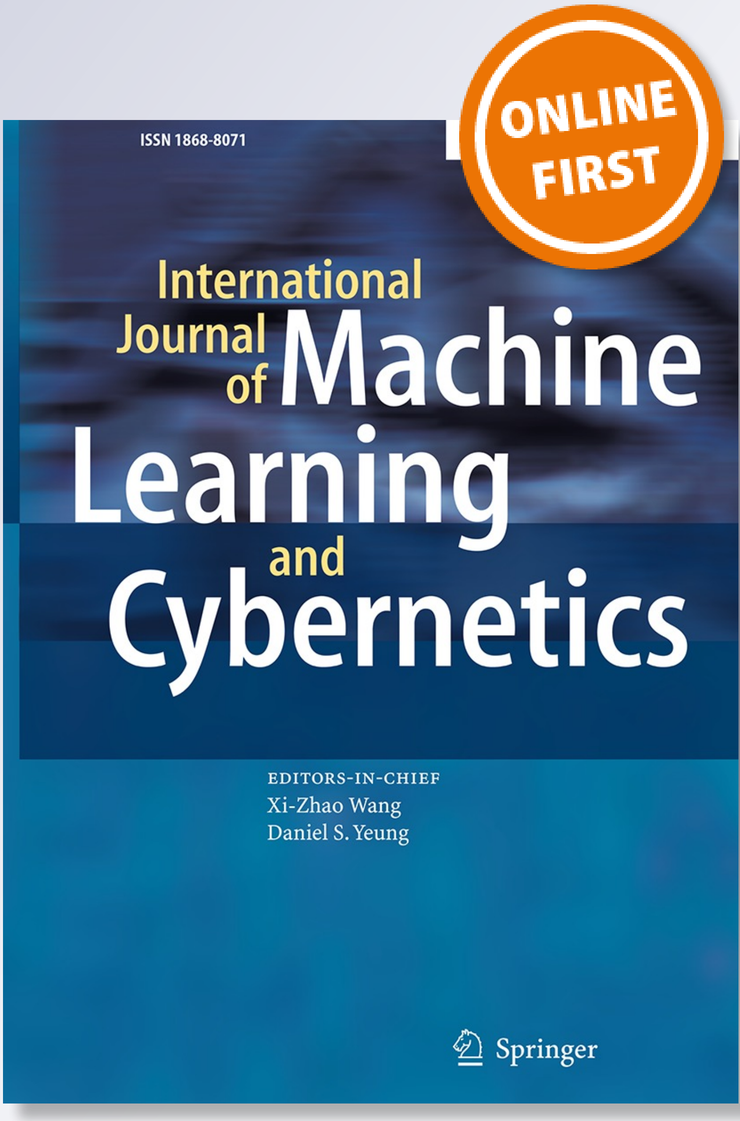
Effectiveness of approximation strategy in surrogate-assisted fireworks algorithm

Yan Pei, Shaoqiu Zheng, Ying Tan & Hideyuki Takagi

International Journal of Machine Learning and Cybernetics

ISSN 1868-8071

Int. J. Mach. Learn. & Cyber.
DOI 10.1007/s13042-015-0388-8



Springer

Int. J. Mach. Learn. & Cyber.
DOI 10.1007/s13042-015-0388-8

ORIGINAL ARTICLE

Effectiveness of approximation strategy in surrogate-assisted fireworks algorithm

Yan Pei[1] · Shaoqiu Zheng[2] · Ying Tan[2] · Hideyuki Takagi[3]

Received: 3 December 2014 / Accepted: 12 June 2015

Abstract We investigate the effectiveness of approximation strategy in a surrogate-assisted fireworks algorithm, which obtains the elite from approximate fitness landscape to enhance its optimization performance. We study the effectiveness of approximation strategy from the aspects of approximation method, sampling data selection method and sampling size. We discuss and analyse the optimization performance of each method. For the approximation method, we use least square approximation, spline interpolation, Newton interpolation, and support vector regression to approximate fitness landscape of fireworks algorithm in projected lower dimensional, original and higher dimensional search space. With regard to the sampling data selection method, we define three approaches, i.e., best sampling method, distance near the best fitness individual sampling method, and random sampling method to investigate each sampling method's performance. With regard to sample size, this is set as 3, 5, and 10 sampling data in both the approximation method and sampling method. We discuss and compare the optimization performance of each method using statistical tests. The advantages of the fireworks algorithm, a number of open topics, and new discoveries arising from evaluation results, such as multi-production mechanism of the fireworks algorithm, optimization performance of each method, elite rank, interpolation times and extrapolation times of elites are analysed and discussed.

Keywords Fireworks algorithm · Fitness landscape approximation · Elite strategy · Surrogate-assisted fireworks algorithm · Dimensionality reduction

✉ Yan Pei
peiyan@u-aizu.ac.jp;
http://web-ext.u-aizu.ac.jp/~peiyan/

Shaoqiu Zheng
zhengshaoqiu@pku.edu.cn

Ying Tan
ytan@pku.edu.cn

Hideyuki Takagi
takagi@design.kyushu-u.ac.jp;
http://www.design.kyushu-u.ac.jp/~takagi/

[1] Computer Science Division, The University of Aizu, 965-8580 Aizuwakamatsu, Japan

[2] Department of Machine Intelligence, School of Electronics Engineering and Computer Science, Key Laboratory of Machine Perception (Ministry of Education), Peking University, Beijing 100871, People's Republic of China

[3] Faculty of Design, Kyushu University, Fukuoka 815-8540, Japan

1 Introduction

Evolutionary computation (EC) has demonstrated the powerful capability to solve complex industrial and engineering optimization problems. From a framework viewpoint, there are three parts in an EC-based optimization system, which include a target system that should be optimized, an EC algorithm (including an user interface for interactive evolutionary computation (IEC) [21]) that conducts the concrete optimization operations, and one or multiple fitness function(s) (including a human user for IEC). These three parts encompass the corresponding three study aspects of an EC-based optimization system, i.e., EC applications, EC algorithms, and EC fitness function(s) or an IEC human user.

The EC algorithm performance is a serious issue for its real application. There are three promising study directions

Published online: 14 July 2015

aiming to meet this challenge in recent decades within EC community [9]. First is to approximate fitness landscape and make an operation or a model to assist the EC search. Takagi, et. al. proposed to use a single peak function to approximate fitness landscape and conduct a local search by using an elite obtained from approximated landscape [22]. Pei, et. al. extended this work by conducting the approximation in each lower dimensional search space for simplifying the approximation computation [12]. Other approximate methods by obtaining a frequency component information from fitness landscape was proposed in [11, 16]. Second is to develop effective search strategy or revise search mechanism into an EC algorithm to obtain an accelerated search convergence. Memetic algorithm was proposed to conduct individual learning or local improvement procedures for optimization [7]. It also can be introduced into the IEC for obtain a better optimization performance [15]. A series of search strategies were developed in differential evolution (DE) community, which include SaDE [18], jDE [1], SaJADE [2], JADE [25], etc. A triple and quadruple comparison-based DE was proposed in [14], in which an opposition-based learning process is embedded into the DE selection process. These novel strategies and mechanisms can be categorized into two aspects, the one is searching in a form of hybrid global evolutionary algorithm coupled with a learning procedure capable of local refinements, the other is conducting the different search strategies according to the knowledge obtained from the search process. Third is to create a new type of EC algorithm that is inspired by biological, physical or mathematical schemes and phenomenon, such as the fireworks algorithm [23], chaotic evolution [8, 10], water wave optimization [27], water cycle algorithm [19], optics inspired optimization [6], etc. This is the primary subject of investigation in this paper.

Fireworks algorithm (FWA) is a recently developed algorithm [23]. It is inspired by the explosion of fireworks in the night sky, which can illuminate the dark space. The process of illuminating the nearby space of a firework position can be considered as the search among the feasible range. The FWA work mechanism is described in the following. A number of fireworks are set off to the feasible range, and each position of the firework will be evaluated by the fitness function. The explosion amplitude and spark number for each firework are calculated based on the fitness of the fireworks. For a minimal objective problem, the principle is that the firework with smaller fitness will have a smaller explosion amplitude and a larger number of sparks. The fireworks are exploded under the calculated explosion amplitude and spark number. Moreover, the Gaussian explosion sparks are also generated to enhance the diversity of the FWA. A number of fireworks are selected from the candidate sets, which include the fireworks, regular explosion sparks and Gaussian explosion sparks. The FWA will continue until the termination condition is met.

We proposed to use a surrogate model for improving the performance of FWA. Surrogate-assisted FWA uses efficient computational models for approximating the fitness landscape where we obtain the elite from search space to accelerate FWA search. The method is to approximate the fitness landscape in the one-dimensional search space, and obtain and synthesize the elite from the approximated landscape curves. When the elite is better than the worst firework, surrogate-assisted FWA replaces the worst one with this elite into the next generation to enhance the FWA search. An empirical study on influence of approximation methods on the FWA acceleration performance was initially investigated and analysed [17]. However, some of the problems in applying this method need further study and investigation.

This paper extends the work of [17] and conducts a further investigation on the approximation method and strategy influence on surrogate-assisted FWA. We especially investigate: (1) several approximation methods for approximating fitness landscape and their acceleration performance, (2) sampling methods and sizes' influence on acceleration performance, (3) the obtained elite rank in the population and its effectiveness, (4) the obtained elite from within or beyond the approximation interval, (5) statistical test on the proposed methods' significance. Several related issues are also investigated and discussed. The evaluation metrics include the convergent fitness value after the same number of generations, Wilcoxon signed-rank test, elite rank, and the number of interpolation and extrapolation, etc.

