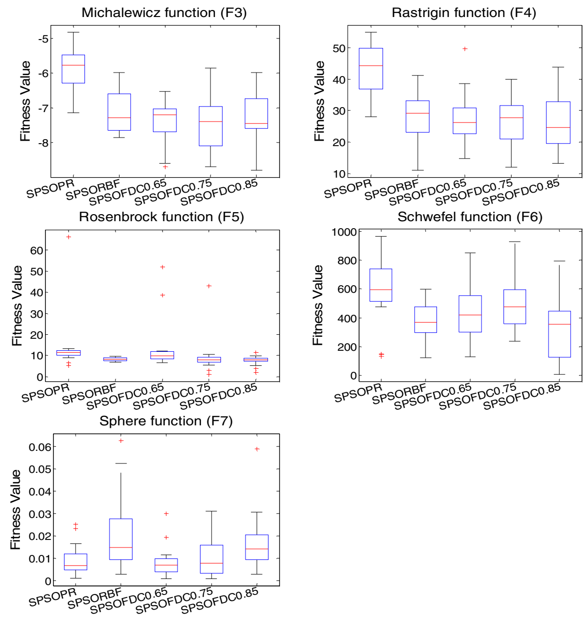


Fig. 6. Boxplots of the statistic results of SPSOPR, SPSORBF, and SPSOFDC with three different ε_{FDC} on the benchmark functions after 6000 exact evaluations.

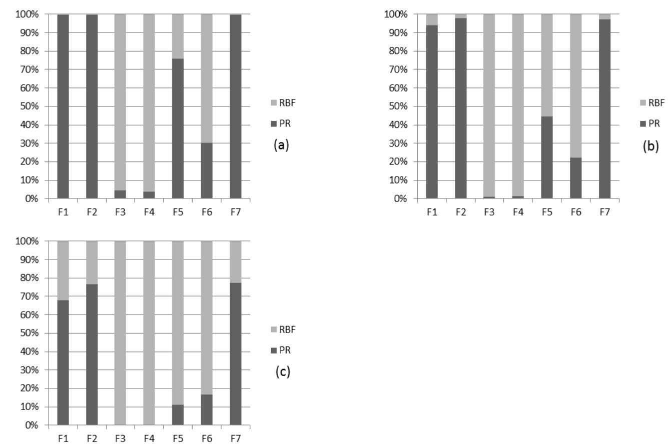


Fig. 7. Surrogate frequencies of usage in SPSOFDC search with different ε_{FDC} , (a) for ε_{FDC}=0.65 , (b) for ε_{FDC}=0.75 , and (c) for ε_{FDC}=0.85 .

V. Summary and Conclusions

In this paper, we present a preliminary study on using fitness landscape analysis for selecting surrogates in surrogate-assisted evolutionary algorithms. The relationship between candidate metamodels and features of local problem landscape for each individual in the current population is characterized using a fitness distance correlation metric. The effectiveness of the proposed strategy is validated on seven benchmark functions by incorporating it into a surrogate-assisted particle swarm optimization algorithm. Comparative convergence results on seven benchmark problems indicated that the proposed SPSOFDC method is effective. Consistently satisfactory performance of the proposed algorithm has been achieved on all benchmark functions studied in this work.

Last but not least, the performance of our method can be further improved by adjusting the threshold parameter ε_{FDC} in the future. In addition, more surrogate models, such as Kriging and support vector machine, will be considered to be used as a candidate surrogate model so as to improve the quality of the fitness approximation.

VI. APPENDIX

In this section, we introduce the general forms of two surrogate models used in our method, namely, polynomial regression and radial basis function. Given a group of samples $\boldsymbol{X} = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, ..., \mathbf{x}^{(n)}\}$ and their fitness $\boldsymbol{f} = \{f_1, f_2, ..., f_n\}$, the two surrogate models are constructed as follows.

A. Polynomial Regression

Generally, the second-order polynomial regression is often used and can be expressed as follows:

$$\tilde{f}(\mathbf{x}) = \beta_0 + \sum_{i=1}^{N} \beta_i x_i + \sum_{i=1}^{N} \sum_{j \le i}^{N} \beta_{ij} x_i x_j \tag{5}$$

where β_0, β_i and β_{ij} are regression coefficients estimated by least squares, N is the dimension of problem, and x_i is the i^{th} design variable.

