

# Accelerating IEC and EC searches with elite obtained by dimensionality reduction in regression spaces

Yan Pei · Hideyuki Takagi

Received: 16 November 2012 / Revised: 2 May 2013 / Accepted: 3 May 2013 / Published online: 16 May 2013  
© Springer-Verlag Berlin Heidelberg 2013

**Abstract** We propose a method for accelerating interactive evolutionary computation (IEC) and evolutionary computation (EC) searches using elite obtained in one-dimensional spaces and use benchmark functions to evaluate the proposed method. The method projects individuals onto  $n$  one-dimensional spaces corresponding to each of the  $n$  searching parameter axes, approximates each landscape using interpolation or an approximation method, finds the best coordinate from the approximated shape, obtains the elite by combining the best  $n$  found coordinates, and uses the elite for the next generation of the IEC or EC. The advantage of this method is that the elite may be easily obtained thanks to their projection onto each one-dimensional space and there is a higher possibility that the elite individual locates near the global optimum. We compare the proposal with methods for obtaining the landscape in the original search space, and show that our proposed method can significantly save computational time. Experimental evaluations of the technique with differential evolution using a simulated IEC user (Gaussian mixture model with different dimensions) and 34 benchmark functions show that the proposed method substantially accelerates IEC and EC searches.

**Keywords** Interactive evolutionary computation · Evolutionary computation · Fitness landscape · Convergence acceleration · Dimensionality reduction · Elite strategy

## 1 Introduction

The success of evolution in nature outstrips that of evolutionary computing even though it is based on the same principles. This can be attributed primarily to the abundance of resources in nature, such as time and memory, which are constrained in a computer system. This is especially true in the case of systems which involve human interactivity, such as interactive evolutionary computation (IEC). Consequently, accelerating evolutionary computation (EC) is necessary for many EC applications to improve the performance of their target systems. Consider the case of an IEC that optimizes a target system based on the IEC user's subjective evaluations; user fatigue is a serious problem limiting such a system's practical application. Multiple trials for accelerating IEC and EC have therefore been proposed [12, 23].

One of the methods taken for accelerating EC is to approximate the fitness landscape and apply an elite strategy [22]. If the search space is high dimensional, non-linear, non-differentiable and non-convex, conventional approximation methods encounter difficulties dealing with the complex space, and the best solution is to reduce the search space dimension. This kind of dimensionality reduction strategy was originally proposed in Pei and Takagi [10, 13] and applied to the traveling salesman problem in a real-world application [11].

We have used interpolation or approximation methods to obtain the fitness landscape in a dimensionally reduced

---

Y. Pei (✉)  
Graduate School of Design, Kyushu University, 4-9-1, Shiobaru,  
Minami-ku, Fukuoka 815-8540, Japan  
e-mail: peiyan@kyudai.jp

H. Takagi  
Faculty of Design, Kyushu University, 4-9-1, Shiobaru,  
Minami-ku, Fukuoka 815-8540, Japan  
e-mail: takagi@design.kyushu-u.ac.jp

search space [10, 13]. Reaching the global optimum from the elite is easier in the regression space. Although it may not be the actual global optimum, it may be a neighbor to the global optimum in the original search space. Finding the actual global optimum from the elite is therefore an easier task. Here, *regression space* refers to a lower dimensional space consisting of  $k$  dimensional ( $k$ -D) axes of the original  $n$ -D axes, where we use  $k = 1$  in this paper. In other words, individuals are projected onto the  $k$ -D space to simplify finding the elite. This elitism does not destroy the original EC search space so much by approximation, but can accelerate the IEC and EC convergence with less computational cost by conducting the elite search in the  $n$  regression spaces of  $k$ -D ( $k = 1$ ).

We extend our previous works [10, 13] and present a comprehensive investigation of our proposal through the series of experiments which we present in this paper. Specifically, we investigate: (1) accelerated convergence of our proposals in a simulated IEC user model. (2) the computational cost of our proposal in comparison to the work in [22] for ordinary EC, and (3) EC acceleration performance for higher dimensional benchmark tasks. A Gaussian mixture model with different dimensions and 34 well-known benchmark functions are employed in our experiments. Some related analysis and discussions are involved in each investigation. From a framework viewpoint, this method can be embedded in any IEC and EC algorithm in order to obtain accelerated convergence. The investigation of the above-listed three topics contributes to this paper's originality.

Following this introductory section, an overview of methods used for accelerating EC is reported in Sect. 2. We introduce the interpolation and approximation methods which relate to our study in Sect. 3. In Sect. 4, we explain in detail our proposed method is enabled by a technique for reducing the dimensionality of the search space, and we show how the elite can be obtained from regression search spaces. In Sects. 5 and 6, IEC and EC are experimental evaluated using a Gaussian mixture model with various different dimensions and 34 benchmark functions, and their results and related topics are analyzed and discussed. Finally, we discuss some general issues in Sect. 7 and conclude with the performance of our proposed methods presented here and discuss some open topics, further opportunities and future works in Sect. 8.

## 2 Conventional techniques

There are many complex optimization tasks in many areas that involve combinatorial, dynamic, severely non-linear, non-differentiable, and non-convex space problems. Although EC is a good metaheuristic optimization tool for them, its search performance for such complex tasks

naturally decreases. EC usually finds areas of better fitness easily and quickly, however, finding the global optimum accurately may be time-consuming. Several conventional accelerating EC convergence methods have been proposed to solve this problem [12]. They include approximating the fitness landscape [15, 16, 22], adaptive evolution control methods [6], developing new mechanisms embedded into existent EC algorithms [14], information fusion methods [5, 8, 27] and so on.

Using the EC landscape information directly is a promising method for EC acceleration. A single-peak function is used to obtain the landscape of the original search space [Eq. (1)], and an elite individual is found from the peak of that landscape's curve to accelerate the EC search [22]. This method can be considered both high return and low risk; the elite is obtained from a fitted single-peak function and the worst fitness individual is simultaneously removed from the population, which does not significantly destroy the original search space. It is, however, costly to fit the single peak function in a high dimensional search space with the requirement that it needs at least  $2D + 1$  individuals in the fitting process, where  $D$  is the number of dimensions of the problem. Our proposed methods obtain the fitness landscape in lower dimensional search spaces, and thus significantly reduce the computational cost in the fitting stage. In this paper, we compare our proposed methods with the method in [22].

$$f(X) = \sum_{i=1}^n a_i(x_i - b_i) + c \quad (1)$$

Adaptive evolution control methods [6] control and self-adaptively tune the EC parameters, such as operator rate, or use a dynamic search strategy to balance global exploration and local exploitation such that EC performance is enhanced. In differential evolution, there have been some novel strategies proposed in recent years, such as SaDE [18], JADE [20], jDE [1] and JASaDE [4].

Information fusion methods are an important acceleration method. In this case, EC is fused with other optimization techniques to change the search strategy, to decide the search direction, or to choose high priority individuals, etc. These fusion techniques include local search [2, 24, 27], simulated annealing [8], artificial neuron networks [5, 7], and others too numerous to mention here.

## 3 Approximating the fitness landscape by interpolation and approximation

### 3.1 Regression function

The computation of each EC generation produces a set of discrete individuals and their corresponding fitness  $(x_i, y_i), i = 0, 1, \dots, m$ , but the analytical expression

relating the two cannot be known. We make the data regression curve for the required function from the given category  $\Phi$ .

Because there must be some errors in the observed discrete data, we do not require interpolated or approximated curves pass through all the discrete points exactly, but rather that it approaches the original curve at its discrete points, i.e. near  $x_i$ , as much as possible.