Following this introductory section, in Sect. 2, an overview on the FWA is presented. The history, inspiration and development of the FWA are described in detail. In Sect. 3, we introduce several techniques on fitness landscape approximation and theoretically discuss the computational complexity. The surrogate-assisted FWA enabled by a technique for reducing dimensionality of the search space is explained, and we show how elite can be obtained from the regression search space. The fitness landscape can be approximated in lower dimensional space, original space and higher dimensional space. In Sect. 4, experimental evaluations with 25 benchmark functions with 10 dimension (10-D) and 30-D are conducted, and their results are analysed and discussed. Finally, we discuss our proposed methods and obtained results in Sect. 5, and conclude the analysis and investigation results of the proposed methods, present future opportunities and open topics in Sect. 6.

2 An overview of the fireworks algorithm

As one of the swarm intelligence algorithms, the FWA is inspired by the phenomenon of the explosion of fireworks in the night sky. When a firework explodes in the night sky, it illuminates the dark space, which can be considered as a kind of search process. In the FWA, the fireworks generate sparks candidates around themselves in search space. The search operators, which maintain the exploration ability for the heuristic algorithms, can be implemented by simulating some fireworks' explosions in the potential space.

There are three promising research aspects in recent studies of the FWA. The first one is to construct a hybrid optimization framework by using the FWA with other EC algorithms or computational mechanisms. Zheng et. al. proposed a hybrid FWA with biogeography-based optimization to enhance the optimization performance of canonical FWA [24]. Ding et. al. implemented a parallel FWA by using a parallel processing architecture, GPU. The second aspect is to establish a FWA optimization framework to solve multi-objective optimization problems [28], constrained optimization problems, combinatorial optimization problems, etc. The third aspect is to apply the FWA to the real-world applications. Janecek et. al. used the FWA with other EC algorithms, such as particle swarm optimization, genetic algorithms, differential evolution, and fish school search, for improving the initialization of the non-negative matrix factorization problem [4].

In general, given the following single objective function $\{min f : \Omega \subseteq R^n \rightarrow R\}$, the FWA is to find a point $x \in \Omega$, for which x has the minimal value. Algorithm 1 presents the framework of FWA. In each generation of FWA, N fireworks are randomly initialized in the feasible search range, and the fitness of each firework will be evaluated to determine the explosion amplitude and explosion sparks number. The basic principle for the explosion amplitude and sparks number is that the firework with better fitness will have smaller explosion amplitude and larger population of sparks to increase the exploited ability, while the firework with worse fitness will have a small population of explosion sparks and large explosion amplitude to maintain search ability. After the calculation of the explosion sparks number and explosion amplitude for each firework, the FWA performs the explosion process to generate the regular explosion sparks by Algorithm 2. To increase the diversity of the FWA, it introduces another sparks named Gaussian sparks generated by Gaussian mutation operation (Algorithm 3). The selection operator is performed to cause the N fireworks from the candidate set which includes regular sparks, $\hat{m}$ Gaussian explosion fireworks and N fireworks to remain. The algorithm continues until the termination criterion is met, i.e., maximum generation or evaluation, maximum running time, or the optimum is found.

Algorithm 1 The FWA Framework. N is the number of fireworks, G means generation, and $maxIter$ means the maximum generation number.

```
Initialization.
Randomly generate N fireworks at N locations
for G = 1 to maxIter do
    Generate the regular explosion sparks by Algorithm 2
    Generate the Gaussian explosion sparks ( i.e., Gaussian
    explosion fireworks) by Algorithm 3
    Obtain N fireworks for the next iteration (the best one
    firework is kept, and the rest N−1 fireworks is selected
    by the roulette wheel method)
end for
Return the optima
```

2.1 Regular explosion sparks

Regular explosion operator is one of the crucial operators in the FWA. For the simulation of the fireworks explosion process, the FWA generates a number of sparks s_i within the explosion amplitude A_i by Eqs. (1) and (2), where $y_{max} = max(f(x_i))$ and $y_{min} = min(f(x_i))$, $i = 1, 2, \ldots, N$. Here $\hat{A}$ and M are constants which are determined by a practical optimization problem. After the calculation of the number of sparks and explosion amplitude, the fireworks perform the generation of the regular explosion sparks by Algorithm 2.

Algorithm 2 – Generating "regular explosion sparks" in the FWA [26].

```
Initialize the location of the "regular explosion sparks":
X̂_i = X_i
Calculate offset displacement: △X = A_i × rand(−1, 1)
Set z^k = round(rand(0, 1)), k = 1, 2, ..., d
for each dimension of X̂_i^k, where z^k == 1 do
    X̂_i^k = X̂_i^k + △X
    if X̂_i^k out of bounds then
        X̂_i^k = X_min^k + |X̂_i^k| % (X_max^k − X_min^k)
    end if
end for
```

$$s_i = M \cdot \frac{y_{max} - f(x_i) + \varepsilon}{\sum_{i=1}^{n} (y_{max} - f(x_i)) + \varepsilon} \tag{1}$$

$$A_i = \hat{A} \cdot \frac{f(x_i) - y_{min} + \varepsilon}{\sum_{i=1}^{n} (f(x_i) - y_{min}) + \varepsilon} \tag{2}$$

2.2 Gaussian explosion sparks

To increase the diversity of the explosion sparks, another kind of sparks is introduced, named Gaussian explosion sparks. The generation principle of the Gaussian explosion sparks is to create the fireworks by a random value from a Gaussian distribution. The detail of this process is implemented by Algorithm 3.

2.3 Selection for the next generation

After the explosion of fireworks, a number of sparks (regular explosion sparks and Gaussian explosion sparks) are generated. In the selection of fireworks for the next iteration, the candidate (fireworks or sparks) with the best fitness is always kept, the others are selected by the roulette wheel method by using the following Eqs. (3) and (4), where the location X_i, the selection probability $p(X_i)$ is calculated. Here, K denotes the set of all current locations which includes the fireworks, regular explosion sparks and Gaussian explosion sparks (without the best candidate).

Algorithm 3 – Generating "Gaussian explosion sparks" in the FWA [26].

```
Initialize the location of the "Gaussian explosion sparks": X̃_i = X_i
Calculate offset displacement: e = Gaussian(1, 1)
Set z^k = round(rand(0, 1)), k = 1, 2, ..., d
for each dimension of X̂_i^k, where z^k == 1 do
    X̃_i^k = X̃_i^k × e
    if X̃_i^k out of bounds then
        X̃_i^k = X_min^k + |X̃_i^k| % (X_max^k − X_min^k)
    end if
end for
```

$$p(X_i) = \frac{R(X_i)}{\sum_{X_j \in K} R(X_j)} \tag{3}$$

$$R(X_i) = \sum_{X_j \in K} d(X_i, X_j) = \sum_{X_j \in K} \left|\left|X_i - X_j\right|\right| \tag{4}$$

3 Surrogate-assisted fireworks algorithm framework

3.1 Motivation

Fitness and Fitness landscape approximations are well-known assisted acceleration techniques in the EC community over the last decade [5]. In some applications, the computation of fitness is time-consuming, so fitness approximation can dramatically save the computation time so as to improve EC performance. On the other hand, fitness landscape approximation can obtain the whole search space structure information to assist EC search. Unlike the conventional EC algorithms, they use the least search information that is supported by the limited individuals' fitness. The computation of either fitness or fitness landscape approximations increases an additional time cost in the EC optimization process. Some of the conventional approximation methods are time-consuming, so reducing the approximation time and developing an efficient approximation method is a promising subject to improve optimization performance of the surrogate-assisted EC.

Approximation method selection and data sampling technique are important issues in the computation of fitness landscape approximation. We have obtained some empirical results on the surrogate-assisted FWA in [17] that shows: (1) The elite strategy is an efficient method to enhance the FWA search capability significantly. (2) The sampling method cannot take effect in isolation, it must be with a proper approximation model to accelerate the FWA search for a certain benchmark function, i.e., a certain fitness landscape. (3) For some benchmark problems, the best sampling method and the random sampling method have the same acceleration performance. (4) The surrogate-assisted FWA can be obtained by a fitness landscape approximation with more sampling size and a proper approximation method. In our study, the better approximation method is the non-linear model. (5) From a practical point of view, the random sampling method is a better sampling way to obtain the higher acceleration performance in both computational time and final solution quality. However, there are many remaining issues that we need to further investigate, such as the approximate method, the sampling method and the sampling size. These investigations and analyses are one of original features of this paper.

3.2 Approximation methods

In this work, we use the following approximation methods to investigate our proposed acceleration performance influenced by the approximation method. They are least square approximation, spline interpolation, Newton interpolation, and support vector regression.

3.2.1 Least square approximation

The approximation method does not require approximated curves to pass through all discrete points exactly, but rather to approach the original curve at its discrete points, i.e. (x_i, y_i), as near as possible. When we define the error vector norm as 2-norm, the approximation method is referred to as the least squares method.

