# A Self-adaptive Similarity-based Fitness Approximation for Evolutionary Optimization


Jie Tian
[1] Division of Industrial and System Engineering, Taiyuan University of Science and Technology, Taiyuan, 030024 China
[2] College of Information Technology, Shandong Womens University, Jinan, 250300 China
tianjie1023@outlook.com

Chaoli Sun
[1] Department of Computer Science and Technology, Taiyuan University of Science and Technology, Taiyuan, 030024 China
[2] State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University, Shenyang, 110004
chaoli.sun.cn@gmail.com

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

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, 030024 China
zengjianchao@263.net



***Abstract*—Evolutionary algorithms used to solve complex optimization problems usually need to perform a large number of fitness function evaluations, which often requires huge computational overhead. This paper proposes a self-adaptive similarity-based surrogate model as a fitness inheritance strategy to reduce computationally expensive fitness evaluations. Gaussian similarity measurement, which considers the ruggedness of the landscape, is proposed to adaptively regulate the similarity in order to improve the accuracy of the inheritance fitness values. Empirical results on three traditional benchmark problems with 5, 10, 20, and 30 decision variables and on the CEC'13 test functions with 30 decision variables demonstrate the high efficiency and effectiveness of the proposed algorithm in that it can obtain better or competitive solutions compared to the state-of-the-art algorithms under a limited computational budget.**



***Keywords—Fitness inheritance; Fitness estimation; similarity; Computationally expensive optimization; Particle swarm optimization***


## I. INTRODUCTION

Evolutionary algorithms (EAs) have much strength for optimization, for example, no requirement on differentiability of the objective functions, ease implementation, and better global search capability, compared to traditional single-point optimization algorithms. For these reasons, EAs have been applied to many real-world applications and have emerged as well-established and powerful optimization tools, especially in the field of engineering design. However, many real-world engineering optimization problems involve complex computational simulations, which may take a few minutes, hours or even days to perform one simulation [1-4], such as computational fluid dynamics [5], finite element method computations [6], and aerodynamic shape optimization [7], just to name a few. As EAs usually need a large number of fitness evaluations to obtain an acceptable solution, it is essential to reduce the number of fitness evaluations due to the constraint of limited computational resources.

To tackle computationally expensive optimization problems, use of computationally cheap fitness approximation to replace the real expensive fitness evaluations is a common approach [8-10]. The most popular method of fitness approximation is surrogate models (also called meta-models) [11-14]. Commonly used surrogate models include the polynomial method [15], the kriging method (Gaussian process) [16], artificial neural networks (ANN) [17], radial basis functions (RBF) [2, 18], and support vector machines (SVM) [19]. A comprehensive description of these methods can be found in [8-10]. Among various fitness approximation methods, fitness inheritance techniques are a generic approach that does not rely on the use of surrogate models. One clear benefit of fitness inheritance is that its computational expense is much less than that of training a surrogate model [9]. In this paper, we propose to use a relatively simple fitness approximation method to determine which individual will be evaluated using the real objective function, and which one will be reliably estimated using fitness inheritance. The idea of fitness inheritance was originally proposed by Smith [20]. It is an intuitive idea assuming that the fitness of an individual can be derived from its parents. Two types of fitness inheritance strategies have been proposed, one is known as the averaged inheritance and the other weighted inheritance. Salami and Hendtlass [21] introduced a "fast evolutionary algorithm", in


This work was supported in part by National Natural Science Foundation of China (Grant No. 61472269, 61403272 and 61403271), and the State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University, China.

which a fitness value and an associated reliability value were assigned to each new individual, and the individual is evaluated using the real fitness function only when its reliability value is below a threshold. As mentioned in [22], in many real-world applications, the fitness of offspring cannot be reliably estimated from the fitness of their parents, as they are not necessarily similar to their parents. Instead, a newly generated offspring individual may be similar to those in its vicinity. For example in a particle swarm optimization (PSO) algorithm, Cui [23] proposed a fitness inheritance model based on the similarity or credibility to an individual's previous position. In [24], a fitness estimation strategy was proposed for particle swarm optimization, called FESPSO, based on the positional relationship between two individuals at same iteration. To further reduce the number of fitness evaluations for FESPSO, Sun et al [25] introduced a similarity in the fitness estimation strategy method later. Kim [26] used a clustering technique to group similar individuals and the individuals that are closest to cluster center will be evaluated using the real objective function, while the fitness of the rest individuals will be inherited from those with their fitness being evaluated using the real objective function.

Reyes-Sierra and Coello [27] suggested a fitness inheritance model that also used the average method according to the Euclidean distances between individuals in multi-objective particle swarm optimization, which was shown to have reduced the time of optimization and improved the performance in solving the classical test functions. Gomide [28] utilized a fuzzy clustering method with fuzzy adaptive clustering of c-means. In each cluster, the fitness of the most representative individual was calculated using the real fitness function and the fitness of the other individuals is estimated using a fitness inheritance method weighted by the Euclidean distance and the membership degree. Fonseca et al. [29] introduced a similarity-based surrogate model based upon the k-nearest neighbors method (KNN). The similarity proposed by Fonseca is also based on the Euclidean distance among individuals. Subsequently, Fonseca et al [30] applied three different types of adaptive value inheritance models, including the averaged inheritance model, weighted inheritance model, and parent inheritance model, in a real-coded genetic algorithm for comparisons. Jin and Sendhoff [31] also used the k-means method applied to group the individuals of a population into a number of clusters. For each cluster, only the individual which closest to the cluster center is evaluated and the fitness of other individuals are estimated using a neural network ensemble.