B. Radial Basis Function

The radial basis function used in this paper is of an interpolation form:

$$\tilde{f}(\mathbf{x}) = \sum_{i=1}^{n} \lambda_i \varphi\left(\left\|\mathbf{x} - \mathbf{x}^{(i)}\right\|\right) + p(\mathbf{x}), \mathbf{x} \in R^d \tag{6}$$

where $\varphi(\bullet)$ is the kernel function, $\|\bullet\|$ is the Euclidean norm, λ_i, $i = 1, 2, ..., n$, is combination coefficient of each kernel function, n is the number of interpolation points $\{\mathbf{x}^{(i)}\}_{i=1}^{n}$, $p(\mathbf{x})$ is a linear polynomial in d variables. In this paper, Gaussian kernel function was used, which is defined as $\varphi(x) = \exp(-x / \beta)$.

C. Benchmark Functions

The benchmark functions used for algorithm performance test are show in Table II.

TABLE II. BENCHMARK TEST FUNCTIONS

F1	**Ackley**	$F(\mathbf{x}) = 20 + e - 20\exp\left(-0.2\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2}\right) - \exp\left(\frac{1}{d}\sum_{i=1}^{d}\cos(2\pi x_i)\right)$, $-32.768 \le x_i \le 32.768$, $i = 1, 2, ..., d$.
F2	**Griewank**	$F(\mathbf{x}) = \sum_{i=1}^{d} \frac{x_i^2}{4000} - \prod_{i=1}^{d} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$, $-600 \le x_i \le 600$, $i = 1, 2, ..., d$.
F3	**Michalewicz**	$F(\mathbf{x}) = -\sum_{i=1}^{d} \sin(x_i) \sin^{2m}\left(\frac{i x_i^2}{\pi}\right)$, $0 \le x_i \le \pi$, $i = 1, 2, ..., d$.
F4	**Rastrigin**	$F(\mathbf{x}) = 10d + \sum_{i=1}^{d} [x_i^2 - 10\cos(2\pi x_i)]$, $-5.12 \le x_i \le 5.12$, $i = 1, 2, ..., d$.
F5	**Rosenbrock**	$F(\mathbf{x}) = \sum_{i=1}^{d-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$, $-2.048 \le x_i \le 2.048$, $i = 1, 2, ..., d$.
F6	**Schwefel**	$F(\mathbf{x}) = 418.9829d - \sum_{i=1}^{d} x_i \sin(\sqrt{\lvert x_i \rvert})$, $-500 \le x_i \le 500$, $i = 1, 2, ..., d$.
F7	**Sphere**	$F(\mathbf{x}) = \sum_{i=1}^{d} x_i^2$, $-5.12 \le x_i \le 5.12$, $i = 1, 2, ..., d$.

ACKNOWLEDGMENT

This work is supported in part by the National Natural Science Foundation of China (Grant Nos. 61472269, 61403272, and 61403271) and the State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University, China.

REFERENCES

[1] Y. Jin, "Surrogate-assisted evolutionary computation: Recent advances and future challenges," *Swarm and Evolutionary Computation,* vol. 1, pp. 61-70, 2011.

[2] F. H. Lesh, "Multi-dimensional least-squares polynomial curve fitting," *Communications of the ACM,* vol. 2, pp. 29-30, 1959.

[3] N. V. Queipo, R. T. Haftka, W. Shyy, T. Goel, R. Vaidyanathan, and P. K. Tucker, "Surrogate-based analysis and optimization," *Progress in aerospace sciences,* vol. 41, pp. 1-28, 2005.

[4] J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn, "Design and analysis of computer experiments," *Statistical science,* pp. 409-423, 1989.

[5] M. A. El-Beltagy and A. Keane, "Metalmodeling techniques for evolutionary optimization of computationally expensive problems: promises and limitations," 1999.

[6] D. Büche, N. N. Schraudolph, and P. Koumoutsakos, "Accelerating evolutionary algorithms with Gaussian process fitness function models," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on,* vol. 35, pp. 183-194, 2005.

[7] J. P. Kleijnen, "Kriging metamodeling in simulation: A review," *European Journal of Operational Research,* vol. 192, pp. 707-716, 2009.

[8] B. Liu, Q. Zhang, and G. G. Gielen, "A Gaussian process surrogate model assisted evolutionary algorithm for medium scale expensive optimization problems," *Evolutionary Computation, IEEE Transactions on,* vol. 18, pp. 180-192, 2014.

[9] Y. Jin, M. Olhofer, and B. Sendhoff, "A framework for evolutionary optimization with approximate fitness functions," *Evolutionary Computation, IEEE Transactions on,* vol. 6, pp. 481-494, 2002.

[10] Y. Jin, "A comprehensive survey of fitness approximation in evolutionary computation," *Soft Computing - A Fusion of Foundations, Methodologies and Applications,* vol. 9, pp. 3-12, 2005.