We let the function  $\varphi(x)$  belong to  $\Phi$ , and  $\delta_i = \varphi(x_i) - y_i$  be the error at  $\varphi(x_i)$ , such that the error vector is  $\delta = (\delta_0, \delta_1, \dots, \delta_m)^T$ . In the regression calculation process, we let the norm of the vector  $\|\delta\|$  be minimized. For a different norm, we can construct a different regression function. The function series is shown in Eq. (2).

$$\Phi = \text{Span}\{\varphi_0(x), \varphi_1(x), \dots, \varphi_n(x)\} \tag{2}$$

The members in Eq. (2), i.e.  $\varphi_0(x), \varphi_1(x), \dots, \varphi_n(x)$ , are linearly independent in the interval  $[a, b]$ , where the nodes  $x_i$  are contained. Any type of function  $\varphi(x)$  can be expressed as Eq. (3).

$$\varphi(x) = a_0\varphi_0(x) + a_1\varphi_1(x) + \dots + a_n\varphi_n(x) \tag{3}$$

The EC search space is complex for some nonlinear, non-differentiable, and non-convex space problems. The regression process brings additional cost to the EC, so a better solution is to select a simple function as the regression function to reduce its additional cost. A polynomial function is suitable for regression computing, so in this paper, we select it as the regression function in our interpolation and approximation methods.

### 3.2 Lagrange interpolation method

The Lagrange interpolation polynomial has the characteristic of being linear and unique. For one dimensional data from individuals  $x_0, x_1, \dots, x_n$ , we can set up an  $n$  degree polynomial  $l_0(x), l_1(x), \dots, l_n(x)$ . We set its type as  $l_i(x_j) = \delta_{ij}$ , where the form of  $\delta_{ij}$  is as shown in Eq. (4).

$$\delta_{ij} = \begin{cases} 0 & \text{if } i = j \\ 1 & \text{if } i \neq j \end{cases} \tag{4}$$

From the definition of  $l_k(x)$ , we can obtain Eq. (5), which is the  $n$  degree interpolation polynomial, where  $l_i(x)$  is an  $n$ th degree polynomial. The relationships of the Lagrange interpolation polynomial are shown in Eqs. (5), (6), (7) and (8).

$$p_n(x) = \sum_{k=0}^n l_k(x)y_k = y_i \tag{5}$$

$$l_k(x) = a(x - x_0) \dots (x - x_{k-1})(x - x_{k+1}) \dots (x - x_n) \tag{6}$$

When  $l_k(x_k) = 1$ ,

$$a = [(x_k - x_0) \dots (x_k - x_{k-1})(x_k - x_{k+1}) \dots (x_k - x_n)]^{-1} \tag{7}$$

i.e.,

$$l(x) = \prod_{i=1, i \neq k}^n \frac{(x - x_i)}{(x_k - x_i)} \tag{8}$$

Equations (8) and (9) are the  $n$  degree Lagrange interpolation basis function and the  $n$  degree Lagrange interpolation polynomial, respectively.

$$L_n(x) = \sum_{k=0}^n l_k(x)y_k, \quad i = 0, 1, \dots, n \tag{9}$$

In this paper, we use the binomial Lagrange interpolation polynomial as the simplified regression search space expression, and use the three individuals with relative better fitness as the interpolation points to obtain the concrete regression function. The concrete interpolation polynomial is as in Eq. (10). The Lagrange interpolation method is used to find the parameters of Eq. (10).

$$L(x) = \sum_{k=1}^3 \left\{ \prod_{i=1, i \neq k}^3 \frac{(x - x_i)}{(x_k - x_i)} \right\} y_k \tag{10}$$

### 3.3 Least squares approximation method

As is mentioned above, we want to minimize the norm of the error vector  $\delta = (\delta_0, \delta_1, \dots, \delta_m)^T$ . If we use the 2-norm form vector as the error vector, the calculation process is simplified. When one calculates the error vector norm for the 2-norm, the approximation method is referred to as the least squares method. Given data in the form of the function series, i.e., Eq. (2), we want to find a function that lets the error 2-norm vector be minimized, namely, Eq. (11).

$$\|\delta^*\|_2^2 = \sum_{i=0}^m \delta_i^{*2} = \sum_{i=0}^m [\varphi^*(x_i) - y_i]^2 = \min_{\varphi(x) \in \Phi} \|\delta^*\|_2^2 \tag{11}$$

The approximation function is shown in Eq. (12).

$$\varphi^*(x) = a_0^*\varphi_0(x) + a_1^*\varphi_1(x) + \dots + a_n^*\varphi_n(x) \tag{12}$$

If we can obtain  $\varphi_0(x), \varphi_1(x), \dots, \varphi_n(x)$ , the system will be orthogonal, i.e.  $(\varphi_i, \varphi_j) = 0 (i \neq j)$ , and the coefficient matrix equations will form a diagonal matrix [Eq. (13)].

$$a_k^* = \frac{(f, \varphi_k)}{(\varphi_k, \varphi_k)} \tag{13}$$

So the approximation function is as in Eq. (14).

$$\varphi^*(x) = \sum_{i=0}^n \frac{(f, \varphi_i)}{(\varphi_i, \varphi_i)} \varphi_i(x). \quad (14)$$

Similar to the use of Lagrange interpolation in Sect. 3.2 to obtain a regression search space by interpolation, here we use a linear function to obtain a linear approximation of the regression search space. The concrete expression is shown in Eq. (15).

$$\begin{pmatrix} (\varphi_0 \varphi_0) & (\varphi_0 \varphi_1) \\ (\varphi_1 \varphi_0) & (\varphi_1 \varphi_1) \end{pmatrix} \begin{pmatrix} D \\ E \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \end{pmatrix} \quad (15)$$

The approximation process is used to obtain the parameters  $D$  and  $E$  in Eq. (15) such that the regression search space may be defined in a linear space.

#### 4 Obtaining elite from regression search spaces by dimensionality reduction to accelerate IEC and EC searches

##### 4.1 Dimensionality reduction method

It is not easy for conventional interpolation or approximation methods to find an accurate regression search space corresponding to the original multi-dimensional space. An alternative method is to reduce the dimensionality of the original search space and find approximate curve expressions in the lower dimensional spaces. However, if the parameter variables of the fitness function are dependent, when they are separated into the lower dimensional space, the dimensionality reduction method will destroy the original search space information and the dependent parameter variables' relationships. Although there is a risk that some information will be lost when reducing the search space dimensionality, it is easy to interpolate or approximate the search space to obtain landscape information in lower dimensional search spaces with e.g. one or two dimensions of the original search space.

Our method for reducing the dimensionality of the searching space uses only one of the  $n$  parameter axes at a time instead of all  $n$  parameter axes, and projects individuals onto each 1-D regression space. The landscape of the  $n$ -D parameter space is given by a fitness function,  $y = f(x_1, x_2, \dots, x_n)$ , and the fitness value of the  $m$ -th individual are given by Eq. (16).

$$y_m = f(x_{1m}, x_{2m}, \dots, x_{nm}) \quad (m = 1, 2, \dots, M) \quad (16)$$

There are  $M$  individuals with  $n$ -D parameter variables. We project the individuals onto the  $n$  1-D spaces in  $i$ -th dimension as follows.

$$(x_{i1}, y_1)(x_{i2}, y_2) \dots (x_{im}, y_m)$$

Each of the  $n$  1-D regression spaces has  $M$  projected individuals. The original search space and dimensionally reduced search space are shown in Fig. 2. The dimensionality reduction method simplifies the regression computations, and it is easy to obtain a regression search space in lower dimension, which can be helpful in situations which involve a higher dimensional nonlinear search space.

##### 4.2 Method for simplifying the regression space landscape to select an elite

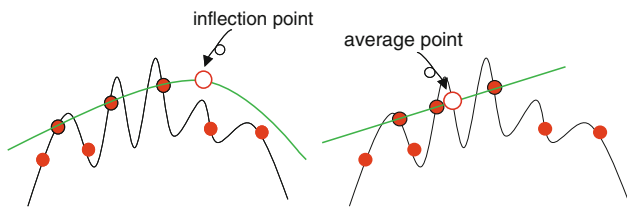
We interpolate or approximate the landscape of each 1-D regression space using the projected  $M$  individuals and select elite from the  $n$  approximated 1-D landscape shapes. In this paper, we test two methods for approximating the 1-D regression search spaces; in the first method we use a binomial Lagrange interpolation and in the other linear least squares approximation. The elite is generated from the resulting approximated shapes.