There is a function spanning space shown in Eq. (5), Φ denotes the function class, *Span* means spanning space, and $\varphi_i(x), (i = 0, 1, \ldots, n)$ denotes a function in the space. We want to find a function that makes the 2-norm error vector $(||\delta^*||_2^2)$ be minimized as shown in Eq. (6). The approximation function is shown in Eq. (7), where $a_i^*, (i = 0, 1, ..., n)$ are the parameters that make the Eq. (6) to be minimized. If we set $\varphi_0(x) = 1, \varphi_1(x) = x$, and $\varphi_0(x) = 1, \varphi_1(x) = x, \varphi_2(x) = x^2$ as the approximation function in Eq. (6), they are called linear least squares approximation method and 2-degree polynomial least squares approximation method, respectively.

$$\Phi = Span\{\varphi_0(x), \varphi_1(x), \ldots, \varphi_n(x)\} \tag{5}$$

$$||\delta^*||_2^2 = \sum_{i=0}^{m} \delta_i^{*2} = \sum_{i=0}^{m} [\varphi^*(x_i) - y_i]^2 = \min_{\varphi(x)\in\Phi} ||\delta^*||_2^2 \tag{6}$$

$$\varphi^*(x) = a_0^*\varphi_0(x) + a_1^*\varphi_1(x) + \cdots + a_n^*\varphi_n(x) \tag{7}$$

3.2.2 Spline interpolation

Spline interpolation is defined by a polynomial function. Compared with polynomial interpolation, spline interpolation has a better approximation performance. We can obtain the interpolation results by spline interpolation with lower degree function, while avoiding the instability of interpolation due to Runge's phenomenon. This shows the advantage of spline interpolation [3].

Suppose that there are different points $(x_i, y_i), (i = 0, 1, \ldots, n)$, the objective of spline interpolation finds an n-degree spline function $S(x)$ defined in Eq. (8), where $S_i(x)$ is a k-degree polynomial function. The linear spline function is given as in Eq. (9).

$$S(x) = \begin{cases} S_0(x) & x \in [x_0, x_1] \\ S_1(x) & x \in [x_1, x_2] \\ \vdots & \vdots \\ S_{n-1}(x) & x \in [x_{n-1}, x_n] \end{cases} \tag{8}$$

$$S_i(x) = y_i + \frac{y_{i+1} - y_i}{x_{i+1} - x_i}(x - x_i) \tag{9}$$

3.2.3 Lagrangian interpolation and newton interpolation

The Lagrange interpolation polynomial has the characteristics of linear and unique. For one-dimensional data from individuals x_0, x_1, ..., x_n, we can set up an n-degree polynomial $l_0(x)$, $l_1(x)$, ..., $l_n(x)$. We set its type as $l_i(x_j) = \delta_{ij}$, where the form of δ_{ij} is as shown in Eq. (10).

$$\delta_{ij} = \begin{cases} 0 & \text{if} \quad \text{i} = \text{j} \\ 1 & \text{if} \quad \text{i} \neq \text{j} \end{cases} \tag{10}$$

From the definition of $l(x)$, we can obtain Eq. (11), which is the n-degree interpolation polynomial, where $l_k(x)$ is an n-degree polynomial. The relationships of the Lagrange interpolation polynomial are shown in Eqs. (11), (12), (13) and (14).

$$p_n(x) = \sum_{k=0}^{n} l_k(x)y_k = y_i \tag{11}$$

$$l_k(x) = a(x - x_0)\ldots(x - x_{k-1})(x - x_{k+1})\ldots(x - x_n) \tag{12}$$

When the condition is $l_k(x_k) = 1$ in Eq. (12), we can obtain Eq. (13). Then, Eq. (12) can be re-written as in Eq. (14). Equations (14) and (15) are the n-degree Lagrange interpolation basis function and the n-degree Lagrange interpolation polynomial, respectively.

$$a = [(x_k - x_0)\ldots(x_k - x_{k-1})(x_k - x_{k+1})\ldots(x_k - x_n)]^{-1} \tag{13}$$

$$l_k(x) = \prod_{i=1, i\neq k}^{n} \frac{(x - x_i)}{(x_k - x_i)} \tag{14}$$

$$L_n(x) = \sum_{k=0}^{n} l_k(x)y_k, i = 0, 1, \ldots, n \tag{15}$$

Because the polynomial interpolation is unique, the Newton interpolation has the same form as the Lagrange interpolation shown in Eq. (15).

3.2.4 Support vector regression

Support vector regression (SVR) is a kernel method that is used in regression. If a problem is non-linear, instead of trying to fit a non-linear model, the SVR projects the problem from the input space to a higher dimensional space, i.e., a feature space, to find a linear model by conducting a non-linear transformation using suitably chosen kernel functions. It uses the linear model in the feature space to solve the problem. There are two aspects in the primary motivation of SVR. One is to project the original space into a high dimensional space to find the linearity in such space. The other is to conduct the regression process with those linearity characteristics.

There are several training sample data in the original space, x_1, x_2,..., x_N, and there is a feature map function $\phi(x_i)$, $i = 1, 2, \ldots, N$, which conducts the feature map to project the data into a higher dimensional space (Eqs. (16), (17) and (18), X denotes a matrix that encompasses all the data that are projected into a feature space, y denotes the matrix that the value of each data $x_i, i = 1, \ldots, n$.). From the normal equation of linear regression, we can obtain the support vector regression expression (Eq. (19)). $K(x, y)$ is a kernel function, which is defined as $K(x,y) = \exp -||x-y||$, i.e., a Gaussian kernel function, in our evaluation.

$$X = \begin{bmatrix} \phi(x_1)^T \\ \vdots \\ \phi(x_N)^T \end{bmatrix} \quad (16)$$

$$\overline{X} = \begin{bmatrix} \overline{\phi(x_1)}^T \\ \vdots \\ \overline{\phi(x_N)}^T \end{bmatrix} = \begin{bmatrix} 1, & \phi(x_1)^T \\ & \vdots \\ 1, & \phi(x_N)^T \end{bmatrix} = [1_{N\times 1}, X] \quad (17)$$

$$y = \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix} \quad (18)$$

$$f(x) = (\alpha_1 + \cdots + \alpha_N) + \alpha_1 K(x_1, x) + \cdots + \alpha_N K(x_N, x) \quad (19)$$

$$\alpha = \overline{X}(\overline{X}^T\overline{X})^{-2}\overline{X}^T y \quad (20)$$

3.3 Sampling selection methods

When a regression model is established by a given approximation method, the sample data should be selected with certain criteria to train the model. This is also a subject of study in this paper. We use three sampling data selection methods in our investigation to understand differences of the FWA acceleration performance by these three sample data selection methods. They are listed as follows.

- The best sampling method selects the best K individuals as sampling data.
- The distance near the best fitness individual sampling method selects the nearest K individuals to the best individual using Euclidean distance as sampling data.
- The random sampling method selects K individuals randomly as sampling data.

3.4 Sampling sizes

The sampling size is another factor that influences the approximate model's accuracy and acceleration performance of the FWA. Theoretically, where a large sampling size is applied in the approximation or interpolation process, the better approximation result will be obtained, if over-fitting does not occur. In this study, the influence of sampling size on acceleration performance of the FWA is also a subject of investigation. Specifically, we set the sample size as 3, 5, 10 for the methods that conduct the approximation and interpolation in lower and higher dimensional space, and set the sampling size as $2D+1$ (D is the dimension of benchmark function) for the approximation and interpolation in original space, because the regression model in original space has $2D+1$ unknown parameters.

3.5 Surrogate-assisted fireworks algorithm

The surrogate-assisted FWA uses an approximation or interpolation model of fitness landscape as the surrogate model to assist FWA search. The approximation or interpolation can be conducted in lower dimensional space, original space and high dimensional space. After we obtain the approximated fitness landscape, we can apply the elite strategy from the approximated fitness landscape to enhance optimization performance of the FWA.

The elite strategy in lower dimensional space for approximating fitness landscape uses only one of the n parameter axes at a time instead of all n parameter axes, and projects individuals onto each 1-dimensional (1-D) space. Each of the n 1-D spaces has K projected individuals, which come from the different sampling methods with a different sampling size. We approximate the landscape of each 1-D space using the projected K individuals and select the elite from the n approximated 1-D landscape shapes. The elite are generated from the resulting 1-D approximated shapes, see Fig. 1.