The main issue that needs to be addressed when employing the fitness inheritance method is the accuracy. In most fitness inheritance strategies, the fitness of an individual is estimated based on that of other individuals according to the positional relationship of these two individuals in the decision space using the Euclidean distance, and the smaller Euclidean distance between two individuals' positions is, the higher the degree of similarity between them. However, fitness landscape analysis proved that the distance metrics will become less meaningful when the dimensionality of the decision space significantly increases [32]. For example, the fitness distance correlation (FDC) [33] indicated that there is no simple relationship between the fitness values of two individuals and the distance between two individuals. Different from the above-mentioned approaches, Davarynejad [34-36] proposed a method of adaptive fuzzy fitness granulation using the fitness of the individuals in the pool for fitness estimation. When the center of the granule has a high fitness, the radius of the granules will be small, and the new individual's fitness is more likely to be computed in this granule. On the other hand, more individuals will be estimated using fitness inheritance in the granule when its center has a poor fitness. Nonetheless, for fuzzy granule rules, an individual may find itself in more than one granule, and granules with Gaussian measures of similarity was proposed to adjust its the width so that the new individual will inherit the fitness from a granule with a poor fitness yet far from this individual, while not inherit the fitness from a granule with good fitness but near the new individual[37]. Therefore, the above-mentioned approaches cannot guarantee the accuracy in fitness estimation. In order to obtain more accurate estimation using a fitness inheritance method, the impact of the fitness landscape on the similarity between two individuals must be considered.

In this paper, an adaptive similarity-based surrogate model is proposed to reduce the computational cost for time-consuming optimization problems, which takes into account the similarity on both smooth fitness landscapes (with small fitness changes in the neighborhood) and rough fitness landscapes (with high fitness variations in the neighborhood). The degree of similarity between individuals will be decreased when the local fitness landscape is believed to be rugged. By utilizing an adaptive similarity-based surrogate models in order to adjust the frequency of using fitness estimation, we aim to achieve a balance between the frequency and the accuracy of fitness estimation, thereby ensuring the estimation accuracy while reducing the required fitness evaluations using the real fitness function.

The remainder of this paper is organized as follows. Section II gives a detailed description of the proposed self-adaptive similarity-based surrogate model and the surrogate-assisted PSO. Experimental results on two groups of test problems are presented and discussed in Section III. Section IV gives a summary of this paper and some discussions for future work.

## II. A SELF-ADAPTIVE SIMILARITY-BASED SURROGATE MODEL

As previously discussed, the main issue with utilizing a fitness inheritance method is to enhance the accuracy of fitness estimation. In the following, we first provide an illustrative example showing why most existing fitness inheritance methods using the Euclidean distance as a measure for similarity may fail. After that, we introduce a new fitness inheritance method.

Fig. 1 gives an illustrative example to show the problem in fitness estimation when a similarity degree is utilized in a one-dimensional Michalewicz function. In this example, there are three solutions $x_1$, $x_2$ and $x_3$, in which the fitness value of $x_1$ and $x_3$ are known and evaluated using the real fitness function. $x_2$ is a new candidate solution whose fitness is to be estimated. The similarity between $x_1$ and $x_2$ is $s_1$, and the similarity between $x_3$ and $x_2$ is $s_3$. Given a similarity threshold $s_0$, $x_2$ will be estimated if and only if $s_1$ or $s_3$ is greater than $s_0$. According to the Euclidean distance, a smaller distance between two points means a higher degree of similarity. If $s_1 < s_0 < s_3$, the fitness of $x_2(y_2)$ can be inherited from that of $x_3$. However, as we can see from Fig. 1, the fitness landscape is smoother between $x_1$ and $x_2$ than between $x_3$ and $x_2$. Therefore, in order to reliably estimate the fitness of $x_2$, a new similarity metric is needed, which takes the ruggedness of the local landscape in addition to the Euclidean distance, so that the fitness of $x_1$ will be used for estimating the fitness of $x_2$, then the fitness estimation error will be smaller than the fitness of $x_3$ used for estimating the fitness of $x_2$ .

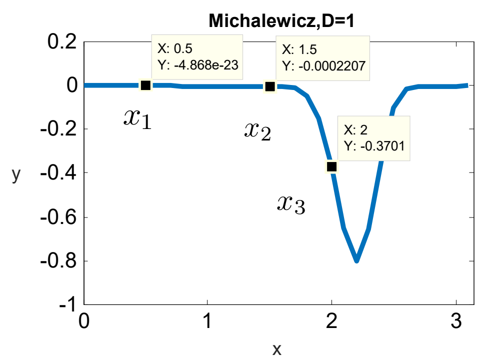


Fig. 1. An example of three solutions in a one-dimensional Michalewicz function

To address the above issue in fitness estimation, this paper proposes a Gaussian similarity measure that is able to adjust the degree of similarity according to the local ruggedness of the fitness landscape. It is assumed that the similarity and the local ruggedness are negatively correlated.

*A. Self-adaptive similarity-based fitness approximation strategy*

Without loss of generality, we consider the following minimization problem:

$$\begin{aligned} &Min f(\mathbf{x}) \\ &Subject\ to: \mathbf{x}_l \leq \mathbf{x} \leq \mathbf{x}_u \end{aligned} \tag{1}$$

where $x \in \mathbb{R}^n$ is the feasible solution set, $n$ denotes the dimensionality of the search space. Below are the main steps of our proposed method for PSO algorithms.

**Step 1:** Initialize a population, perform fitness evaluation using the real objective function for all individuals.
**Step 2:** Determine the personal best position and global best position of PSO. If the fitness of the personal best position or global best position is estimated, then re-evaluate their fitness using real objective function.
**Step 3:** Update the velocity and position of the swarm. For each individual i (i=1,2,…,n), set a neighborhood size of $\mathbf{x}_i$, and select $k$ neighboring individuals. If $k=0$, evaluate the fitness of $\mathbf{x}_i$ using the real objective function, and go to Step 6; otherwise go to Step 4.
**Step 4:** Calculate the Gaussian similarity $\mu_i^r$ between $\mathbf{x}_i$ and its neighbor $\mathbf{x}_r$.
**Step 5:** Estimate the fitness value of $\mathbf{x}_i$ using the fitness inheritance strategy.
**Step 6:** Repeat steps 2-5 if the stop condition is not satisfied, otherwise stop and exit.