Finding the elite corresponds to a kind of local search in the area where relatively better individuals exist in the original search space. The global optimum is expected to be near this area and we may be able to find it with a probability higher than chance [22]. So the elite selection method is a critical step in the proposed acceleration processes.

As the elite obtained by the two different elite selection methods is different, it is expected that the acceleration performance will also differ. Further, the regression EC search space obtained by approximation or interpolation has its own characteristics and particularities, and we must use an efficient method to obtain an elite from this simplified search space after analyzing its characteristics.

Binomial Lagrange interpolation simplifies a regression space with a nonlinear curve, and it is easy to obtain its inflection point from its gradient, using the inflection point as the elite. The linear least squares approximation uses a linear function to approximate the regression space. Its gradient is either descent or ascent. Lacking an inflection point, a safer method, taking into account both descent and ascent, is to select the average point of the linear approximation line as the elite (see Fig. 1).

Our proposed methods replace the worst individual in each generation with an elite selected as above. 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.



**Fig. 1** New elite selection methods from a regression search space. *Left* by Lagrange interpolation method; *Right* by least squares approximation method

### 4.3 Method for synthesizing elite

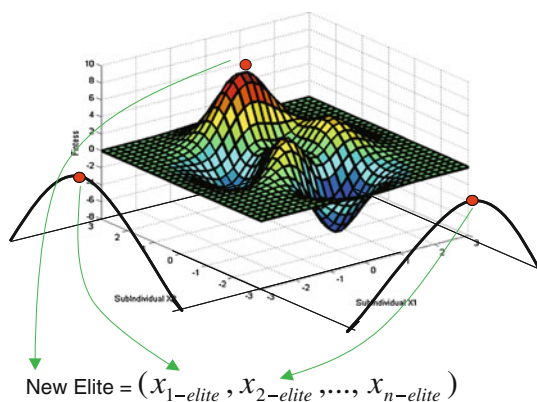
The methods in Sect. 4.2 select  $n$  elite points in  $n$  1-D regression spaces, respectively:  $x_{1-elite}, x_{2-elite}, \dots$ , and  $x_{n-elite}$ . The  $n$ -D elite used for accelerating IEC/EC convergence in the next generation is obtained as follows:

$$\text{New Elite} = (x_{1-elite}, x_{2-elite}, \dots, x_{n-elite}).$$

It is easier to calculate elite in a lower dimensional space than a higher dimensional space. Although we use 1-D as the lower dimensional space in this paper, in general the method need not be restricted to 1-D.

Once a new elite has been obtained, there are two methods for how it can be handled. In one cautious method, the fitness value of this elite is calculated to determine whether the new elite is really useful for acceleration, and in the other straightforward method, the new elite is inserted into the next IEC/EC iteration process without any prior consideration or judgment. In our proposed methods, we choose the first method. If the fitness of elite is better than the worst one, we place it into the next generation.

Our proposed method is based on the hypothesis that (1) elite calculated by interpolation or approximation from relatively better individuals will also have good fitness; (2)



**Fig. 2** Method for synthesizing elite. This proposed method represents a novel local search method for accelerating IEC/EC convergence

synthesizing an  $n$ -D elite from  $n$  elite points in  $n$  1-D regression spaces will also produce a good elite; (3) the probability of the global optimum being located near to the synthesized elite is high. (see Fig. 2)

In general, this proposed method represents a novel local search method for accelerating IEC/EC convergence, and it is this method that represents this paper’s principal original contribution.

## 5 IEC experimental evaluations and analysis

### 5.1 Methods for comparison

Table 1 shows our proposed methods along with conventional methods for comparison. We use differential evolution (DE) [17] in our experiments. As mentioned above, our proposed methods can be embedded as a framework in any IEC or EC algorithm to accelerate its search and is not limited to DE.

### 5.2 IEC user model and experimental conditions

Experimental evaluations frequently demand many repeated experiments under the same conditions, and evaluations with an IEC user model is necessary for this case rather than a real human IEC user. Our IEC user model [9] was designed based on the four specifications of (1) a relatively simple fitness landscape, (2) a multi-modal fitness landscape, (3) a big valley structure, and (4) parametrically controllability of the shape and complexity of the fitness landscape. The rationale of (1) is that although a human IEC user cannot distinguish differences less than the differential threshold of perceptions, he or she can nevertheless obtain practical solutions. The rationale of (2) is the observation that there are graphics, design, music, and others expressions for which fitness values are high but their expressions are quite different. That of the (3) is that an IEC user can reach the global optimum area easily in

**Table 1** Conventional methods and our proposed methods used for experimental comparison

(I)DE-N	Canonical DE [17]
(I)DE-TB	Fitting a single peak function using $n$ best individuals [22]
(I)DE-TN	Fitting a single peak function using $n$ distance nearest best individuals [22]
(I)DE-TA	Fitting a single peak function using all individuals [22]
(I)DE-LS	Fitting a linear function using least square approximation
(I)DE-LR	Fitting a binomial function using Lagrange interpolation

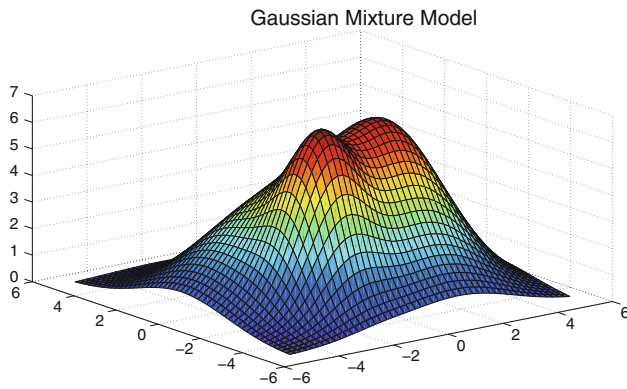


Fig. 3 3-D view of a Gaussian mixture model

spite of (2). The feature of the (4) is essential to conduct experiments with gradually changed several fitness landscapes.

A Gaussian mixture model (GMM) was established as pseudo-IEC user to simulate the user’s evaluation in [21]. The GMM consists of different means, variances and peaks mixed together to express the characteristics of a human user conducting an IEC evaluation experiment. We use a GMM for evaluation in this section. Concretely, we combine four Gaussian functions ( $k = 4$ ) and realize the characteristics expressed by Eqs. (17, 18, 19, 20) in 3 dimensions (3-D), 5-D, 7-D, and 10-D. Fig. 3 is a 3-D example of a Gaussian mixture model.

The big difference between an IEC user model and ordinary fitness functions is the implementation of the (a) relative and (b) discrete fitness evaluations of a human user. A human IEC user compares given individuals relatively and, unlike a fitness function, does not give absolute fitness. He or she also cannot give precise fitness values but rather assigns discrete ones, e.g. 1–5 points, every generation, while ordinary fitness functions give continuous values. When the difference between individuals is less than the minimum discrete fitness range, i.e. an evaluation threshold, a human IEC user cannot distinguish the difference, and it becomes fitness noise that must be realized by an IEC user model.