The actual least square regression functions used is polynomial curve fitting, given by Eq. (21), where x_{ij}, $(i = 1,2,\ldots,D)$ and $(j = 1,2,\ldots,K)$ are the projected individual of point set X_i, $(i = 1,2,\ldots,D)$ and their fitness values among (x_{ij}, y_j) in the i-th 1-D regression space for $(i = 1,2,\ldots,D)$ and $(j = 1,2,\ldots,K)$, a_0, a_1, ..., a_k are the parameters obtained by least squares method, t is the power of polynomial function.

$$\begin{pmatrix} x_{11}^t & x_{12}^{t-1} & \ldots & x_{1K}^0 \\ x_{21}^t & x_{21}^{t-1} & \ldots & x_{2K}^0 \\ \vdots & \vdots & & \vdots \\ x_{D1}^t & x_{D2}^{t-1} & \ldots & x_{DK}^0 \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_K \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_K \end{pmatrix} \quad (21)$$

Least square approximation by a two-degree polynomial function (t = 2) simplifies a fitness landscape with a non-linear curve, and it is easy to obtain its inflection point from its gradient, using the inflection point as the elite. Linear least square approximation uses a linear function (t = 1) to approximate the fitness landscape. Its gradient is either descent or ascent. A safer approach, taking into account both descent and ascent, is to select the average point of the linear approximation line as the elite. The other approximation and interpolation methods obtain the elite from

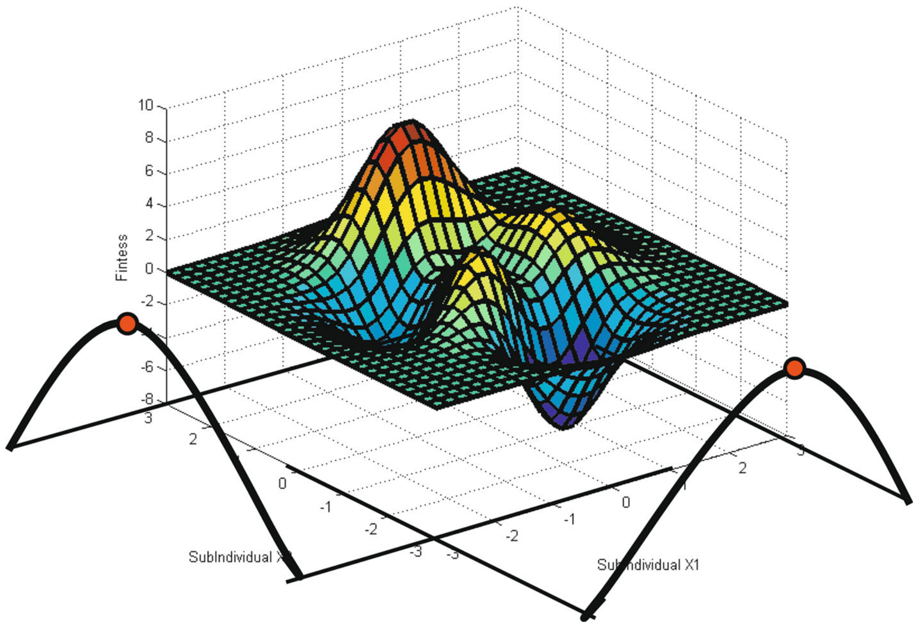


Fig. 1 Original n-D space and 1-D spaces obtained by reducing the dimensions of the original one. In our proposed approximation of fitness landscape in lower dimensional space, first, we project fitness landscape in each lower dimension (in this paper, we project it into one dimensional search space, but it is not limited to one dimensional search space), second, we conduct approximation in each lower dimensional search space and obtain elite from each simple shape or surface, finally, we combine these lower dimensional elite together in each related dimensional position as the final elite to enhance FWA optimization performance

their shapes by re-sampling 100 times and selecting the re-sample data with the best fitness as the elite.

The proposed methods replace the worst individual in each generation with the selected elite. Although we cannot deny the small possibility that the global optimum is located near the worst individual, the possibility that the worst individual will become a parent in the next generation is also low. Removing the worst individual therefore presents the least risk and is a reasonable choice.

The whole workflow of the surrogate-assisted FWA with an elite strategy is shown in Algorithm 4, where algorithm initialization is in step 2, the optimization of one generation is presented in steps 3 to 17, and in step 18, the algorithm returns the best obtained solution. It is the crucial process in step 11 to step 16 that explains the elite strategy. This includes (a) obtain sampling data with a certain sampling select method and sampling size from N fireworks (step 11), (b) approximate fitness landscape (step 12), (c) obtain a spark from approximated curves by elite strategy (step 13), and (d) if the fitness of the elite is better than that of the worst firework, the worst one will be replaced (steps 14–16).

Algorithm 4 Pseudo-code of the surrogate-assisted FWA with an elite strategy.

```
1:  Initialization.
2:  Randomly select N fireworks at N locations
3:  for G = 1 to maxIter do
4:      Evaluate the N fireworks using objective function
5:      Calculate the number of sparks and explosion amplitude of each fireworks
6:      Generate regular explosion sparks by Algorithm 2
7:      Generate Gaussian explosion sparks by Algorithm 3
8:      Obtain the optimal candidate in the Set which includes generating sparks, m̂ Gaussian explosion fireworks and N fireworks
9:      Randomly select the other N − 1 fireworks
10:     Obtain N fireworks for the next iteration
11:     Obtain sampling data with a certain sampling select method and sampling size from N fireworks
12:     Approximate fitness landscape
13:     Obtain a spark from approximated curves by elite strategy
14:     if EliteFitness > WorstIndividualFitness then
15:         Replace the worst individual
16:     end if
17: end for
18: Return the optima
```

4 Experimental evaluation

4.1 Experimental setting

In our evaluations, the sampling size is set as 3, 5 and 10, i.e., $K = 3$, $K = 5$ and $K = 10$ to investigate the sampling size's influence on the acceleration performance of surrogate-assisted FWA. This sampling size setting is applied in the approximation and interpolation methods in lower and higher dimensional space. It needs at least $2D + 1$ sampling number for least squares approximation with one and two degree polynomial functions, because there are $2D + 1$ unknown parameters. We investigate least squares approximation with one and two degree polynomial functions, spline interpolation, Newton interpolation for approximating the fitness landscape in projected 1-D space, and least squares approximation with two degree polynomial functions for approximating fitness landscape in original space, and the SVR with Gaussian kernel function for approximating fitness landscape in high dimensional space. Three sampling selection methods are applied in each approximation and interpolation methods with different sampling sizes.

The number of fireworks (N) and Gaussian explosion firework ($\hat{m}$) are both set as 8 and other parameters are set as in [23]. Experimental evaluations run 30 trials of 1000 generations on each benchmark function independently. The experimental platform is MATLAB 2011b, running under Windows 7 on an Intel Core i7-2600 CPU with 3.7 GHz and 8GB RAM. To validate the performance of the proposed algorithm and investigate the influences of different approximation strategies, two groups of experiments are designed: performance comparison on benchmark functions with dimension set to 10 and 30 (10-D and 30-D). The abbreviations of the proposed algorithm, sampling method and sampling size are shown in Table 1.

Table 1 Abbreviation of proposed algorithm, sampling method and sampling size

Abbreviation	Meaning
LS1	Least square approximation with linear function [12]
LS2	Least square approximation with second degree polynomial function [12]
Spline	Spline interpolation
Newton	Newton interpolation
OLS	Least square approximation with a single peak function in original space [22]
SVR	Support vector regression [13]
BST	Best sampling method
DIS	Distance sampling method
RND	Random sampling method
3	Sampling size is 3
5	Sampling size is 5
10	Sampling size is 10

In this paper, we use six approximation methods, three sampling data selection methods and three sampling sizes in the evaluation experiments. There are total of 48 algorithms plus one canonical FWA

4.2 Benchmark functions

We investigate our proposed algorithms and related issues by using benchmark functions from CEC2005 benchmark test suite [20]. In the benchmark functions, 25 functions are included, which presents a variety of fitness landscapes, such as uni-modal and multi-modal, shifted, rotated, global optimum on bounds, etc. Table 2 presents a detailed description of these benchmark functions.

4.3 Evaluation metrics

For comparing and analysing the performance of different algorithms, the fitness values, elite rank, elite location, approximation interval are recorded. The Wilcoxon signed-rank test is applied to validate the significant difference between two algorithms. In our evaluation, the significance level is set to 5 %, i.e., $p < 0.05$ (p means p-value in Wilcoxon signed-rank test.).

Tables 3 and 4 present the 10-D benchmark function's mean values of each algorithm, and the Wilcoxon signed-rank test results from comparing the canonical FWA and our proposed surrogate-assisted FWA with different approximation methods, different sampling methods and sampling sizes. The † mark in these tables means the proposed method significantly outperforms the canonical FWA by Wilcoxon signed-rank in the significant level $p < 0.05$. Table 5 shows the average elite rank from each proposed algorithm for 10-D and 30-D benchmark functions. Figure 2 describes the average times of elite obtained outside of approximation interval, i.e., extrapolation phenomenon occurs. We discuss and analyse optimization performance of each method based on these results.

5 Discussion

5.1 Analysis of approximation approach influence

In the approximation method aspect, the approximation method in low dimensional space and original space seems to obtain the same optimization results. However, the approximation method in the high dimensional space, i.e., support vector regression method, is effective in a few of

Table 2 Benchmark functions in our evaluation experiments from [20]

No.	Type	Description	Bounds	Optima
f_1	Uni	Sh Sphere		−450
f_2		Sh Schwefel 1.2		−450
f_3		Sh Rt Elliptic	$[-100, 100]$	−450
f_4		f_2 with Noise		−450
f_5		Schwefel 2.6 GB		−310
f_6	Multi	Sh Rosenbrock	$[-100, 100]$	390
f_7		Sh Rt Griewank	[0, 600]	−180
f_8		Sh Rt Ackley GB	$[-32, 32]$	−140
f_9		Sh Rastrigin	$[-5, 5]$	−330
f_{10}		Sh Rt Rastrigin	$[-5, 5]$	−330
f_{11}		Sh Rt Weierstrass	$[-0.5, 0.5]$	90
f_{12}		Schwefel 2.13	$[\pi, \pi]$	460
f_{13}		Sh Expanded F8F2	$[-3, 1]$	−130
f_{14}		Sh Rt Scaffer F6	$[-100, 100]$	−300
f_{15}	Hybrid	HC Function		120
f_{16}		Rt HC Function 1		120
f_{17}		f_{16} with Noise		120
f_{18}		Rt HC Function 2		10
f_{19}		$f_{18}$with Basin		10
f_{20}		f_{18} with GB	$[-5, 5]$	10
f_{21}		Rt HC Funtion 3		360
f_{22}		f_{21} with NM		360
f_{23}		NC Rt f_{21}		360
f_{24}		Rt HC Function 4		260
f_{25}		f_{24} without Bounds		260

The abbreviations in this table present as follows. *Uni* Uni-modal, *MultI* Multi-modal, *Sh* Shifted, *Rt* Rotated, *GB* Global on Bounds, *HC* Hybrid Composition, *NM* Number Matrix

the benchmark problems. In the projected high dimensional feature space, linear regression can be perfectly constructed by the SVR method, however, this model cannot present the actual fitness landscape in the original space. The over-fitting issue is the reason that leads to the worse fitness approximation and optimization performance, so, from a practical viewpoint, the SVR is not a good approximation method that can be applied in a fitness landscape approximation field. The same results are also obtained in the work of [13].