*B. Gaussian Similarity*

Gaussian similarity, which uses a Gaussian function, is a similarity matrix that commonly be used as a clustering calculation method for similarity models,

$$s(i,r) = exp(-\frac{d^2(\mathbf{x}_i,\mathbf{x}_r)}{2\sigma^2}) \tag{2}$$

In (2), $d^2(\mathbf{x}_i,\mathbf{x}_r)$ is the Euclidean distance between two samples $i$ and $r$, and $\sigma$ is the metric parameter. The most important issue in the Gaussian similarity is the selection of metrics. Different metric parameters will lead to different clustering results. In this paper, we incorporate the roughness of the local fitness landscape to help adjust the metric of the Gaussian similarity. The similarity will be self-adaptively calculated according to the changes in the local roughness, which will reduce the estimation errors in fitness inheritance. In the following, we will elaborate the proposed self-adaptive similarity-based surrogate model in detail.

*C. Defining a measure for local roughness $\delta$*

It is conceivable that the local roughness where individual $i$ is located will influence the accuracy of fitness estimation much more than the global roughness of the fitness landscape. In this paper, the local roughness will be considered when the fitness of individual $i$ is to be estimated.

Suppose the radius of neighborhood of individual $i$ is $\mathbf{R}_i$, the neighboring $U(\mathbf{x}_i,\mathbf{R}_i)$ is defined by $[\mathbf{x}_i-\mathbf{R}_i,\mathbf{x}_i+\mathbf{R}_i]$. Apparently, the size of radius $\mathbf{R}_i$ will directly affects the number of individuals in the neighborhood. If the neighborhood is too small, there might not be enough neighboring individuals around individual $i$ , and the Gaussian similarity cannot be constructed. If the neighborhood is appropriately set, there will be sufficient neighbors and the fitness of individual $i$ will be inherited from its neighbors. At the earlier generations, the global search ability is paid more attention to the algorithm, so a relatively large radius $R_i$ is adopted. Then, $R_i$ will be decreased to improve the accuracy of fitness estimation at later generations. In this paper, an adaptive radius is proposed according to the number of the individuals distributed in the neighborhood of individual $i$ . The upper and lower bounds of the neighborhood on $j$-th dimension of individual $i$ are defined as $cx_{jmax}$ and $cx_{j\min}$, that is:

$$cx_{jmax} = \max\left(\{x_{ij}, i=1,,2,...,n\}\right) \tag{3}$$

$$cx_{j\min} = \min\left(\{x_{ij}, i=1,,2,...,n\}\right) \tag{4}$$

where $n$ is the population size. Correspondingly, the value of $R_i$ on $j$-th dimension is set as:

$$R_{ij} = \alpha\left|cx_{j\max} - cx_{j\min}\right| \tag{5}$$

In(5), $\alpha$ is called the convergence factor. Apparently, $\mathbf{R}_i = \{R_{ij}, j = 1, 2, \ldots, l\}$ is different on each dimension, $l$ is the number of variables.

Suppose that there are $m$ individuals whose fitness is evaluated using the real objective function, we save their positions and corresponding fitness into an archive $\mathbf{X_E}= \{\mathbf{x}_{ev}, f(\mathbf{x}_{ev}), ev = 1,..., m\}$ , which will enable individuals to find their neighbors in their neighboring space $U(\mathbf{x}_i, \mathbf{R_i})$. The neighbors of each individual $i$ will be also saved in an archive $\mathbf{X}_{ir} = \{\mathbf{x}_r, f(\mathbf{x}_r), r = 1, ..., k\}$. $\mathbf{x}_r$ and $f(\mathbf{x}_r)$ represent the position and fitness of $r$-th neighbor of individual $i$, respectively. Eq.(6) defines the method to calculate the local roughness $\delta$ around individual $i$. Obviously, if the fitness variation among the neighbors is large, the local roughness $\delta$ will be relatively large, and vice versa.

$$\delta = \frac{\max\limits_{r\in\{1,,2,\ldots,k\}} (f(\mathbf{x}_r)) - \min\limits_{r\in\{1,,2,\ldots,k\}} (f(\mathbf{x}_r))}{\max\limits_{ev\in\{1,,2,\ldots,m\}} (f(\mathbf{x}_{ev})) - \min\limits_{ev\in\{1,,2,\ldots,m\}} (f(\mathbf{x}_{ev}))} \quad (6)$$

We can easily find that $\delta \in [0, 1]$.

*D. Defining metric parameter* $\sigma$

The metric parameter $\sigma$ plays an important role in the Gaussian similarity. We firstly consider the value of the proper metric parameter $\sigma$ on one-dimensional problems when the fitness of an individual will be estimated. The similarity $\mu_i^r$ between the interested individual $i$ and its neighboring individuals $r$ is defined according to (2).

$$\mu_i^r = e^{-\frac{(x_r - x_i)^2}{2\sigma^2}}, r = 1, ..., k \quad (7)$$

The next step is to select those neighbors with their similarity $\mu_i^r$ higher than the threshold $\theta$. All neighbors, which are called estimation neighbors, will be used to estimate the fitness of $x_i$ using the inheritance method. The values of both parameters, $\sigma$ and $\theta$, are key factors in selecting estimation neighbors.

We divide individuals around individual $i$ into three parts. According to the 3σ principle of Gaussian function and the similarity defined in (7), the position of $x_i$ is the center of the neighborhood, and individuals in the range $[x_i - \sigma, x_i + \sigma]$ are called core neighbors, individuals located in the range $[x_i - 2\sigma, x_i + 2\sigma]$ are called outskirt neighbors, and those located in the range $[x_i - 3\sigma, x_i + 3\sigma]$ are called weak outskirt neighbors. When the core neighbors are used to estimate the fitness of individual $i$, the threshold for the similarity level is set to $\theta = e^{-\frac{1}{2}} \approx 0.6$. Then the value of $\sigma$ will directly affects the range of the core neighbors $[x_i - \sigma, x_i + \sigma]$. Therefore, the value of $\sigma$ value will correspondingly affects the similarity among individuals. In this paper, a local roughness of the fitness function $\delta$ is proposed to be incorporated in the setting of metric parameters, which aims to adjust the size of the neighborhood the core neighbors are located. The size of this space will be shrunken as the local roughness increases. Obviously, if the roughness is low, the core neighbors will be selected from a wide range of individuals around individual $i$, as a result, the fitness of individual $i$ will be more likely to be estimated; otherwise, the fitness of individual $i$ will be more likely to be evaluated using the real objective function. Eq. (8) gives the definition of $\sigma$ proposed in this paper:

$$\sigma = \frac{\beta}{\delta} R_i \quad (8)$$

where $\beta$ is the convergence factor, and $\delta$ represents the local roughness of the fitness function. Because $\delta \in [0, 1]$, we can deduce that when the variation on fitness of neighbors is small in the neighboring space, it is possible for $\delta$ to be near zero with $\sigma$ approaching the infinity. Therefore, we give an upper bound of $\sigma$, $\sigma \le R_i$, to set $\sigma$ in a limited range. Then we can get $\sigma = \frac{\beta}{\delta} R_i \le R_i$ which we can derive that $\delta \ge \beta$; as a result, when $\delta \le \beta$, we have $\delta = \beta$, and $\delta \in [\beta, 1]$, and subsequently, $\sigma \in [\beta R_i, R_i]$.

When the individuals are in multi-dimensional space, $\sigma$ and $\mathbf{R}_i$ are $n$ dimensional vectors with the same number of dimensions as $\mathbf{x}_i$, and in the $j$th dimension:

$$\sigma_j = \frac{\beta}{\delta} R_{ij} \quad (9)$$

The similarity $\mu_{ij}^r$ between individual $i$ and its neighbor $r$ on $j$-th dimension is given in the following:

$$\mu_{ij}^r = e^{-\frac{(x_{rj} - x_{ij})^2}{2\sigma_j{}^2}}, r = 1, ..., k \quad (10)$$

The similarity $\mu_i^r$ between $\mathbf{x}_i$ and $\mathbf{x}_r$, then, will be the average of the similarity on all dimension:

$$\mu_i^r = \frac{\sum_{j=1}^{l} \mu_{ij}^r}{l}, r = 1, ..., k \quad (11)$$

If the core neighbors are used for estimating the fitness of $\mathbf{x}_i$ in a weighed manner, the similarity threshold should be $\theta = e^{-\frac{1}{2}} \approx 0.60$. We suppose that individuals with a higher fitness value are commonly surrounded by more individuals, so when the average fitness value of the neighboring individuals is relatively high, the individual of interest is more likely to have a higher fitness value. Especially at the end of evolution, the population converges gradually, and the individual whose fitness is to be estimated is often located in an area of high fitness. As a result, the number $k'$ is relatively high in the core neighbors' archive $\mathbf{X}_{kr} = \{\mathbf{x}_{kr}, f(\mathbf{x}_{kr}), r = 1, ..., k'\}$. Therefore, the fitness of the neighbors should be also considered when selecting neighbors for fitness estimation. In this paper, the threshold of similarity $\theta$ is constructed based on the averaged fitness value of the core neighbors, denoted as $mean\,(f(\mathbf{x}_{kr}))$, and the threshold $\theta$ is adapted according to the value of $mean\,(f(\mathbf{x}_r))$:

$$mean(f(\mathbf{x}_{kr})) = \frac{\sum_{kr=1}^{k'} f(\mathbf{x}_{kr})}{k'}, kr = 1, ..., k' \quad (12)$$

$$\theta = 0.9 - 0.3 * \frac{mean\,(f(\mathbf{x}_{kr})) - \min\limits_{ev\in\{1,,2,\ldots,n\}} (f(\mathbf{x}_{ev}))}{\max\limits_{ev\in\{1,,2,\ldots,n\}} (f(\mathbf{x}_{ev})) - \min\limits_{ev\in\{1,,2,\ldots,n\}} (f(\mathbf{x}_{ev}))} \quad (13)$$

From (13) we can see that $\theta \in [0.6, 0.9]$. By neighboring clusters from the core individuals, those neighbors used for fitness estimation are selected based on their merits according to $mean\left(f\left(\mathbf{x}_{kr}\right)\right)$. When the average fitness $mean\left(f\left(\mathbf{x}_{kr}\right)\right)$ of the individuals is weak in the neighborhood, the fitness of the neighbors of $x_i$ is undesirable overall, and $\theta$ will be reduced accordingly, making the interested individual more likely to be estimated. On the other hand, when the average fitness $mean\left(f\left(\mathbf{x}_{kr}\right)\right)$ of the individuals is large in the neighborhood, the value of $\theta$ will be increased, thus increasing the likelihood for the individual's fitness to be evaluated using the real objective function.

### *E. Fitness inheritance strategy*

The individuals in the set $\mathbf{X}_r$ that satisfy $\mu_i^r > \theta$ constitute the set $SN = \{\mathbf{sn}_1, ...\mathbf{sn}_t\}$. The similarity between $i$ and individuals from the set $SN$ is $\{\mu_1, ...\mu_t\}$. Suppose that there are potentially more good individuals around the current best individual, there should be the historically best individuals found previously in the set $SN$, the new individual $i$ is also considered to potentially locate at the best position. Thus, the fitness of individual $i$ is obtained with the following procedure:

- If the historically best individuals are in the set $SN$, the fitness of individual $i$ is evaluated using the real objective function.
- If $1 \le t \le T$,

$$f\left(\mathbf{x}_i\right) = \frac{\sum_{l=1}^{t}(\mu_l)^2 f(\mathbf{sn}_l)}{\sum_{l=1}^{t}(\mu_l)^2}, 1 \le t \le T \tag{14}$$

- If $t > T$, the similarities between individuals and current individual are ranked according to a decreasing order of their fitness, and $T$ individuals were selected to assign $f\left(x_i\right)$ value in a weighted manner. $T$ is the maximum number of neighboring individuals used for fitness estimation.

$$f\left(x_i\right) = \frac{\sum_{l=1}^{T}(\mu_l)^2 f(sn_l)}{\sum_{l=1}^{T}(\mu_l)^2}, t > T \tag{15}$$

## III. EXPERIMENTAL STUDIES

In order to examine the performance of the proposed self-adaptive similarity-based fitness approximation for particle swarm optimization, called SS-Based FAPSO, for computationally expensive problems, the algorithm is tested on two sets of widely used benchmark problems. In addition, the efficiency and effectiveness of SS-Based FAPSO is also demonstrated by comparing to a few existing fitness inheritance strategies proposed for PSO.