The IDE user simulation in this section randomly chooses either a trial vector or a target vector and leaves it as offspring in the next generation when the difference of their fitness values is less than a certain value to simulate the unavailability of a human IDE user’s comparison; we set the difference threshold as 1/50 of the difference between the maximum and minimum fitness values present in the population for each generation.

$$GMM(x) = \sum_{i=0}^k a_i \exp\left(-\sum_{j=0}^n \frac{(x_{ij} - \mu_{ij})^2}{2\sigma_{ij}^2}\right) \quad (17)$$

Table 2 IEC experiment parameter settings

Population size	20
Search range of parameters	[−5.12, 5.12]
Scale factor $F$	0.9
Crossover rate	0.8
IDE operation	IDE/best/1/bin
Max. search generation, $MAX_{NFC}$	20
Dimensions of Gaussian mixture model, $D$	3, 5, 7, 10
No. of trial runs	50

$$\sigma = \begin{pmatrix} 1.5 & 1.5 & 1.5 & 1.5 & 1.5 & 1.5 & 1.5 & 1.5 & 1.5 & 1.5 \\ 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \end{pmatrix} \quad (18)$$

$$\mu = \begin{pmatrix} -1 & 1.5 & -2 & 2.5 & -1 & 1.5 & -2 & 2.5 & -1 & 1.5 \\ 0 & -2 & 3 & 1 & 0 & -2 & 3 & 1 & 0 & -2 \\ -2.5 & -2 & 1.5 & 3.5 & -2.5 & -2 & 1.5 & 3.5 & -2.5 & -2 \\ -2 & 1 & -1 & 3 & -2 & 1 & -1 & 3 & -2 & 1 \end{pmatrix} \quad (19)$$

$$a_i = (3.1, 3.4, 4.1, 3)^T \quad (20)$$

IDE parameters are set as the Table 2. Population size, 20, is used to reflect a realistic choice for IEC experiments with a real human user.

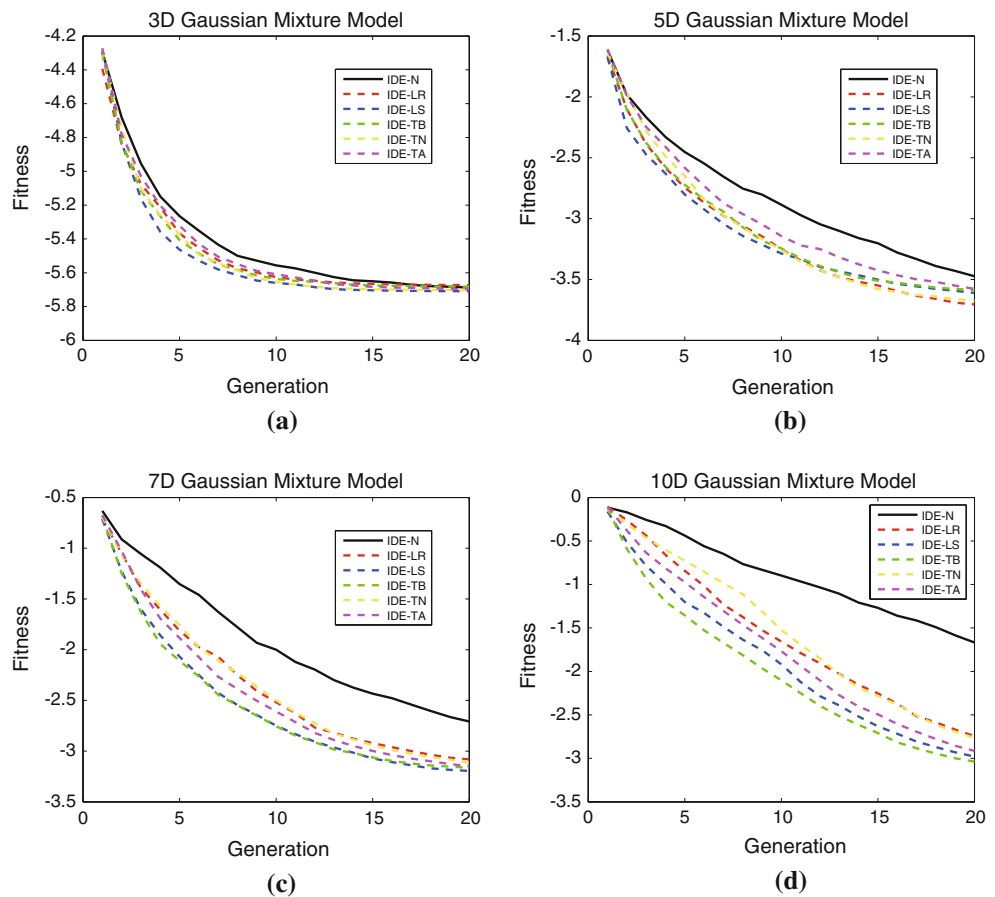
### 5.3 Evaluations of our proposal

Figure 4 shows the average convergence curves of the best fitness values for 50 trial runs of IDE-LS and IDE-LR with their competitors using GMM with 3-D, 5-D, 7-D and 10-D. Table 3 shows the average fitness value at the 20th generation.

From these results it can be seen that, in general, our proposed methods are able to accelerate all Gaussian mixture models, i.e., our proposed methods are significantly better than canonical IDE, with the exception of the IDE-LR method applied to a 3-D model. This indicates that our proposed acceleration methods can be effectively used in some IEC applications. However, for some cases, our proposed methods are not more effective than the previously proposed acceleration method.

To obtain an estimate of the performance of our proposed acceleration methods under real IEC application conditions, we conducted IEC evaluation experiments with 20 individuals per generation. In Table 3, it can be seen that the acceleration performance improves with higher dimensions, i.e. the 10-D, 7-D and 5-D GMM with 20 individuals are more efficient than in the 3-D case. This indicates that our proposed methods increase the diversity of the population in higher dimensions more significantly

**Fig. 4** Average convergence curves of 50 trial runs for **a** 3-D, **b** 5-D, **c** 7-D and **d** 10-D Gaussian mixture model with population size of 20



**Table 3** Average fitness value at 20th generation of 3-D, 5-D, 7-D and 10-D Gaussian mixture models (pseudo-IDE user)

Method	3-D	5-D	7-D	10-D
IDE	-5.59	-3.28	-2.77	-2.76
IDE-TB	-5.59	-3.37	-3.25	-3.14
IDE-TN	-5.76	-3.38	-3.22	-2.74
IDE-TA	-5.58	-3.33	-2.83	-2.98
IDE-LS	<b>-5.75<sup>†,§</sup></b>	<b>-3.39<sup>†,‡,§</sup></b>	<b>-2.78</b>	<b>-3.03<sup>§</sup></b>
IDE-LR	<b>-5.58<sup>§</sup></b>	<b>-3.37<sup>§</sup></b>	<b>-3.27<sup>‡,§</sup></b>	<b>-3.07<sup>‡,§</sup></b>

Bold font, <sup>†</sup>, <sup>‡</sup>, <sup>§</sup> marks values mean IDE-LS or IDE-LR significantly better than IDE, IDE-TB, IDE-TN and IDE-TA, respectively, by Wilcoxon signed-rank tests ( $p < 0.05$ )

than in lower dimensions. When the dimension of the GMM is higher, our proposed methods seem to offer the same performance.

Compared with the IDE-TA method, all of our proposed methods outperform it, except IDE-LS applied in the 7-D modal case. The proposed method IDE-LS and IDE-LR can perform significantly better than IDE-TB in lower dimensional problems (3-D and 5-D) and IDE-TN in higher dimensional problems (7-D and 10-D), respectively. These observations indicate that our proposed

methods may work better than IDE-(TB, TN, TA) when the distribution of individuals in a search space has less diversity, while IDE-(TB, TN, TA) have better performance even in such cases.

## 6 EC experimental evaluations and analysis

### 6.1 Benchmark functions and experimental conditions

We evaluate our proposed methods with 34 benchmark functions. Our proposed methods aim to reduce IDE user fatigue by accelerating IDE search. However, their applicability is not limited to IEC; they can be used for any EC search with fitness functions. Information about some of these functions and their related parameters can be found in literature [3, 19, 25], and [26]. See definitions, search ranges of optimization parameters, and characteristics of the benchmark functions in Table 4.