Compared with approximation methods in lower dimensional space and that in original space, optimization results by lower dimensional approximation are better than that by original dimensional approximation from the observations of Tables 3 and 4. This indicates that rough and simple approximation of fitness landscape can be useful in EC enhancement application, and the exact approximation performance sometimes leads to local optimal. Approximation in original space needs more time because of many matrix operations, its computational complexity is higher than that of approximation in lower dimensional space. Along with the dimension increasing, the computational complexity of approximation in original space will dramatically rise up. Comparing these two methods applied in 10-D and 30-D benchmark problems, optimization performance obtained by approximation in low dimensional space outperforms that obtained by approximation in original space. As dimensionality is more higher, the optimization result is more better. So, from whatever theoretical analysis and practical evaluation of these two approximation method, approximation in low dimensional space by dimension reduction technique has its advantages and practical benefits.

Our proposed methods fail to optimize the complex fitness landscape benchmark functions (e.g., f_{18}, f_{19}, and f_{20}) and specific fitness landscape structure (e.g., f_8: optimum at bounds, f_{12}: shifted global optimum, etc.). This result indicates a limitation of our proposed methods, which is effective for the problems with simple fitness landscape. We should analyze the characteristics of the optimization problem and select a proper approximation method to obtain the better optimization performance before we apply our proposed surrogate-assisted FWA.

5.2 Analysis of sampling method influence

In this work, we propose three sampling data selection methods, i.e., the BST method, the DIS method, and the RND method, which mean selecting sampling data from the best n data, selecting sampling data from n nearest data from the best data, and selecting sampling data randomly, respectively. From the optimization results by comparing these three methods, when the BST and RND methods can obtain significant optimization performance, the DIS method cannot obtain the same result. From this observation, the DIS method seems less useful than other two methods. Sampling data distribution is a significant factor that influences approximation effect. The DIS method selects data that are nearest to the current best point, it leads to local approximation rather than global approximation that can be obtained by the other two methods. This maybe a reason that why the DIS method fails in most of the benchmark problems.

Both the DIS method and BST method need to use search and sorting algorithms for selection. However, the RND method processes the sampling data randomly with a certain distribution probability. From the practical viewpoint, the computational complexity of the RND method is less than that of other two methods, and the RND method can obtain significantly better optimization performance from the above observation and analysis, we conclude that

Table 3 Mean fitness value of F1-F14 with 10 dimension. The abbreviations used here is in Table 1

Method	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}
Normal	−442.817	194.3666	3529551	2105.924	2656.02	2747.469	−177.339	−119.899	−313.709	−281.939	96.29126	3093.602	−128.999	−296.465
LS1-BST3	−449.86†	−101.5†	1419161.51†	132.3†	1911.9	1290.73†	−178.12	−119.92	−323.85†	−287.4	96.8	1900.35	−129.26†	−296.39
LS1−BST5	−449.78†	−165.23†	1402447.33†	117.92†	1610.44	1367.29†	−178.24	−119.91	−324.89†	−285.74	96.89	2028.88	−129.32†	−296.44
LS1−BST10	−449.76†	−158.04†	788825.5†	43.87†	1333.34†	1651.82†	−178.21†	−119.89	−323.52†	−281.69	97.3†	1748.98†	−129.33†	−296.46
LS1−DIS3	−436.11	59.34	3435249	524.68†	2480.33	2608.99	−178.17	−119.92	−316.02	−289.1	96.79	3092	−129.08	−296.43
LS1−DIS5	−436.49	−9.53†	1970932	291.02†	2468.56	3171.58	−178.18	−119.9	−317.74	−286.67	96.33	2451.55	−129.22†	−296.41
LS1−DIS10	−442.83	−31.6	1764738.82†	111.79†	2569.26	2992.26	−178.05	−119.9	−317.82	−287.97	96.07	2602.44	−129.12	−296.32
LS1−RND3	−448.28†	−271.92†	858698.8†	−156.2†	1731.47	2417.13	−178.05†	−119.91	−320.82†	−288.47	96.35	2339.96	−129.23†	−296.39
LS1−RND5	−447.61†	−279.7†	975242.14†	−136.35†	2737.96	3101.7	−178.19†	−119.9	−316.39	−289.29	97.13†	2738.46	−129.23†	−296.42
LS1−RND10	−447.13†	−224.5†	886462.29†	−79.06†	2097.08	2393.86	−178.27†	−119.9	−318.63	−289.4	96.49	2139.09	−129.05	−296.45
LS2−BST3	−449.89†	−257.04†	1192195.74†	−15.19†	965.55†	1188.36†	−178.57†	−119.9	−324.15†	−286.98	97.35†	2103.86	−129.17†	−296.39
LS2−BST5	−449.89†	−215.57†	1414917.38†	−83.43†	1402.14†	1858.9	−178.4†	−119.91	−323.94†	−281.64	97.24†	2949.65	−129.25†	−296.44
LS2−BST10	−449.92†	−277.6†	1351030.14†	−212.34†	1302.74†	1314.58†	−178.48†	−119.91	−325.06†	−284.26	97.66†	1913.94	−129.2	−296.39
LS2−DIS3	−446.63†	32.59	1320490.13†	289.61†	1665.86	2504.29	−178.2	−119.88	−315.95	−288.34	95.41†	3715.42	−129.19†	−296.48
LS2−DIS5	−443.47	−21.89†	1277189.38†	121.5†	1640.35	3104.63	−178.43†	−119.9	−316.51	−294.73†	95.29†	2721.99	−129.27†	−296.39
LS2−DIS10	−444.23	−3.62†	956602.9†	93.92†	1025.44†	1931.72	−178.48†	−119.9	−317.39	−291.66	94.79†	1835.85†	−129.21	−296.44
LS2−RND3	−449.94†	−400.39†	849730.96†	−321.26†	1002.5†	1681.65†	−178.83†	−119.87	−323.14†	−286.08	96.8	1596.95†	−129.3†	−296.47
LS2−RND5	−449.91†	−408.07†	670058.42†	−346.44†	662.76†	2387.03	−178.87†	−119.91	−322.68†	−291.16†	96.39	1610.89†	−129.24†	−296.5
LS2−RND10	−449.29†	−395.9†	791469.78†	−348.92†	−148.03†	2784.89	−178.77†	−119.91	−319.99†	−293.86†	96.55	2709.44	−129.23†	−296.46
Spline−BST3	−449.92†	−247.85†	1456508.65†	17.25†	1357.85†	1146.91†	−178.3	−119.93	−324.79†	−283.57	96.87	1329.48†	−129.19	−296.48
Spline−BST5	−449.91†	−286.57†	989402.74†	−83.7†	1468.55†	1305.05†	−178.31†	−119.89	−323.28†	−288.39	97.14	1760.23†	−129.27†	−296.47
Spline−BST10	−449.89†	−258.16†	894168.3†	−140.85†	941.19†	1223.47†	−178.46†	−119.9	−323.89†	−287.49	97.06†	1693.48†	−129.25†	−296.41
Spline−DIS3	−444.66	11.91	1215298.95†	151.95†	2173.85	2988.47	−178.3	−119.91	−316.77	−288.65	95.94	2265.62	−129.1	−296.31†
Spline−DIS5	−446.31†	−27.49†	1564324.95†	155.87†	1397.71	3298.73	−178.01	−119.9	−318.4	−288.17	96.14	3144.26	−129.09	−296.42
Spline−DIS10	−442	189.68	2333430.22†	403.14†	2520.19	3825	−178.31†	−119.92	−317.87	−292.67†	96.43	2803.83	−129.21†	−296.35
Spline−RND3	−449.96†	−390.02†	825508.08†	−269.14†	1452.06†	1887.33	−178.99†	−119.89	−322.37†	−294.82†	96.7	2227.35	−129.23†	−296.46
Spline−RND5	−449.04†	−378†	912587.82†	−285.24†	1068.86†	1582.92†	−178.29	−119.9	−322.85†	−293.09†	96.72	2681.79	−129.24†	−296.5
Spline−RND10	−447.26†	−315.03†	1097016.49†	−138.69†	1342.57†	3415.72	−178.1	−119.89	−318.63	−297.78†	97.04†	3660.93	−129.13	−296.48
Newton−BST3	−449.92†	−222.3†	1088009.58†	−24.61†	1320.09†	951.15†	−178.28†	−119.93	−325.38†	−290.73†	97.1†	1499.62†	−129.18	−296.37
Newton−BST5	−449.88†	−282.92†	1302796.27†	−177.06†	1219.89†	1816.16†	−178.38†	−119.91	−324.09†	−283.87	96.96†	1624.2†	−129.23†	−296.34†
Newton−BST10	−449.91†	−269.11†	1043788.28†	−192.28†	1037.99†	1046.51†	−178.48†	−119.89	−324.59†	−282.3	96.56	1902.99†	−129.28†	−296.47
Newton−DIS3	−432.54	−48.45†	926684.82†	182.46†	2591.17	2205.05	−177.98	−119.91	−317.26	−288.24	95.43†	2699.62	−129.25†	−296.4
Newton−DIS5	−440.33	−5.52†	1892874.1†	400.36†	2809.02	20998.53	−177.74	−119.87	−314.79	−291.19	96.1	3101.58	−129.16	−296.51
Newton−DIS10	−442.37	64.79	2570233	507.12†	1169.78†	3362.79	−177.25	−119.9	−314.81	−291.8†	96.53	2857.18	−129.15	−296.31†
Newton−RND3	−449.97†	−389.88†	569467.54†	−305.42†	1524.2	1318.48†	−178.5†	−119.9	−322.91†	−285.49	96.39	1492.15†	−129.19	−296.57
Newton−RND5	−448.15†	−313.72†	1050597.74†	−199.77†	1385.18	2467.76	−178.47†	−119.89	−318.66	−293.95†	96.71	1937.32	−129.24†	−296.47