### *A. Traditional Optimization Benchmark Functions*

To investigate the efficiency of our proposed approach, three traditional optimization benchmark functions suggested in [35] and 28 CEC functions are adopted. Four sets of dimensions are utilized for three traditional optimization benchmark functions:$D = 5, 10, 20 and 30$. The characteristics of the three benchmark functions （Griewank, Rastrigin and Ackley）, referring to [35] for more details. These three benchmark functions are scalable and are commonly used to assess the performance of optimization algorithms. They have some intriguing features which most optimization algorithms find hard to deal with. All the compared algorithms are run for 10 independent times on each test problem in Matlab R2014a. The code for the traditional PSO is provided by [38].

In this paper, The PSO and the k-nearest neighbors method for PSO, called KNNPSO, are adopted to compare on the performance with SS-Based FAPSO under the same fixed fitness evaluations. In KNNPSO, we select $k$ as the number of the closest neighbors to the individuals to be estimated for performing the calculation of fitness inheritance. The parameters for the KNNPSO method are set as follows: $k$ is the number of the closest neighbors in radius $\mathbf{R}$, which is the same as the radius set for SS-Based FAPSO. From (5) and (6) with $\alpha = 0.5$, fitness inheritance is performed for the individual of interest when$10 \gg k \ge 1$, but only the 10 closest neighbors to $x_i$ are selected for estimating the fitness of individual $i$ using the inheritance strategy when $k \ge 10$. The parameters of KNNPSO and SS-Based FAPSO are set as follows in the experiment: The size of the swarm is 20, the cognitive and social parameters are both set to 2.05, the velocity $v$ has the same range as the position $x$ on each dimension, and the maximum number of real fitness evaluations is set to $ME = 100 * D$, which is the same as those used in [35].

Figs. 2-4 shows the results obtained by PSO, KNNPSO, and SS-Based FAPSO with the same number of fitness evaluations. The algorithm stops when the fitness evaluations reach the maximum value of fitness evaluations using the real objective function. A number of experiments were conducted and showed that if the value of $\alpha$ in (5) is set too small, the size of the neighborhood will be small too, resulting in an insufficient number of the neighboring individuals. If the value of $\beta$ in (9) is set too small, the number of the core individuals will be small too, which will then greatly reduce the number of fitness estimations. However, if the value of $\alpha$ is too large, there will be an excessive number of neighboring individuals involved in fitness evaluations, which will increase the time of fitness approximation. In this paper, $\sigma \in [\beta R_i, R_i]$ was derived when defining $\sigma$, if the value of $\beta$ is set too large, the range of $\sigma$'s variation will be much small, and it will not be able to properly reflect the impact of the local roughness on the selection of neighboring individuals used for fitness estimation. In addition, if the range for selecting core neighbors is too large, it will lower the accuracy of the fitness estimation. The values of the parameters $\alpha$ in (5) and $\beta$ in (9) are both set by trials and errors. Therefore, in this experiment, $\alpha = 0.5$, and $\beta = 0.3$. Fonseca [29] chose the different largest neighboring number of individuals $T = 1, 2, 5, 10, 15$ to verify the results. The results obtained when $T = 10, 15$ are superior to those obtained when $T = 1, 2, 5$. To improve the accuracy of fitness inheritance and the computational efficiency, the maximum number of neighboring individuals involved in fitness estimation $T$ is set to be 10.

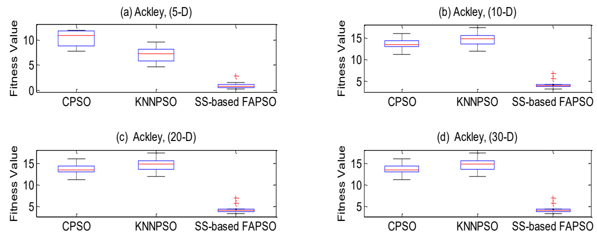


Fig. 2. CPSO,KNNPSO and SS-based FAPSO Box plot of the best fitness in the final generation for Ackley function

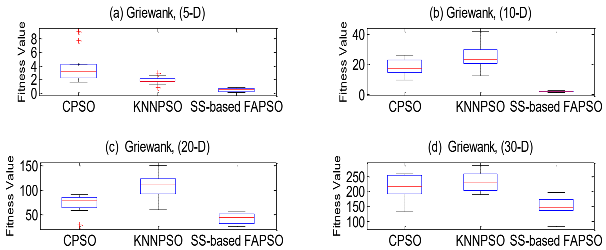


Fig. 3. CPSO,KNNPSO and SS-based FAPSO Boxplots of the best fitness in the final generation for Rastrigin function

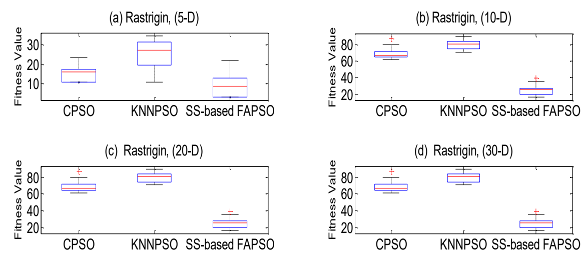


Fig. 4. CPSO,KNNPSO and SS-based FAPSO Box plot of the best fitness in the final generation for Griewank function

To further demonstrate the performance of fitness inheritance proposed in our method, Figs. 2-4 give the boxplots of the best fitness in the final generation of CPSO, KNNPSO and SS-based FAPSO on the three functions. Fig. 5 plots the convergence profile of PSO, KNNPSO, and SS-Based FAPSO on benchmark problems （Griewank, Rastrigin and Ackley）. As we can see from Figs. 2-4, the proposed SS-Based FAPSO method has fewer outliers compared to the other two methods in 10 independent runs, indicating that it performs more robust than the other two algorithms. It can be see that our proposed SS-Based FAPSO method obtained best results nearly in all instances in terms of the best, the worst and the mean results of the optimal results averaged over ten independently runs on three functions of different dimensions compared to PSO and KNNPSO. From Fig. 5, we can see that the proposed SS-Based FAPSO method converges much faster than the other two methods. Although, initially, the SS-Based FAPSO method showed slightly worse or comparable performance on converge (e.g., in the Rastrigin function in 30 dimensions), it shows much better convergence capability as the number of fitness evaluations increases (i.e. in the later stages).