Some evaluation indexes for comparing our proposed methods applied to EC are average fitness value until convergence is reached in the maximum search generation, Wilcoxon signed-rank test at the maximum search generation and the total time cost statistics and analysis, which

**Table 4** Benchmark functions used in experimental evaluations, where Range is the scale of the parameters,  $n$  is the dimension of the function and C is the function’s characteristics, respectively

No.	Name	Test function	Range	$n$	C
F1	Sphere	$f(x) = \sum_{i=1}^n x_i^2$	[-5.12, 5.12]	3	U-S
F2	Rosenbrock	$f(x) = 100(x_1^2 - x_2)^2 + (1 - x_1)^2$	[-2.048, 2.048]	2	U-N
F3	DeJong-step	$f(x) = \sum_{i=1}^n \lfloor x_i \rfloor$	[-5.12, 5.12]	5	U-S
F4	Quantic and noise	$f(x) = \sum_{i=1}^n ix_i^4 + Gauss(0, 1)$	[-1.28, 1.28]	30	U-S
F5	Foxholes	$f(x) = [0.02 + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^2 (x_i - a_{ij})^6}]^{-1}$	[-65.536, 65.536]	2	M-S
F6	Rastrigin	$f(x) = (10n) + \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i))$	[-5.12, 5.12]	5	M-S
F7	Schwefel 2.26	$f(x) = \sum_{i=1}^n (-x_i \sin(\sqrt{ x_i }))$	[-512, 512]	5	M-S
F8	Griewank	$f(x) = 1 + \sum_{i=1}^n \frac{x_i^2}{4000} - \prod_{i=1}^n \cos(\frac{x_i}{\sqrt{i}})$	[-512, 512]	5	M-N
F9	Schaffer 1	$f(x) = 0.5 + \frac{\sin^2(\sqrt{(x_1^2 + x_2^2)}) - 0.5}{ 1.0 + 0.001(x_1^2 + x_2^2) ^2}$	[-100, 100]	2	M-N
F10	Schaffer 2	$f(x) = (x_1^2 + x_2^2)^{0.25} [\sin^2(50(x_1^2 + x_2^2)^{0.1}) + 1.0]$	[-100, 100]	2	M-S
F11	Schwefel 2.22	$f(x) = \sum_{i=1}^n  x_i  + \prod_{i=1}^n  x_i $	[-10, 10]	30	U-N
F12	Schwefel 1.2	$f(x) = \sum_{i=1}^n (\sum_{j=1}^i x_j)^2$	[-10, 10]	30	U-N
F13	Hartman-3	$f(x) = -\sum_{i=1}^4 c_i \exp[-\sum_{j=1}^3 a_{ij}(x_j - p_{ij})^2]$	[0, 1]	3	M-N
F14	Step	$f(x) = \sum_{i=1}^n (\lfloor x_i + 0.5 \rfloor)^2$	[-10, 10]	30	U-S
F15	Beale	$f(x) = (1.5 - x_1 + x_1 x_2)^2 + (2.25 - x_1 + x_1 x_2^2)^2 + (2.625 - x_1 + x_1 x_2^3)^2$	[-4.5, 4.5]	2	U-N
F16	Kowalik	$f(x) = \sum_{i=1}^{11} [a_i - \frac{x_i(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4}]^2$	[-5, 5]	4	M-N
F17	Carnel-Back	$f(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1 x_2 - 4x_2^2 + 4x_2^4$	[-5, 5]	2	M-N
F18	Branin	$f(x) = (x_2 - \frac{5.1}{4\pi}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 + 10(1 - \frac{1}{8\pi}) \cos(x_1) + 10$	[-5, 10] × [0, 15]	2	M-S
F19	Goldstein-Price	$f(x) = [1 + (x_1 + x_2 + 1)^2 \times (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1 x_2 + 3x_2^2)] \times [30 + (2x_1 - 3x_2)^2 \times (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1 x_2 + 27x_2^2)]$	[-2, 2]	2	M-N
F20	Hartman-6	$f(x) = -\sum_{i=1}^4 c_i \exp[-\sum_{j=1}^6 a_{ij}(x_j - p_{ij})^2]$	[0, 1]	6	M-N
F21	Sh Sphere	same as F1 with Sh	[-100, 100]	50	Sh-U-S
F22	Sh Schwefel 1.2	same as F12 with Sh	[-100, 100]	50	Sh-U-N
F23	Sh Rt Elliptic	$f(x) = \sum_{i=1}^n (10^6)^{\frac{i-1}{n-1}} x_i^2$	[-100, 100]	50	Sh-Rt-U-N
F24	F22 with Noise	same as F22 with noise	[-100, 100]	50	Sh-U-N
F25	Sh Schwefel 2.6 GB	$f(x) = \text{Max}  A_i x - B_i $	[-100, 100]	50	Sh-U-N
F26	Sh Rosenbrock	same as F2 with Sh	[-100, 100]	50	Sh-M-N
F27	Sh Rt Griewank	same as F8 with Sh and Rt	[0, 600]	50	Sh-Rt-M-N
F28	Sh Rt Ackley GB	$f(x) = -20 \exp(-0.2(\frac{1}{n} \sum_{i=1}^n x_i^2)^{\frac{1}{2}}) - \exp(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)) + 20 + e$	[-32, 32]	50	Sh-Rt-M-N
F29	Sh Rastrigin	same as F6 with Sh	[-5, 5]	50	Sh-M-Sep
F30	Sh Rt Rastrigin	same as F6 with Sh and Rt	[-5, 5]	50	Sh-Rt-M-N
F31	Sh Rt Weierstrass	$f(x) = \sum_{i=1}^n (\sum_{k=0}^{k_{max}} [a^k \cos(2\pi b^k (x_i + 0.5))]) - n \sum_{k=0}^{k_{max}} [a^k \cos(2\pi b^k \times 0.5)]$	[-0.5, 0.5]	50	Sh-Rt-M-N
F32	Schwefel 2.13	$f(x) = \sum_{i=1}^n (\sum_{j=1}^n (a_{ij} \sin x_i + b_{ij} \cos x_i))^2$	[-100, 100]	50	Mul-N
F33	Sh Expanded F28F22	$f(x) = F28F22(x_1, x_2, \dots, x_n)$	[-3, 1]	50	Sh-M-N
F34	Sh Rt Scaffer F6	same as F9 with Sh and Rt	[-100, 100]	50	Sh-Rt-M-N

U unimodal, M multimodal, Sh shifted, Rt rotated, GB global on bounds, N non-separable, and S separable



can show us to what degree our proposed methods are time-saving.

EC experimental parameters are set as shown in Table 5. We use differential evolution (DE/best/1/bin) as the optimization method to evaluate the methods. The evaluation is conducted under difficult search conditions; only 100 individuals search 50-D/30-D functions for which the search parameter ranges are expanded, and the other uses 30 individuals.

### 6.2 Evaluations of our proposal

Convergence characteristics of the 34 benchmark functions are shown in Tables 6 and 7. In the tables, Wilcoxon signed-rank tests are applied between our proposed methods with canonical DE and previous acceleration methods

**Table 5** EC experiment parameter settings

Population size	100 (D = 50, D = 30)/30 (others)
Scale factor <i>F</i>	0.9
Crossover rate	0.8
DE operations	DE/best/1/bin
Max. search generation	50
No. of trial runs	50

[22]. As was previously mentioned, for higher dimensional problems (F21–F34), the previous acceleration methods require the # of population size must be  $2D + 1 = 2 \times 50 = 101$ , which exceeds the experimental settings, so our proposed methods can be applied to such higher dimensional problems but the previous methods cannot. This demonstrates one advantage of our proposed methods.