Table 3 continued

Method	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}
Newton−RND10	−443.95	−109.22†	1613831.5†	0.6†	2337.95	2973	−177.39	−119.88	−317.17	−296.4†	96.2	3225.04	−129.12	−296.51
OLS−BST	−450†	559.63†	3617660	1343.82†	2393.34	3948.39	−177.58	−119.9	−312.86	−282.75	97.28†	4286.03	−128.92	−296.31†
OLS−DIS	−450†	538.24	3539541	1867.47	2305.32	3983.74	−177.6	−119.89	−314.52	−281.25	97.6†	4359.67†	−128.84	−296.32†
OLS−RND	−450†	376.47	3837140	1652.45	2408.79	15393.14	−177.86	−119.9	−313.87	−282.28	97.44†	3265.02	−128.84	−296.4
SVR−BST3	−426.64	91.3	4065658	790.6†	2458.76	5481.64	−177.02	−119.89	−313.73	−282.11	97.21†	3562.28	−128.87	−296.42
SVR−BST5	−441.55	99.95	2990677	809.31†	2509.09	2112.06	−177.81	−119.89	−316.18	−274.96	97.19†	2761.54	−128.89	−296.26†
SVR−BST10	−442.19	187.68	3023460	738.06†	2057.05	3348.84	−176.97	−119.89	−315.45	−285.85	97.37†	2798.39	−128.79	−296.29†
SVR−DIS3	−444.15	237.32	4064241	941.43†	2055.91	20456.17	−176.72	−119.9	−315.24	−287.89	97.42†	3037.76	−128.85	−296.34
SVR−DIS5	−439.18	263.48	3503445	831.27†	2473.14	2846.32	−177.14	−119.9	−311.29	−283.09	97.56†	3899.99	−128.91	−296.22†
SVR−DIS10	−423.74	420.05	4459042	823.04†	2715.31	3491.04	−177.56	−119.92	−312.53	−284.94	97.22†	3449.06	−128.93	−296.36
SVR−RND3	−442.46	311.95	3303740	734.16†	1900.99	2632.49	−177.13	−119.89	−310.99	−283.3	97.06†	3149.49	−128.94	−296.31†
SVR−RND5	−443.38	227.1	3920565	838.7†	2308.59	2039.71	−177.65	−119.89	−315.4	−284.39	97.39†	3265.26	−128.91	−296.3†
SVR−RND10	−440.78	399.52	2876402	785.14†	2315.02	2544.44	−177.54	−119.9	−309.91	−284.97	97.21†	3501.03	−128.87	−296.42

The fitness value with † mark presents that this algorithm is significantly better than canonical FWA by Wilcoxon signed−rank in the significant level $p < 0.05$

the RND method is a practical sampling selection method that can be reasonably applied in the real world conditions.

5.3 Analysis of sampling size influence

Theoretical speaking, a large sampling size means more accuracy in approximation performance, except over fitting issues. From the evaluation results (i.e., Tables 3 and 4), we can obtain the same conclusion by comparing the same approximation method with different sampling sizes. However, more sampling size requires more time spent in the approximation process. When we apply one of the approximation methods with a certain sampling size mentioned in this paper, we should consider the final optimization results and time cost used in approximation process to achieving a balance state for a better optimization performance.

5.4 Analysis of obtained elite rank

One characteristic of our proposed method for enhancing the FWA optimization performance is obtaining an elite from simplified approximation fitness landscape. For approximation method in low dimensional space, we obtain elite in each lower dimensional space and combine these elite into original space. The fitness rank of obtained elite is one of the evaluation metrics for evaluating performance of our proposed method. Here, we discuss this issue based on Table 5, which shows the average elite rank from our evaluation experiments. From Table 5, the average ranks of elite obtained by the BST method, the RND method and the DIS method are about 2, 5, and 6, respectively. The rank of elite presents the accuracy and performance of each approximate method and each sampling method.

From this average rank result and mean value of each method obtained from Tables 3 and 4, we found that sampling method is an essential factor that influences the optimization performance of our proposed surrogate-assisted FWA. The influence of the sampling method on the optimization performance enhancement is more effective than that of sampling size. On the contrary, if the sampling method is inefficient, whatever the sampling size is, the proposed surrogate-assisted FWA cannot be enhanced and improved significantly. This is a new discovery arising from our evaluation experiments.

5.5 Analysis of interpolation and extrapolation of obtained elite

The primary objective of our proposed method is to approximate fitness landscape with a simple shape, so that we can roughly find promising global optimum region to

Table 4 Mean fitness value of F15–F25 with 10 dimension

Method	f_{15}	f_{16}	f_{17}	f_{18}	f_{19}	f_{20}	f_{21}	f_{22}	f_{23}	f_{24}	f_{25}
Normal	340.16	354.47	367.22	910	899.05	910	145.75	1312.65	1465.55	1266.35	1233.37
LS1-BST3	357.54	315.17†	358.34	903.36	910	910	1334.14	1287.47	1347.34	1195.87	1211.92
LS1-BST5	344.51	309.48†	339.62	910	910	910	1265.84†	1240.96†	1411.41	1171.43	1243.55
LS1-BST10	262.14†	332.2†	335.6	910	910	906.68	1289.92	1234.98†	1341.27	1195.16	1147.69
LS1-DIS3	308.37	340.78	351.17	910	910	906.69	1294.18	1273.91†	1386.8	1209.28	1032.63
LS1-DIS5	319.88	318.58†	356.68	910	910	910	1388.08	1247.67†	1458.21	1164.79	1044.72
LS1-DIS10	334.73	321.64†	345.82	910	899.25	910	1341.51	1225.36†	1292.33†	1091.6	1027.73
LS1-RND3	397.81	306.97†	345.66	910	910	910	1227.54†	1226.6†	1252.72†	1085.68	1194.81
LS1-RND5	327.68	309.34†	332.2†	906.7	906.69	910	1284.35	1249.68†	1358.92	1112.71	1065.58
LS1-RND10	353.65	295.95†	337.04	910	910	910	1438.49	1255.8	1254.12†	930.92†	1215.71
LS2-BST3	391.73	314.24†	353.79	907.31	910	910	1276.67†	1268.75	1283.6†	985.34†	922.65†
LS2-BST5	324.93	320.25†	323.63†	910	910	902.68	1335.73	1279.87	1246.34†	891.72†	1043.86
LS2-BST10	331.02	324.96†	345.05	910	910	910	1437.01	1270.78†	1185.12†	862.58†	786.74†
LS2-DIS3	306.85	317.02†	319.9†	910	910	910	1338.18	1224.89†	1307.18†	754.94†	859.67†
LS2-DIS5	342.62	301.65†	324.21†	910	910	910	1314.91†	1215.73†	1341.27†	712.9†	632.18†
LS2-DIS10	376.87	295.44†	308.5†	903.45	910	910	1137.49†	1131.81†	1236.72†	642.33†	580.64†
LS2-RND3	348.45	306.21†	305.55†	906.67	910	906.67	1292.36	1228.29†	1323.84†	903.38†	839.89†
LS2-RND5	339.32	302.14†	310.4†	908.17	910	894.02	1331.99	1212.39†	1323.47†	801.76†	841.93†
LS2-RND10	411.44	293.62†	318.24†	906.68	906.69	906.68	1204.18†	1178.71†	1313.45†	771.28†	735.21†
Spline-BST3	360.71	329.85†	340.09	892.53	910	910	1352.12	1228.52†	1255.02†	983.17†	1006.88
Spline-BST5	347.33	308.89†	339.53	910	910	910	1345.99	1257.68	1288.41†	710.01†	703.95†
Spline-BST10	332.89	325.3†	331.46†	910	910	906.69	1360.34	1257.97†	1232.96†	592.28†	664.01†
Spline-DIS3	308.51	307.12†	321.01†	910	897.54	910	1204.07†	1225.84†	1355.62	550.51†	696.96†
Spline-DIS5	269.16	306.49†	317.29†	910	901.41	910	1373.23	1174.35†	1383.61	602.23†	552.81†
Spline-DIS10	322.63	299.11†	326.97†	910	906.8	910	1347.24	1228.09†	1248.46†	586.59†	484.93†
Spline-RND3	376.83	302.59†	322.68†	900.02	906.67	910	1301.39	1250.29†	1245.65†	789.09†	790.49†
Spline-RND5	307.3	292.15†	310.09†	905.16	910	892.54	1444.25	1239.6†	1245.14†	678.41†	694.62†
Spline-RND10	350.39	281.49†	293.82†	896.52	910	906.7	1192.21†	1203.38†	1229.96†	533.59†	554.19†
Newton-BST3	336.2	320.26†	345.13	897.39	910	910	1394.47	1275.63	1260.61†	961.87†	814.9†
Newton-BST5	346.33	316.93†	359.37	908.92	906.68	910	1363.35	1234.88†	1302.64†	793.32†	1039.07
Newton-BST10	373.71	315.97†	350.54	910	910	910	1285.76	1249.44†	1290.51†	537.27†	533.59†
Newton-DIS3	344.12	292.97†	322.62†	910	910	900.19	1331.69	1172†	1348.93	554.34†	779.32†
Newton-DIS5	293.21	300.04†	317.9†	910	910	910	1374.93	1224.84†	1252.63†	644.01†	799.26†
Newton-DIS10	263.68	309.35†	319.67†	910	910	910	1323.63	1259.07†	1329.89	634.88†	702.7†
Newton-RND3	359.21	307.62†	322.73†	910	910	890.43	1243.67†	1211.18†	1303.19†	748.81†	666.79†
Newton-RND5	346.64	298.06†	314.96†	898.75	900.07	910	1349.1	1189.05†	1282.72†	961.56†	809.37†
Newton-RND10	384.83	281.32†	302.47†	910	910	910	1227.61†	1214.41†	1289.92†	730.17†	918.82†
OLS-BST	323.65	334.42†	373.7	910	910	910	1506.18	1310.7	1464.12	1174.51	1205.43
OLS-DIS	418.33	342.6	354.52	910	910	910	1438.71	1299.62	1427.72	1038.17†	1101.28
OLS-RND	337.8	338.82†	381.78	910	910	910	1537.26	1290.63	1438.77	1183.43	1264.36
SVR-BST3	338.64	339.07	352.95	910	910	910	1492.54	1338.57	1417.2	1375.7	1340.49
SVR-BST5	362.44	326.14†	380.58	910	910	910	1419.81	1323.67	1410.95	1414.75	1089.03
SVR-BST10	357.54	357.94	358.49	910	910	910	1445.33	1337.11	1400.47	1250.93	1175.33
SVR-DIS3	304.78	357.39	381.44	910	910	910	1459.13	1326.04	1608.32†	1333.63	1193.76
SVR-DIS5	357.54	359.95	360.65	910	910	910	1469.35	1300.15	1502.86	1233.33	1345.24
SVR-DIS10	357.54	361.12	346.53	910	910	907.12	1531.2	1304.61	1425.09	1201.96	1188.03
SVR-RND3	326.5	341.8	364.48	910	910	910	1499.58	1299.03	1482.22	963.23†	1277.5
SVR-RND5	357.54	335.69	362.71	910	904.7	910	1487.42	1310.99	1430.69	1185.11	1336.99