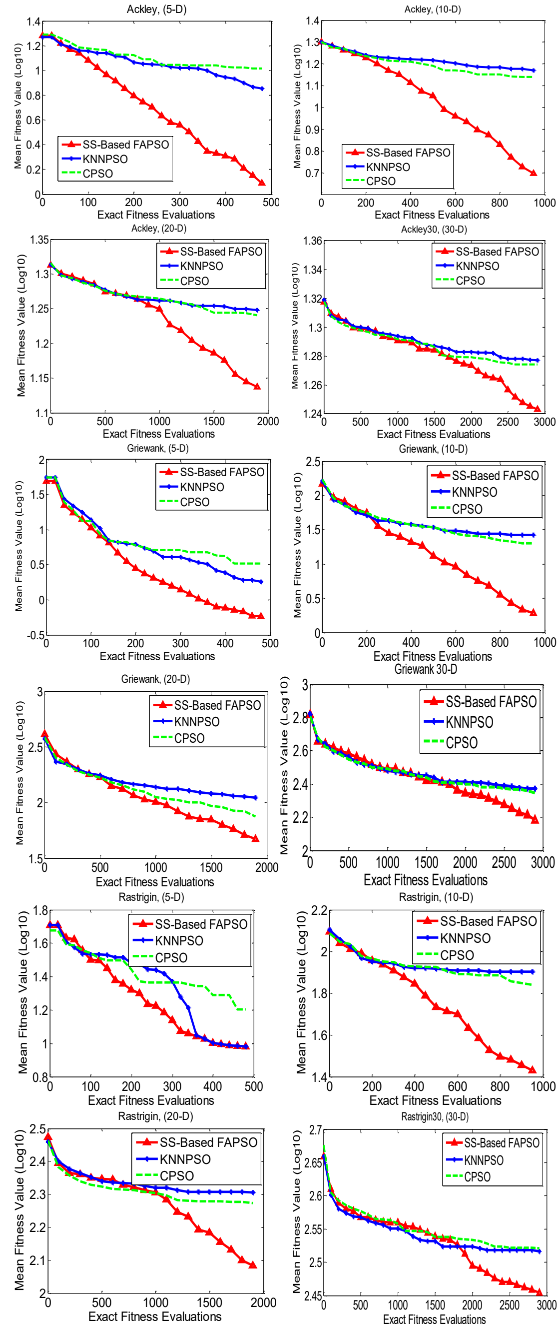


Fig. 5. Convergence profiles of the CPSO,KNNPSO and SS-Based FAPSO on the three benchmark problems

## *B. CEC' 13 Function Tests*

TABLE I. THE DETAILS OF FUNCTIONS ON WHICH FESPSO AND SS-BASED FAPSO HAVE YIELDED BEST PERFORMANCE.

| Function number | Detail |
|---|---|
| ***F1*** | Sphere Function(f1) |
| ***F2*** | Rotated Bent Cigar Func Function(f3) |
| ***F3*** | Rotated Rosenbrock's Function(f6) |
| ***F4*** | Rotated Schaffers F7 Function |
| ***F5*** | Composition Function (n=5,Rotated)(f21) |
| ***F6*** | Composition Function 8 (n=5,Rotated) |

To further compare the performance of these different methods, the fitness estimation strategy for particle swarm optimization proposed by Sun et al. (2013), called FESPSO, was adopted for comparison. Table II presents the experimental results on CEC'13 testing functions of a dimension of 30 obtained by PSO, KNNPSO, and FESPSO, with a maximum of 10,000 real fitness evaluations. Table I lists the details of the test functions, where we can see that F1 to F2 are unimodal functions, F3 to F4 are multi-modal functions, and F5 to F6 are complex combinations of unimodal and multi-modal functions.

TABLE II. COMPARATIVE RESULTS ON THE 28 CEC'13 TEST FUNCTIONS

| Func | Approach | Best_fit | Worst_fit | Fit_mean | std_fit | opt |
|---|---|---|---|---|---|---|
| ***F1*** | ***PSO*** | 7.3442E+02 | 4.1341E+03 | 2.1467E+03 | 1.1623E+03 | –1.4000E+03 |
| | ***KNNPSO*** | 1.6618E+03 | 3.9159E+03 | 2.7568E+03 | 7.4661E+02 | |
| | ***FESPSO*** | 6.7716E+02 | 2.1454E+04 | 8.5867E+03 | 7.4134E+03 | |
| | ***SS-Based FAPSO*** | **-1.3730E+03** | **-1.1468E+03** | **-1.2947E+03** | **6.7340E+01** | |
| ***F2*** | ***PSO*** | 1.1036E+10 | 3.3524E+10 | 1.7215E+10 | 6.5599E+09 | –1.2000E+03 |
| | ***KNNPSO*** | 1.2686E+10 | 4.0761E+10 | 2.0541E+10 | 9.3043E+09 | |
| | ***FESPSO*** | 4.3694E+10 | 2.1760E+14 | 2.1891E+13 | 6.8765E+13 | |
| | ***SS-Based FAPSO*** | **1.3160E+09** | **1.7470E+10** | **7.7893E+09** | **4.6107E+09** | |
| ***F3*** | ***PSO*** | -6.0964E+02 | -4.2390E+02 | -5.0822E+02 | 5.1481E+01 | –9.0000E+02 |
| | ***KNNPSO*** | -6.4413E+02 | -3.8613E+02 | -5.4236E+02 | 8.3366E+01 | |
| | ***FESPSO*** | -7.1181E+02 | 7.2392E+02 | 1.7961E+02 | 4.6235E+02 | |
| | ***SS-Based FAPSO*** | **-8.3849E+02** | **-7.1419E+02** | **-7.9001E+02** | **3.8836E+01** | |
| ***F4*** | ***PSO*** | -6.9753E+02 | -5.7398E+02 | -6.6374E+02 | 3.5328E+01 | –8.0000E+02 |
| | ***KNNPSO*** | -7.0807E+02 | -6.0298E+02 | -6.6711E+02 | **2.7688E+01** | |
| | ***FESPSO*** | -5.0243E+02 | 7.0854E+04 | 6.8009E+03 | 2.2507E+04 | |
| | ***SS-Based FAPSO*** | **-7.1658E+02** | **-5.5708E+02** | **-6.5522E+02** | 6.3001E+01 | |
| ***F5*** | ***PSO*** | 1.5673E+03 | 2.1877E+03 | 1.8976E+03 | 2.1048E+02 | 7.0000E+02 |
| | ***KNNPSO*** | 1.5416E+03 | 2.6750E+03 | 2.0670E+03 | 4.2450E+02 | |
| | ***FESPSO*** | 1.0019E+03 | 1.5132E+03 | 1.2173E+03 | 1.7606E+02 | |
| | ***SS-Based FAPSO*** | **9.9914E+02** | **1.2192E+03** | **1.1298E+03** | **6.6561E+01** | |
| ***F6*** | ***PSO*** | 2.9499E+03 | 3.8629E+03 | 3.3923E+03 | 2.7246E+02 | 1.4000E+03 |
| | ***KNNPSO*** | 3.2284E+03 | 4.1382E+03 | 3.5554E+03 | **3.2631E+02** | |
| | ***FESPSO*** | 4.0730E+03 | 5.4601E+03 | 4.7597E+03 | 4.5402E+02 | |
| | ***SS-Based FAPSO*** | **1.8140E+03** | **3.1845E+03** | **2.0548E+03** | 4.0516E+02 | |