The proposed methods converge faster than the conventional methods at the same generation for both lower and higher dimensional problems except for a few functions. This effect cannot be observed for F3, F5, F7, F8, F23, F28, F31, F32 and F34 functions because there are no significant differences between canonical DE and DE with either of our proposed methods. Our proposed method’s acceleration performance look similar, and if there is any difference, the superiority depends on the task being performed. From the Wilcoxon signed-rank tests comparison of our proposed methods with previous acceleration methods, the performance of our proposed methods is better than one of them in F2, F3, F4, F10, F11, F12, F14, F15, F16, F18, F19 and F20 in lower dimension problems. The DE-LS method’s performance is better than that of DE-LR method in comparison with previous mentioned acceleration methods, i.e. DE-TB, DE-TN and DE-TA. For higher dimensional benchmark functions (F21–F34), our

**Table 6** Average fitness value of F1–F20 benchmark functions

Fun.	DE-N	DE-TB	DE-TN	DE-TA	DE-LS	DE-LR
F1	2.02E–06	9.40E–10	1.65E–11	0.00E+00	<b>2.97E–08</b>	4.28E–03
F2	7.12E–07	5.64E–07	2.66E–07	5.99E–07	<b>2.20E–07</b> <sup>†,‡,§</sup>	<b>3.81E–07</b> <sup>†,§</sup>
F3	–3.00E+01	–3.00E+01	–2.99E+01	–3.00E+01	–2.99E+01	–3.00E+01 <sup>‡,§</sup>
F4	1.33E+01	5.56E–01	1.38E+00	1.86E–01	<b>9.22E–01</b> <sup>‡</sup>	<b>1.67E+00</b>
F5	1.10E+00	1.04E+00	1.61E+00	1.10E+00	1.18E+00	1.58E+00
F6	6.85E+00	3.78E+00	3.53E+00	3.79E+00	<b>4.38E+00</b>	<b>4.92E+00</b>
F7	–1.97E+03	–1.98E+03	–1.94E+03	–1.96E+03	–1.96E+03	–1.95E+03
F8	3.60E–01	3.60E–01	3.60E–01	3.60E–01	3.60E–01	3.60E–01
F9	1.01E–02	8.45E–03	1.13E–02	8.19E–03	<b>9.10E–03</b>	<b>8.11E–03</b>
F10	7.78E–02	3.59E–03	6.27E–03	1.10E–02	<b>1.56E–02</b>	<b>9.25E–03</b> <sup>§</sup>
F11	6.26E+02	2.32E+01	2.54E+01	5.46E+00	<b>7.72E+00</b> <sup>†,‡</sup>	<b>4.60E+01</b>
F12	3.25E+04	1.70E+03	5.20E+03	4.88E+02	<b>3.58E+03</b> <sup>‡</sup>	<b>8.51E+03</b>
F13	6.38E+00	2.70E+00	4.05E+00	1.25E+00	<b>3.85E+00</b>	<b>4.82E+00</b>
F14	1.93E+02	2.64E+01	4.69E+01	9.86E+00	<b>3.69E+01</b> <sup>‡</sup>	<b>6.72E+01</b>
F15	1.78E+01	8.57E+00	1.21E+01	5.46E+00	<b>1.00E+01</b> <sup>‡</sup>	<b>1.26E+01</b>
F16	3.49E–03	1.77E–03	4.16E–03	9.54E–04	<b>1.67E–03</b> <sup>†,‡</sup>	<b>2.58E–03</b>
F17	2.82E–09	5.20E–13	2.00E–13	1.20E–11	<b>2.65E–11</b>	<b>8.15E–06</b>
F18	3.98E–01	3.98E–01	3.98E–01	3.98E–01	<b>3.98E–01</b> <sup>‡</sup>	3.98E–01
F19	3.00E+00	3.00E+00	3.00E+00	3.00E+00	<b>3.00E+00</b> <sup>‡,§</sup>	<b>3.00E+00</b>
F20	–3.26E+00	–3.27E+00	–3.27E+00	–3.26E+00	<b>–3.27E+00</b> <sup>‡</sup>	<b>–3.26E+00</b>

Bold font, <sup>†,‡,§</sup> marks values mean IDE-LS or IDE-LR significantly better than ordinary IDE, IDE-TB, IDE-TN and IDE-TA, respectively, by Wilcoxon signed-rank tests ( $p < 0.05$ )

**Table 7** Average fitness value of F21–F34 benchmark functions

Fun.	DE-N	DE-LS	DE-LR
F21	1.58E+05	<b>1.16E+05</b>	<b>1.39E+05</b>
F22	4.04E+05	<b>3.81E+05</b>	3.98E+05
F23	5.07E+09	4.97E+09	5.07E+09
F24	4.95E+05	<b>4.40E+05</b>	<b>4.51E+05</b>
F25	4.97E+04	<b>4.65E+04</b>	<b>4.91E+04</b>
F26	1.28E+11	<b>7.97E+10</b>	<b>1.07E+11</b>
F27	6.01E+03	<b>4.76E+03</b>	<b>5.84E+03</b>
F28	−1.19E+02	−1.19E+02	−1.19E+02
F29	5.69E+02	<b>4.78E+02</b>	<b>5.52E+02</b>
F30	1.10E+03	<b>9.07E+02</b>	<b>1.06E+03</b>
F31	1.68E+02	1.68E+02	1.68E+02
F32	6.86E+06	6.86E+06	6.84E+06
F33	2.62E+03	<b>1.28E+03</b>	<b>1.47E+03</b>
F34	−2.76E+02	−2.76E+02	−2.76E+02

Bold font values mean DE-LS or DE-LR significantly better than ordinary DE by Wilcoxon signed-rank tests ( $p < 0.05$ )

proposed methods also show the capability to accelerate EC search, except in some cases, and even to problems with shifted and rotated characteristics.

### 6.3 Discussion on acceleration performance

With the exception of F3, F5, F7 and F8, our proposed methods can accelerate all lower dimensional benchmark functions. There were no cases where the proposed methods were significantly poorer than canonical DE. Although our proposed acceleration methods use a dimensionality reduction technique to obtain new elite in a lower dimensional search space, which would seem to imply more efficiency in problems with the characteristic of variable separability (i.e. F1, F4, F6, F10, F14 and F18), the experimental results show that DE-LR and DE-LS also exhibit better performance when the variables are non-separable (i.e. F2, F9, F11–F13, F15–F17, F19 and F20). For uni-modal and multi-modal problems, the experimental results indicate that our proposed methods have the same capability to accelerate these two kinds of problems.

As the global optimum of F3 is on the edge of a searching space, the elite obtained around the global optimum by function approximation may be located outside of the searching range. Our experiment does not use the elite in this case, and the DE using our proposed methods becomes identical to canonical DE. This would explain why there is no significant difference between DE-N, DE-LR and DE-LS.

F7 is made by putting multiple sine curves on a sloped plane that is otherwise similar to F3. The difficulty with this task is that the minimum point, i.e. the global

optimum, is located near the maximum point of the plane and there are local minimums around the global optimum. When local minimums and the global optimum are located close together, the elite obtained from an approximated curve may be located in a gap between these peaks and its fitness may be poor. This may be a reason why the performance of our proposed methods is the same as with canonical DE.

F4 is a quartic function with Gaussian noise and has its global optimum in the center of the searching space. Although it is a multi-modal function due to the quartic function and fluctuations due to noise, the influence of the noise is relatively small and its whole shape is close to a quadratic function such as in F1. Elite from an approximated function using individuals that fluctuate slightly should be located in the center, and the performance of DE-LR and DE-LS should be better, as with F1, while that of canonical DE is negatively influenced by the fluctuations. This may explain the good performance of our proposed method with F4.

For higher dimensional problems (F21–F34), our proposed methods are also effective except F23, F28, F31, F32 and F34; most of them have complex fitness landscapes with a rotation characteristic. As rotated fitness landscape changes the real location of global optimum dramatically, our projected one-dimensional landscape can not use other dimensional search space information collaboratively. Obtained elite are not correct in each lower dimension. Maybe this is the reason why our proposed methods are ineffective in those benchmark tasks.

Except in special cases, the performance of DE-LR and DE-LS are better than that of DE-TN. This demonstrates that the original search space landscape fitted by those individuals with the lowest distance from the best individual is not accurate as a regression of the original search space; i.e. individuals near the best individual are not necessarily in regions of better fit in the original search space.

### 6.4 Discussion on computational complexity

Reducing computational cost is the greatest feature offered by our methods. This method of approximating landscapes and using the peak of an approximated function belongs to the same category as [22], but we extend that work by introducing projection onto a 1-D space, synthesizing an elite from the elites on multiple 1-D functions, and thus further reducing computational cost. Let's examine the efficiency of the methods by calculating the time cost (see Table 8).