Table 4 continued

Method	f_{15}	f_{16}	f_{17}	f_{18}	f_{19}	f_{20}	f_{21}	f_{22}	f_{23}	f_{24}	f_{25}
SVR-RND10	357.54	341.91	371.05	910	910	910	1336.42	1315.77	1347.34	1078.39†	1214.85

The abbreviations used here are as in Table 1. The fitness value with † mark presents that this algorithm is significantly better than the canonical FWA by Wilcoxon signed-rank in the significant level $p < 0.05$

Table 5 Average elite rank of 10-D and 30-D benchmark function when generation (G) equals to 10, 100 and 1000

Method	10-D			30-D		
	G = 10	G = 100	G = 1000	G = 10	G = 100	G = 1000
LS1-BST3	3.46	3.59	4.03	4.21	4.89	5.34
LS1-BST5	2.48	2.39	2.32	2.84	2.64	2.67
LS1-BST10	2.68	2.30	2.24	2.55	2.27	2.27
LS1-DIS3	6.96	7.29	7.51	6.81	7.59	7.61
LS1-DIS5	6.35	6.49	6.53	6.29	6.58	6.61
LS1-DIS10	6.09	6.12	6.18	5.73	5.76	6.02
LS1-RND3	5.23	5.07	5.04	5.36	5.05	5.07
LS1-RND5	5.29	4.96	5.14	5.12	5.10	5.05
LS1-RND10	5.21	5.29	5.24	5.05	5.19	5.13
LS2-BST3	2.19	1.94	1.96	2.26	1.99	1.97
LS2-BST5	2.14	**1.88**	**1.90**	2.15	1.99	1.92
LS2-BST10	**2.11**	1.91	1.91	2.21	**1.97**	**1.90**
LS2-DIS3	5.90	5.72	5.78	5.60	5.70	5.69
LS2-DIS5	5.42	5.31	5.54	4.94	5.12	5.37
LS2-DIS10	4.87	5.37	5.34	4.47	4.89	4.98
LS2-RND3	4.49	4.13	4.10	4.33	4.36	4.31
LS2-RND5	4.49	4.30	4.45	4.32	4.48	4.58
LS2-RND10	4.71	4.94	5.02	4.36	4.76	4.97
Spline-BST3	**2.11**	2.03	2.06	2.17	2.06	2.01
Spline-BST5	2.32	2.13	2.10	2.51	2.19	2.14
Spline-BST10	2.66	2.26	2.31	2.61	2.30	2.26
Spline-DIS3	5.68	5.51	5.64	5.33	5.33	5.43
Spline-DIS5	6.02	5.82	6.01	5.69	5.79	5.89
Spline-DIS10	6.30	6.26	6.34	5.98	6.12	6.25
Spline-RND3	4.49	4.05	4.07	4.29	4.21	4.18
Spline-RND5	5.19	5.19	5.21	5.05	5.19	5.15
Spline-RND10	5.95	5.91	5.95	5.84	5.98	5.90
Newton-BST3	2.19	2.00	2.07	**2.12**	2.06	2.03
Newton-BST5	2.43	2.11	2.16	2.34	2.09	2.11
Newton-BST10	2.58	2.27	2.41	2.60	2.38	2.34
Newton-DIS3	5.67	5.56	5.66	5.39	5.28	5.58
Newton-DIS5	6.11	6.07	6.06	5.70	5.87	6.00
Newton-DIS10	6.45	6.48	6.63	6.29	6.51	6.48
Newton-RND3	4.48	4.18	4.08	4.31	4.35	4.24
Newton-RND5	5.59	5.55	5.55	5.50	5.62	5.47
Newton-RND10	6.56	6.62	6.71	6.49	6.59	6.57
OLS-BST	5.65	5.54	5.43	7.25	6.37	6.33
OLS-DIS	7.02	6.95	6.75	7.29	6.56	6.36

Table 5 continued

Method	10-D			30-D		
	G = 10	G = 100	G = 1000	G = 10	G = 100	G = 1000
OLS-RND	6.96	6.74	6.62	7.51	6.93	6.96
SVR-BST3	7.58	7.21	7.16	7.21	7.05	7.09
SVR-BST5	7.51	7.13	7.21	6.91	6.69	6.69
SVR-BST10	7.36	6.75	6.73	6.99	6.71	6.76
SVR-DIS3	7.62	7.17	7.13	7.29	7.09	7.05
SVR-DIS5	7.46	6.78	6.74	7.02	6.67	6.73
SVR-DIS10	7.55	6.70	6.77	7.14	6.67	6.72
SVR-RND3	8.05	7.56	7.69	7.63	7.43	7.47
SVR-RND5	7.72	7.33	7.30	7.30	7.12	7.18
SVR-RND10	7.06	6.84	6.75	7.01	6.97	7.03

There are 8 individuals in our evaluation experiments, the number in this table shows the average rank of elite according to its fitness value. The rank with bold font presents the winner algorithms in each generation of 10-D and 30-D benchmark functions

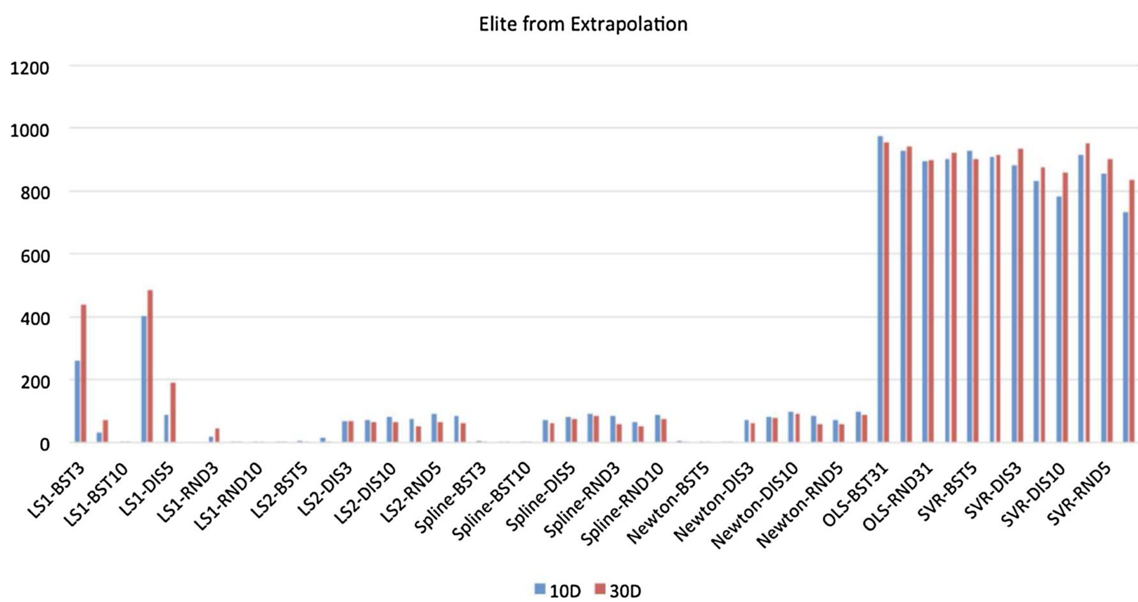


Fig. 2 Extrapolation times of each method, when we obtain the elite from approximated fitness landscape, we judge whether it comes from the interval of sampling data, if it is out of the interval, we define it as extrapolation one time. This figure presents that better local search can improve the optimization performance of FWA. Y-axis and X-axis show the number of average extrapolation times from 25 benchmark functions of each method and each method's abbreviation, respectively

search from this simple shape. From the approximation point of view, the elite is obtained from interpolation or from extrapolation, i.e., the elite is from approximation range or outside of it. This metric is one of the evaluation factors to evaluate approximation performance of our proposal.