As the results shown in Table II indicate, SS-Based FAPSO method has obtained better results than the compared algorithms on both unimodal and multi-modal test functions. SS-Based FAPSO shows significantly better overall performance on F1. In F4 and F6, the standard deviation obtained by SS-Based FAPSO was slightly inferior to the compared methods.

## IV. CONCLUSION AND FUTURE WORK

This paper proposed a new self-adaptive similarity-based surrogate model for fitness estimation in PSO to solving computationally expensive problems. In this model, the radius of the neighborhood to be selected for estimating the fitness of the individuals is self-adaptive. The proposed method requires a lower number of fitness calculations using the real fitness function while a high accuracy in fitness estimations. Our future work includes the development of a more accurate method in evaluating local roughness of the fitness landscape. Furthermore, the adaptive similarity-based surrogate model proposed in this study can be seen as a local surrogate model. Hence, our future work will consider the situations where the condition for fitness inheritance is not satisfied. In these cases, we could combine global surrogate models with the local estimation method proposed in this work to provide more reliable fitness estimations.

## V. APPENDIX

In the canonical PSO, when searching in a D-dimensional hyperspace, each particle $i$ has a velocity $\mathrm{V_i} = [v_{i1}, v_{i2}, ..., v_{iD}]$ and a position $\mathrm{X_i} = [x_{i1}, x_{i2}, ..., x_{iD}]$. The vectors $\mathrm{V_i}$ and $\mathrm{X_i}$ are initialized randomly and then updated by (16) and (17) through the guidance of its personal best position $\mathrm{P_i}$, and the global best position $\mathrm{P_n}$:

$$\mathrm{V_i}^{(t+1)} = \omega \mathrm{V}_i^{(t)} + c_1 r_1 (\mathrm{P}_i^{(t)} - \mathrm{X}_i^{(t)}) + c_2 r_2 (\mathrm{P_n}^{(t)} - \mathrm{X}_i^{(t)}) \quad (16)$$

$$\mathrm{X}_i^{(t+1)} = \mathrm{X}_i^{(t)} + \mathrm{V}_i^{(t+1)} \quad (17)$$

where $\mathrm{X}_i^{(t)}$ and $\mathrm{V}_i^{(t)}$ are the position and velocity of $i$ particle at generation $t$ respectively., The Coefficients $\omega$ is called the inertia weight, $c_1$ and $c_2$ are acceleration parameters which are commonly set to 2.0. $r_1$ and $r_2$ are two diagonal matrix whose diagonal elements are random numbers uniformly generated within the range of [0, 1].

## ACKNOWLEDGMENT

This work was supported in part by National Natural Science Foundation of China (Grant No. 61472269, 61403272 and 61403271), and the State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University, China.

## REFERENCES

[1] D. Lim, Y. Jin, Y.-S. Ong, and B. Sendhoff, "Generalizing surrogate-assisted evolutionary computation," *Evolutionary Computation, IEEE Transactions on,* vol. 14, 2010 , pp. 329-355.

[2] D. Lim, Y.-S. Ong, Y. Jin, and B. Sendhoff, "Trusted evolutionary algorithm," in *Evolutionary Computation, 2006. CEC 2006. IEEE Congress on*, 2006, pp. 149-156.

[3] Y. S. Ong, P. Nair, A. Keane, and K. Wong, "Surrogate-assisted evolutionary optimization frameworks for high-fidelity engineering design problems," in *Knowledge Incorporation in Evolutionary Computation*, ed: Springer, 2005, pp. 307-331.

[4] Y. S. Ong, P. B. Nair, and A. J. Keane, "Evolutionary optimization of computationally expensive problems via surrogate modeling," *AIAA journal,* vol. 41, 2003, pp. 687-696.

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

[6] K. S. Won and T. Ray, "A framework for design optimization using surrogates," *Engineering optimization,* vol. 37, 2005, pp. 685-703.

[7] M. K. Karakasis, A. P. Giotis, and K. C. Giannakoglou, "Inexact information aided, low - cost, distributed genetic algorithms for aerodynamic shape optimization, " *International Journal for Numerical Methods in Fluids ,* vol. 43, 2003 ,pp. 1149-1166.

[8] Y. Jin, "A comprehensive survey of fitness approximation in evolutionary computation," *Soft computing,* vol. 9, 2005 , pp. 3-12.