[11] C. Praveen and R. Duvigneau, "Low cost PSO using metamodels and inexact pre-evaluation: Application to aerodynamic shape design," *Computer Methods in Applied Mechanics & Engineering,* vol. 198, pp. 1087-1096, 2009.

[12] M. Powell, "Radial basis function methods for interpolation to functions of many variables," in *HERCMA*, pp. 2-24, 2001.

[13] N. Dyn, D. Levin, and S. Rippa, "Numerical procedures for surface fitting of scattered data by radial functions," *SIAM Journal on Scientific and Statistical Computing,* vol. 7, pp. 639-659, 1986.

[14] Y. Pei and H. Takagi, "Fitness landscape approximation by adaptive support vector regression with opposition-based learning," in *Systems,*

Man, and Cybernetics (SMC), 2013 IEEE International Conference on, pp. 1329-1334, 2013.

[15] J. Y. Lin, C. T. Cheng, and K. W. Chau, "Using support vector machines for long-term discharge prediction," *Hydrological Sciences Journal,* vol. 51, pp. 599-612, 2006.

[16] T. W. Simpson, J. J. Korte, T. M. Mauery, and F. Mistree, "Comparison of response surface and kriging models for multidisciplinary design optimization," 1998.

[17] A. A. Giunta, L. T. Watson, and J. Koehler, "A comparison of approximation modeling techniques: polynomial versus interpolating models," *AIAA paper,* 1998.

[18] R. Jin, W. Chen, and T. W. Simpson, "Comparative studies of metamodelling techniques under multiple modelling criteria," *Structural and Multidisciplinary Optimization,* vol. 23, pp. 1-13, 2001.

[19] D. Zhao and D. Xue, "Performance comparison of metamodeling methods from the perspective of sample quality merits," in *ASME 2009 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pp. 699-709, 2009.

[20] D. Yang and S. J. Flockton, "Evolutionary algorithms with a coarse-to-fine function smoothing," in *Evolutionary Computation, 1995., IEEE International Conference on*, pp. 657-662, 1995.

[21] Y. Pei and H. Takagi, "Fourier analysis of the fitness landscape for evolutionary search acceleration," in *Evolutionary Computation (CEC), 2012 IEEE Congress on*, pp. 1-7, 2012.

[22] Y. Pei and H. Takagi, "Comparative study on fitness landscape approximation with fourier transform," in *Genetic and Evolutionary Computing (ICGEC), 2012 Sixth International Conference on*, pp. 400-403, 2012.

[23] E. Weinberger, "Correlated and uncorrelated fitness landscapes and how to tell the difference," *Biological cybernetics,* vol. 63, pp. 325-336, 1990.

[24] M. Muñoz, M. Kirley, and S. K. Halgamuge, "Landscape characterization of numerical optimization problems using biased scattered data," in *Evolutionary Computation (CEC), 2012 IEEE Congress on*, pp. 1-8, 2012.

[25] M. Muñoz, M. Kirley, and S. K. Halgamuge, "Exploratory landscape analysis of continuous space optimization problems using information content," *Evolutionary Computation, IEEE Transactions on,* vol. 19, pp. 74-87, 2015.

[26] V. K. Vassilev, T. C. Fogarty, and J. F. Miller, "Information characteristics and the structure of landscapes," *Evolutionary computation,* vol. 8, pp. 31-60, 2000.

[27] L. Vanneschi, M. Tomassini, P. Collard, and S. Vérel, "Negative slope coefficient: A measure to characterize genetic programming fitness landscapes," in *Genetic Programming*, ed: Springer, pp. 178-189, 2006.

[28] S. Verel, P. Collard, and M. Clergue, "Where are bottlenecks in nk fitness landscapes," in *Evolutionary Computation, 2003. CEC'03. The 2003 Congress on*, pp. 273-280, 2003.

[29] B. Yossi and R. Poli, "Information landscapes and the analysis of search algorithms," in *Proceedings of the 7th annual conference on Genetic and evolutionary computation*, pp. 1287-1294, 2005.

[30] Y. Borenstein and R. Poli, "Information landscapes," in *Proceedings of the 7th annual conference on Genetic and evolutionary computation*, pp. 1515-1522, 2005.

[31] T. Jones and S. Forrest, "Fitness Distance Correlation as a Measure of Problem Difficulty for Genetic Algorithms," in *ICGA*, pp. 184-192, 1995.

[32] K. M. Malan and A. P. Engelbrecht, "A survey of techniques for characterising fitness landscapes and some possible ways forward," *Information Sciences,* vol. 241, pp. 148-163, 2013.