Figure 2 presents average extrapolation times of each approximation method applied to 25 benchmark problems. Except for some special cases of approximation in lower dimensional spaces, these methods can obtain the elite from extrapolation, i.e., outside of approximation range, with about 100 times in 1000 generations. Although the elite comes from these methods rarely outside its approximation range, from the evaluation results, these methods can obtain a better optimization performance in most benchmark problems. It indicates that better local search can improve the optimization performance of surrogate-assisted FWA. The times of elite from extrapolation by approximation in original space and high dimensional space are more than that of approximation in low dimension. However, as demonstrated in Tables 3 and 4, they cannot present better acceleration performance for most benchmark problems. This result indicates that wrong approximation and over-fitting problem can happen when applying these two types of approximation method.

6 Conclusion

In this paper, we investigated and discussed the effectiveness of approximation strategy in surrogate-assisted FWA. We discussed the effectiveness of approximation strategy from the aspect of approximation method, sampling data selection method and sampling size. We analysed and studied the optimization performance of each method. For the approximation method, we used least square approximation, spline interpolation, Newton interpolation, and support vector regression to approximate fitness landscape of FWA in projected lower dimensional, original and higher dimensional search space. We found that approximation in lower dimensional search space can effectively obtain the rough fitness landscape information, and its optimization performance outperforms that of approximation in original and higher dimensional search space. The problem of over-fitting problem frequently happens when applying the SVR method in this fitness landscape approximation problem. In the sampling data selection method aspect, we defined three sampling data selection methods, i.e., the best sampling method, the distance near the best fitness individual sampling method, and the random sampling method to investigate each sampling method's performance. We found that the RND sampling selection method is better than the other two methods from the viewpoints of computational complexity and performance of optimization. From the sample size viewpoint, we set it as 3, 5, and 10 sampling data in each approximation method and sampling selection method. We notice that we should balance the final acceleration performance we obtained and time cost in approximation process we spent. This is a crucial issue when we apply proposed FWA to a variety of problems and real-world applications.

One of the advantages of FWA is that the multi-production mechanism presents in its algorithm. This means, one firework can generate a number of sparks within the explosion amplitude in each generation. We can use this local fitness information and combine several sparks to explore the global fitness landscape by our approximation method. This is one aspect of our future work. The IEC application requires the optimization algorithm to have a better performance with small number of generations and population size. The fact that the FWA can obtain a relatively better optimization performance with a small population size, is another advantage. This characteristic provides an opportunity to establish an interactive framework of FWA, i.e., interactive FWA. Because the evaluation space of humans is relatively simple, we can obtain a better optimization result when we apply our proposed surrogate-assisted FWA in interactive FWA by fitness landscape approximation. Some other opportunities for investigation, such as surrogate-assisted FWA selection issue, surrogate-assisted FWA for hard optimization problems, will be involved in our future work.

Acknowledgments This work was supported by the JSPS Grant-in-Aid for Scientific Research (23500279), Japan. This work was also supported by the Natural Science Foundation of China (NSFC) under grant no. 61375119 and 61170057, and partially supported by National Key Basic Research Development Plan (973 Plan) Project of China with grant no. 2015CB352302.

References

1. Brest J, Greiner S, Boskovic B, Mernik M, Zumer V (2006) Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems. IEEE Trans Evol Comput 10(6):646–657
2. Gong W, Cai Z, Ling CX, Li C (2011) Enhanced differential evolution with adaptive stategies for numerical optimization. IEEE Trans Syst Man Cybern 41(2):397–413
3. Hazewinkel M (2001) Encyclopedia of mathematics, chapter Spline interpolation. Springer, New York
4. Janecek A, Tan Y (2011) Swarm intelligence for non-negative matrix factorization. Int J Swarm Intell Res (IJSIR) 2(4):12–34
5. Jin Y (2005) A comprehensive survey of fitness approximation in evolutionary computation. Soft comput 9(1):3–12
6. Kashan AH (2015) A new metaheuristic for optimization: optics inspired optimization (OIO). Comput Oper Res 55:99–125
7. Ong Y-S, Lim MH, Chen X (2010) Research frontier: memetic computation—past, present and future. IEEE Comput Intell Mag 5(2):24–31
8. Pei Y (2014) Chaotic evolution: fusion of chaotic ergodicity and evolutionary iteration for optimization. Natural Comput 13(1):79–96
9. Pei Y (2014) Study on effecient search in evolutionary computation. Doctoral Dissertation, Kyushu University, Japan
10. Pei Y (2015) From determinism and probability to chaos: chaotic evolution towards philosophy and methodology of chaotic optimization. Sci World J 704587
11. Pei Y, Takagi H (2012) Fourier analysis of the fitness landscape for evolutionary search acceleration. In: 2012 IEEE Congress on Evolutionary Computation (CEC2012), Brisbane, Austrilia, pp 2934–2940
12. Pei Y, Takagi H (2013) Accelerating IEC and EC searches with elite obtained by dimensionality reduction in regression spaces. J Evolut Intell 6(1):27–40
13. Pei Y, Takagi H (2013) Fitness landscape approximation by adaptive support vector regression with opposition-based learning. In: 2013 IEEE International Conference on Systems, Man, and Cybernetics (SMC2013), Manchester, UK, pp 1329–1334
14. Pei Y, Takagi H (2013) Triple and quadruple comparison-based interactive differential evolution and differential evolution. In: Foundations of Genetic Algorithms Workshop XII (FOGA 2013), Adelaide, Australia, pp 173–182
15. Pei Y, Takagi H (2015) Local information of fitness landscape obtained by paired comparison-based memetic search for interactive differential evolution. In: 2015 IEEE Congress on Evolutionary Computation (CEC2015), Sendai, Japan, pp 2215–2221

16. Pei Y, Takagi H (2012) Comparative study on fitness landscape approximation with fourier transform. In: 2012 Sixth International Conference on Genetic and Evolutionary Computing (ICGEC2012), Kitakyushu, Japan, pp 400–403
17. Pei Y, Zheng S, Tan Y, Takagi H (2012) An empirical study on influence of approximation approaches to enhance fireworks algorithm. In: 2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC2012), Seoul, Korea, pp 1322–1327
18. Qin AK, Huang VL, Suganthan PN (2009) Differential evolution algorithm with strategy adaption for global numberical optimization. IEEE Trans Evol Comput 13(2):398–417
19. Sadollah A, Eskandar H, Bahreininejad A, Kim JH (2015) Water cycle algorithm with evaporation rate for solving constrained and unconstrained optimization problems. Appl Soft Comput 30:58–71
20. Suganthan PN, Hansen N, Liang JJ, Deb K, Chen YP, Auger A, Tiwari S (2005) Problem definitions and evaluation criteria for the CEC2005, special session on real-parameter optimization. KanGAL Report 2005005:2005
21. Takagi H (2001) Interactive evolutionary computation: fusion of the capabilities of EC optimization and human evaluation. Proc IEEE 89(9):1275–1296
22. Takagi H, Ingu T, Ohnishi K (2003) Accelerating a GA convergence by fitting a single-peak function. J Jpn Soc Fuzzy Theory Intell Inform 15(2):219–229 in Japanese
23. Tan Y, Zhu Y (2010) Fireworks algorithm for optimization. In: Advances in Swarm Intelligence, Springer, New York, pp 355–364
24. Zhang B, Zhang M-X, Zheng Y-J (2014) A hybrid biogeography-based optimization and fireworks algorithm. In: 2014 IEEE Congress on Evolutionary Computation (CEC2014), Beijing, China, pp 3200–3206
25. Zhang J, Sanderson AC (2009) JADE: adaptive differential evolution with external achive. IEEE Trans Evol Comput 13(5):945–958
26. Zheng S, Janecek A, Tan Y (2013) Enhanced fireworks algorithm. In: 2013 IEEE Congress on Evolutionary Computation (CEC2013), Cancun, Mexico, pp 2069–2077
27. Zheng YJ (2015) Water wave optimization: a new nature-inspired metaheuristic. Comput Oper Res 55:1–11
28. Zheng YJ, Qin S, Chen SY (2013) Multiobjective fireworks optimization for variable-rate fertilization in oil crop production. Appl Soft Comput 13(11):4253–4263