[9] L. Shi and K. Rasheed, "A survey of fitness approximation methods applied in evolutionary algorithms," in *Computational intelligence in expensive optimization problems*, ed: Springer, 2010, pp. 3-28.

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

[11] E. Sanchez, S. Pintos, and N. V. Queipo, "Toward an optimal ensemble of kernel-based approximations with engineering applications," *Structural and multidisciplinary optimization,* vol. 36, 2008, pp. 247-261.

[12] D. Lim, Y.-S. Ong, Y. Jin, and B. Sendhoff, "A study on metamodeling techniques, ensembles, and multi-surrogates in evolutionary computation," in *Proceedings of the 9th annual conference on Genetic and evolutionary computation*, 2007, pp. 1288-1295.

[13] A. Samad, K.-Y. Kim, T. Goel, R. T. Haftka, and W. Shyy, "Multiple surrogate modeling for axial compressor blade shape optimization," *Journal of Propulsion and Power,* vol. 24, 2008 , pp. 301-310.

[14] C. Sun, Y. Jin, J. Zeng, and Y. Yu, "A two-layer surrogate-assisted particle swarm optimization algorithm," *Soft Computing,* vol. 19, 2015, pp. 1461-1475.

[15] I. Paenke, J. Branke, and Y. Jin, "Efficient search for robust solutions by means of evolutionary algorithms and fitness approximation," *Evolutionary Computation, IEEE Transactions on,* vol. 10, 2006, pp. 405-420.

[16] 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, 2014,pp. 180-192.

[17] S. Ferrari and R. F. Stengel, "Smooth function approximation using neural networks," *Neural Networks, IEEE Transactions on,* vol. 16, 2005 ,pp. 24-38.

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

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

[20] R. E. Smith, B. A. Dike, and S. Stegmann, "Fitness inheritance in genetic algorithms," in *Proceedings of the 1995 ACM symposium on Applied computing*, 1995, pp. 345-350.

[21] M. Salami and T. Hendtlass, "A fast evaluation strategy for evolutionary algorithms," *Applied Soft Computing,* vol. 2, 2003,pp. 156-173.

[22] E. Ducheyne, B. De Baets, and R. De Wulf, "Is fitness inheritance useful for real-world applications?," in *Evolutionary Multi-Criterion Optimization*, 2003, pp. 31-42.

[23] Z. Cui, J. Zeng, and G. Sun, "A fast particle swarm optimization," *International Journal of Innovative Computing, Information and Control,* vol. 2, 2006,pp. 1365-1380.

[24] Z. J. Sun C, Pan J, et al., " A new fitness estimation strategy for particle swarm optimization," *Information Sciences,*vol. 221, 2013,pp.355-370.

[25] C. Sun, J. Zeng, J. Pan, and Y. Jin, "Similarity-based evolution control for fitness estimation in particle swarm optimization," in *Computational Intelligence in Dynamic and Uncertain Environments (CIDUE), 2013 IEEE Symposium on*, 2013, pp. 1-8.

[26] H.-S. Kim and S.-B. Cho, "An efficient genetic algorithm with less fitness evaluation by clustering," in *Evolutionary Computation, 2001. Proceedings of the 2001 Congress on*, 2001, pp. 887-894.

[27] M. Reyes-Sierra and C. A. C. Coello, "A study of fitness inheritance and approximation techniques for multi-objective particle swarm optimization," in *Evolutionary Computation, 2005. The 2005 IEEE Congress on*, 2005, pp. 65-72.

[28] F. Gomide, "Fuzzy clustering in fitness estimation models for genetic algorithms and applications," in *Fuzzy Systems, 2006 IEEE International Conference on*, 2006, pp. 1388-1395.

[29] L. Fonseca, H. Barbosa, and A. Lemonge, "A similarity-based surrogate model for enhanced performance in genetic algorithms," *Opsearch,* vol. 46, 2009,pp. 89-107.

[30] L. G. Fonseca, A. C. Lemonge, and H. J. Barbosa, "A study on fitness inheritance for enhanced efficiency in real-coded genetic algorithms," in *Evolutionary Computation (CEC), 2012 IEEE Congress on*, 2012, pp. 1-8.

[31] Y. Jin and B. Sendhoff, "Reducing Fitness Evaluations Using Clustering Techniques and Neural Network Ensembles," in *Genetic and Evolutionary Computation – GECCO 2004: Genetic and Evolutionary Computation Conference, Seattle, WA, USA, June 26-30, 2004. Proceedings, Part I*, K. Deb, Ed., ed Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 688-699.

[32] Y. Sun, S. K. Halgamuge, M. Kirley, and M. A. Munoz, "On the selection of fitness landscape analysis metrics for continuous optimization problems," in *Information and Automation for Sustainability (ICIAfS), 2014 7th International Conference on*, 2014, pp. 1-6.

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

[34] M.-R. Akbarzadeh-T, M. Davarynejad, and N. Pariz, "Adaptive fuzzy fitness granulation for evolutionary optimization," *International Journal of Approximate Reasoning,* vol. 49, 2008, pp. 523-538.

[35] M. Davarynejad, C. Ahn, J. Vrancken, J. van den Berg, and C. C. Coello, "Evolutionary hidden information detection by granulation-based fitness approximation," *Applied Soft Computing,* vol. 10, 2010,pp. 719-729.

[36] M. Davarynejad, J. Vrancken, J. van den Berg, and C. A. C. Coello, "A fitness granulation approach for large-scale structural design optimization," in *Variants of Evolutionary Algorithms for Real-World Applications*, ed: Springer, 2012, pp. 245-280.

[37] I. Cruz-Vega, M. Garcia-Limon, and H. J. Escalante, "Adaptive-surrogate based on a neuro-fuzzy network and granular computing," in *Proceedings of the 2014 conference on Genetic and evolutionary computation*, 2014, pp. 761-768.

[38] J. Liang, B. Qu, P. Suganthan, and A. G. Hernández-Díaz, "Problem definitions and evaluation criteria for the CEC 2013 special session on real-parameter optimization," *Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou, China and Nanyang Technological University, Singapore, Technical Report,* vol. 201212, 2013.