We calculate the run times of 50 trials of DE-N, DE-LR, DE-LS, DE-TB, DE-TN and DE-TA, and obtain the average running time cost of one computation. The system

**Table 8** Average cost time of each method for one time running (Unit: ms)

F	N	TB	TN	TA	LS	LR
F1	6.26	24.69	24.37	24.07	9.06	8.95
F2	9.58	24.05	23.02	23.45	12.2	12.19
F3	10.52	36.87	39.27	36.67	12.29	12.29
F4	25.00	1,761.25	2,754.58	2,754.49	28.12	28.13
F5	20.63	35.10	34.18	34.47	23.12	23.13
F6	15.31	42.81	40	38.33	17.71	17.92
F7	16.88	44.27	41.15	39.37	19.79	19.69
F8	17.39	44.59	41.47	39.47	19.89	20.00
F9	12.39	27.08	26.25	26.14	15.00	15.21
F10	13.02	27.71	26.98	26.67	15.83	15.94
F11	25.52	1,748.96	2,709.58	2,697.81	28.12	28.55
F12	45.63	1,762.19	2,740.21	2,725.31	48.23	48.01
F13	13.02	35.41	36.25	34.69	15.63	15.53
F14	50.11	2,292.91	2,739.48	2,748.33	44.17	43.85
F15	10.00	24.58	23.45	23.64	12.19	12.92
F16	13.23	35.84	36.97	35.52	15.74	16.04
F17	9.89	24.27	24.89	23.85	12.71	12.61
F18	11.26	25.94	23.85	24.17	14.06	14.16
F19	9.79	24.37	23.65	23.75	12.61	12.71
F20	9.06	41.98	45.11	42.19	11.56	1.67

*F* means function. *N* means canonical DE. LR, LS, TB, TN and TA mean the acceleration methods of DE-LR, DE-LS, DE-TB, DE-TN and DE-TA, respectively

used to run the experiment is powered by a Core2 Duo 2GHz CPU with 1GB RAM running Windows XP (SP2).

The table shows that the time cost of our proposed methods is more than canonical DE and less than the previous acceleration methods. The time cost of our proposed method is almost the same for one benchmark function. The three previous acceleration methods are costly for one benchmark function. We can conclude that our proposed methods save the time cost of optimization when compared to the previous acceleration methods.

To further evaluate the significance of our proposed methods as compared with the previous methods, we conduct a time cost variance analysis with the time cost data sample from our tests. Before calculating the *F* distribution value of the multiple groups' sample time cost data for significance evaluation, we should check whether the sample data fit a normal distribution in theory. For a large data set (data >30 samples), it must fit a normal distribution in accordance with the Central Limit Theorem. We use the Jarque–Bera test to check each time cost data set for goodness-of-fit to a normal distribution, and for abnormal data we smooth it for fitting to a normal distribution. We calculate the *F* value for each of the lower dimensional benchmark functions (F1–F20) in Table 9 with  $r = 5$ ,  $n = 250$ .  $F_{0.05}(r - 1, n - r) = F_{0.05}(4, 245)$ ,

i.e.,  $F_{0.05}(4, \infty) = 2.37$ . From Table 9, it can be seen that all the *F* values are greater than 2.37, indicating that our proposed method's time cost means differ significantly from those of the previous methods.

From Tables 8 and 9, we can conclude that the proposed methods can significantly save time over previous methods. A comparison of DE-(LR and LS) with DE-(TB, TN and TA) shows that our proposed acceleration methods seem more beneficial than previous ones. However, before the regression and fitting process, it is necessary to select the sample data. In the selection process, there are more searching and sorting operations, and these take as much as or even more time than the regression and fitting processing. This shows how essential selection of sample data is when accelerating EC from a landscape obtained from its search space. This, improving the performance of data selection, will be a subject of our future research.

## 7 Discussion of general issues related to our proposal

### 7.1 Regression function selection for landscape approximation

Approximation function selection is a critical issue in the approximation of the IEC/EC landscape. From a practical viewpoint, we should select an appropriate approximation function after analyzing the characteristics of the IEC/EC landscape. In this study, we reviewed related regression methods used to approximate IEC/EC landscapes in Sect. 3, and selected binomial Lagrange interpolation and linear least squares approximation as the regression methods for our study.

The regression search space approximated by binomial Lagrange interpolation and linear least squares approximation are nonlinear and linear, respectively. The characteristic of the nonlinear space is that it is ruleless, which is a feature common to most IEC/EC search spaces. Binomial Lagrange interpolation is a method that can be used to obtain nonlinear landscape information about the IEC/EC search space and we can find new elite in this nonlinear regression space. Maybe there is a rule between the new elite and the global optimum in the regression search space, and we can obtain it by chance. The linear function least squares method approximates some linear relationship between each individual and its fitness in the local search space. If the search space of some IEC/EC application is linear, we can obtain the IEC/EC search space's linear relationship from the linear function approximation. We can also obtain the global optimum from the new elite in the regression linear space.

How to obtain the approximated IEC/EC search space is a promising research direction. In our study, we tried to use

**Table 9** Confidence interval and F value for the 20 benchmark functions

Function	F value	TA-LR	TA-LS	TN-LR	TN-LS	TB-LR	TB-LS
F1	195.16	[24.71, 25.90]	[24.93, 26.12]	[11.48, 12.67]	[11.69, 12.88]	[0.55, 1.74]	[0.76, 1.95]
F2	107.80	[24.28, 25.52]	[24.17, 25.41]	[10.42, 11.66]	[10.31, 11.55]	[0.21, 1.46]	[0.10, 1.34]
F3	673.36	[26.96, 28.41]	[26.76, 28.21]	[16.98, 18.43]	[16.78, 18.23]	[1.86, 3.31]	[1.66, 3.11]
F4	198,746.11	[141.39, 142.5]	[140.5, 141.73]	[197.02, 198.19]	[196.19, 197.36]	[111.39, 112.56]	[110.56, 111.73]
F5	171.23	[24.89, 25.93]	[24.48, 25.52]	[11.04, 12.07]	[10.63, 11.67]	[0.82, 1.86]	[0.42, 1.45]
F6	568.36	[24.87, 26.16]	[24.77, 26.06]	[15.60, 16.89]	[15.50, 16.79]	[4.66, 5.95]	[4.56, 5.85]
F7	371.44	[26.01, 27.13]	[25.79, 26.92]	[16.63, 17.75]	[16.41, 17.54]	[6.52, 7.64]	[6.31, 7.43]
F8	381.21	[26.69, 27.70]	[26.48, 27.48]	[16.25, 17.26]	[16.04, 17.04]	[6.37, 7.38]	[6.16, 7.16]
F9	212.91	[24.73, 25.89]	[24.72, 25.88]	[10.68, 11.83]	[10.67, 11.83]	[0.46, 1.61]	[0.45, 1.61]
F10	121.83	[25.40, 26.67]	[25.30, 26.57]	[10.92, 12.20]	[10.82, 12.10]	[0.09, 1.37]	[0.00, 1.27]
F11	484,763.35	[142.97, 144.12]	[141.51, 142.66]	[197.85, 199.00]	[196.39, 197.54]	[111.19, 112.34]	[109.73, 110.88]
F12	364,138.63	[143.29, 144.81]	[141.74, 143.25]	[197.56, 199.08]	[196.01, 197.53]	[111.52, 113.04]	[109.97, 111.49]
F13	525.10	[26.55, 28.02]	[26.46, 27.93]	[15.19, 16.66]	[15.11, 16.58]	[1.23, 2.70]	[1.15, 2.62]
F14	143,719.42	[146.15, 149.68]	[144.28, 147.80]	[196.15, 199.68]	[194.28, 197.80]	[110.52, 114.04]	[108.65, 112.17]
F15	89.73	[2.50, 25.69]	[24.59, 25.79]	[10.44, 11.64]	[10.54, 11.73]	[0.55, 1.75]	[0.64, 1.84]
F16	591.96	[27.66, 29.00]	[27.46, 28.80]	[15.46, 16.81]	[15.26, 16.61]	[1.19, 2.53]	[0.99, 2.33]
F17	94.37	[24.68, 25.93]	[24.38, 25.63]	[10.83, 12.08]	[10.53, 11.78]	[0.62, 1.87]	[0.32, 1.57]
F18	63.00	[23.59, 24.97]	[23.48, 24.86]	[9.72, 11.11]	[9.61, 10.99]	[0.36, 1.75]	[0.25, 1.63]
F19	118.13	[24.32, 25.69]	[24.12, 25.49]	[10.66, 12.03]	[10.46, 11.83]	[0.66, 2.03]	[0.46, 1.83]
F20	1,069.79	[27.65, 28.81]	[27.85, 29.01]	[22.55, 23.71]	[22.75, 23.91]	[1.81, 2.98]	[2.01, 3.18]