[33] M. A. Muñoz, Y. Sun, M. Kirley, and S. K. Halgamuge, "Algorithm selection for black-box continuous optimization problems: A survey on methods and challenges," *Information Sciences,* vol. 317, pp. 224-245, 2015.

[34] E. Pitzer and M. Affenzeller, "A comprehensive survey on fitness landscape analysis," in *Recent Advances in Intelligent Engineering Systems*, ed: Springer, pp. 161-191, 2012.

[35] M. Tomassini, L. Vanneschi, P. Collard, and M. Clergue, "A study of fitness distance correlation as a difficulty measure in genetic programming," *Evolutionary Computation,* vol. 13, pp. 213-239, 2005.

[36] M. Finker, Stützle, T. , and Lourenço, H.R., "Exploiting fitness distance correlation of set covering problems," *Lecture Notes in Computer Science,* vol. 2279, pp. 61-71, 2002.

[37] P. Merz, "Advanced fitness landscape analysis and the performance of memetic algorithms," *Evolutionary Computation,* vol. 12, pp. 303-325, 2004.

[38] C. L. Müller and I. F. Sbalzarini, "Global characterization of the CEC 2005 fitness landscapes using fitness-distance analysis," in *Applications of Evolutionary Computation*, ed: Springer, pp. 294-303, 2011.

[39] K. M. Malan, "Characterising continuous optimisation problems for particle swarm optimisation performance prediction," University of Pretoria, 2014.

[40] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *IEEE International Conference on Neural Networks, 1995. Proceedings*, vol. 4. pp. 1942-1948, 1995.

[41] D. Wang, J. Wang, H. Wang, R. Zhang, and Z. Guo, "Intelligent optimization methods," *China: Higher Education Press,* 2007.

[42] R. C. Eberhart and Y. Shi, "Particle swarm optimization: developments, applications and resources," in *evolutionary computation, 2001. Proceedings of the 2001 Congress on*, pp. 81-86, 2001.

[43] X. R. Liang, Y. K. Fan, and T. Jiang, "Application of PSO algorithm to coordinated ramp control," in *International Conference on Machine Learning and Cybernetics*, pp. 1712 – 1716, 2009.

[44] O. Jason, S. Masra, and M. S. Muhammad, "Satellite Image Segmentation with the Application of PSO Algorithm," in *Applied Mechanics and Materials*, pp. 157-163, 2016.

[45] Y. Tenne, "A computational intelligence algorithm for expensive engineering optimization problems," *Engineering Applications of Artificial Intelligence,* vol. 25, pp. 1009-1021, 2012.

[46] Y. Tenne, "An optimization algorithm employing multiple metamodels and optimizers," *International Journal of Automation and Computing,* vol. 10, pp. 227-241, 2013.

[47] Y. Tenne, "An adaptive-topology ensemble algorithm for engineering optimization problems," *Optimization and Engineering,* vol. 16, pp. 303-334, 2015.

[48] M. N. Le, Y. S. Ong, S. Menzel, Y. Jin, and B. Sendhoff, "Evolution by adapting surrogates," *Evolutionary computation,* vol. 21, pp. 313-340, 2013.

[49] M. Hüsken, Y. Jin, and B. Sendhoff, "Structure optimization of neural networks for evolutionary design optimization," *Soft Computing,* vol. 9, pp. 21-28, 2005.

[50] M. Hüsken, J. E. Gayko, and B. Sendhoff, "Optimization for problem classes-neural networks that learn to learn," in *Combinations of Evolutionary Computation and Neural Networks, 2000 IEEE Symposium on*, pp. 98-109, 2000.

[51] S. Sakata, F. Ashida, and M. Zako, "Structural optimization using Kriging approximation," *Computer Methods in Applied Mechanics and Engineering,* vol. 192, pp. 923-939, 2003.

[52] Z. Zhou, Y. S. Ong, M. H. Lim, and B. S. Lee, "Memetic algorithm using multi-surrogates for computationally expensive optimization problems," *Soft Computing,* vol. 11, pp. 957-971, 2007.

[53] Y. S. Ong, Z. Zhou, and D. Lim, "Curse and blessing of uncertainty in evolutionary algorithm using approximation," in *Evolutionary Computation, 2006. CEC 2006. IEEE Congress on*, pp. 2928-2935, 2006.

[54] Y. Tang, J. Chen, and J. Wei, "A surrogate-based particle swarm optimization algorithm for solving optimization problems with expensive black box functions," *Engineering Optimization,* vol. 45, pp. 557-576, 2013.