All the intervals are positive, indicating that the previous acceleration methods, i.e. DE-(TB, TN and TA) have significantly greater time cost than our proposed acceleration methods, i.e. DE-(LR and LS)

nonlinear and linear curve expressions to approximate the IEC/EC search space, to obtain the approximation relationship between the approximated and the original IEC/EC search space, and to find the global optimum from the new elite by chance. However, the approximation process influences the performance of the IEC/EC algorithms, so using an efficient approximation method is another topic to be considered when designing accelerated IEC/EC algorithms. We should use the lowest possible degree of interpolation and approximation functions in the lowest possible dimensional spaces to reduce the time cost required for the entire IEC/EC algorithm.

## 7.2 Regression model establishment and usage

The experimental results in this study show that both linear (DE-LS) models and nonlinear (DE-LR) models can be used to efficiently accelerate IEC/EC convergence. However, sign tests indicate that the methods are mostly effective in the initial generations [10], and an alternative solution is to use this kind of acceleration method to speed IEC/EC convergence only in those early generations.

Constructing and selecting different models to better approximate the IEC/EC search space is a worthwhile topic for further research. To achieve optimal performance, we should choose approximation methods that are suited to the

characteristics of the search space. For regression in the discrete domain, a more powerful tool than continuous domain Lagrange interpolation and least squares approximation is required. To find how best to approximate the search space in the discrete domain is a valuable research topic.

Another point deserving special attention is the method used for dimensionality reduction. The objective of the dimensionality reduction is to simplify the computation of the search space regression by performing it in a lower dimension. The dimensionality reduction loses much search space information, however, such as the relationship between the parameters in non-separable problems. How we can reduce the search space dimension as much as possible while simultaneously preserving the necessary search space information for the next generation of the search is a challenging problem for further research into the dimensionality reduction method.

The new elite selection method is also a key step in the acceleration process. The elite selection method is determined by the different search space regression methods. They can be categorized as linear and nonlinear models, corresponding to the classification method of the regression model. A new elite selection method should be determined by analyzing the characteristics of the regression model.

### 7.3 Data sampling technique

It is necessary to sample the data to obtain the original search space information which is used in the approximation processes. The data sampling technique determines the quality of the approximation model, indirectly influencing the model usage and obtained elite. Several data sampling methods have been proposed. These include best data strategy, distance nearest data strategy, whole data strategy and random data strategy. For comparison of those sample studies, we discuss the advantages and disadvantages of the methods as follows.

First, obtaining suitable data for the approximation of the IEC/EC landscape requires time in searching and sorting operations. With a view to maximizing performance and saving time, we need to select an easy way to obtain the sample data. Secondly, from the experimental results, the distance nearest data strategy is more costly than any of the other sampling data strategies. The acceleration performance of the approximation model combined with distance nearest data is not as good as the others. So with a view to practical applications, the distance nearest data strategy exhibits low performance. Third, our proposed acceleration methods that use the best data strategy to approximate the model in the lower search space perform better than the previous acceleration methods in some cases. However, if we select a data sampling strategy which is even more time efficient and does not require searching and sorting, such as the random data strategy, the performance of our proposed methods may be further improved. This is a direction for further research.

### 7.4 Limitations

Additional time is required to approximate the IEC/EC search space when using our proposed acceleration methods, so the IEC/EC algorithm's computational complexity is increased. The performance issue is the biggest problem for the application of our acceleration methods. Although our proposed methods exhibit performance that is better than the previously proposed acceleration methods, performance remains a critical issue when looking to apply the method. When this acceleration is used in a concrete application, we must consider the issue of performance and find a balance between algorithm computational complexity and the speed of convergence.

Population diversity is the key consideration for obtaining the final global optimum in IEC/EC. Acceleration methods can lead to a premature convergence on a local optimum in some special landscape conditions. When a nearly identical elite is inserted generation after generation, our proposed method can lead to premature convergence. The issue of elite insertion strategy selection is a topic of

considerable importance when applying our proposed acceleration methods to concrete IEC/EC applications.

## 8 Conclusion and future work

We proposed IEC/EC convergence be accelerated by using a dimensionality reduction technique in approximating the fitness landscape. The novel feature in these acceleration methods was to use elite synthesized from elite points found in lower dimensional spaces. The main contribution of this study was utilizing the notion of dimensionality reduction for approximating the fitness landscape to accelerate IEC/EC searches. Our experimental evaluations with a Gaussian mixture model and 34 benchmark functions showed that the proposed methods can accelerate IEC and EC searches with lower computational cost. The acceleration performance can be obtained in both lower and higher dimensional problems, especially when the landscape of the tasks takes on a roughly big valley structure. We also analyzed the relationship between the performance of the proposed methods and the landscape shapes.

Theoretical research is but one part of this scientific research; the ultimate objective of our research is to apply this novel method and technique to actual applications with societal benefit. We conducted simulation evaluations to compare characteristics of several methods with multiple different initializations under the same experimental conditions. In the next step, we should quantitatively evaluate IEC user fatigue and collect basic experimental data together with the convergence experiments' results in this paper. After that, we need to conduct human subjective tests with human IEC users, evaluate user fatigue and acceleration performance synthetically, and conclude the evaluation of the proposed methods. It is important to obtain conclusions through its application and thus find new directions by which we can improve our work. We expect these developments will occur on topics such as the selection of the linear or nonlinear model, the actual interpolation or approximation methods used and the expression of the regression space. We will continue conducting research on those topics in the future.

**Acknowledgments** This work was supported in part by Grant-in-Aid for Scientific Research (23500279). Yan Pei would like to thank Yoshida Scholarship Foundation for its support of his doctoral research.

## References

1. Brest J, Greiner S, Boskovic B, Mernik M, Umer V (2008) Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems. *IEEE Trans Evol Comput* 10(6):107–125

2. Carretero JA, Nahon MA (2005) Solving minimum distance problems with convex or concave bodies using combinatorial global optimization algorithms. *IEEE Trans Syst Man Cybern Part B Cybern* 35(6):1144–1155
3. De Jong KA (1975) An analysis of the behavior of a class of genetic adaptive system, Ph.D Dissertation. University of Michigan, USA
4. Gong W, Cai Z, Ling CX, Li H (2011) Enhanced differential evolution with adaptive strategies for numerical optimization. *IEEE Trans Syst Man Cybern* 41(2):397–413
5. Hakimi-Asiabara M, Ghodsypoura SH, Kerachianb R (2009) Multi-objective genetic local search algorithm using Kohonen's neural map. *Comput Ind Eng* 56(4):1566–1576
6. Liu LM, Wang NP, Li FC (2009) Study on convergence of self-adaptive and multi-population composite genetic algorithm. In: *International Conference on machine learning and cybernetics*, vol 5. Baoding, China, pp 2680–2685, 12–15 Jul 2009
7. Lahiri A, Chakravorti S (2004) Electro-spacer contour optimization by ANN aided genetic algorithm. *IEEE Trans Dielectr Electr Insul* 11(6):964–975
8. Mantawy AH, Abdel-Magid YL, Selim SZ (1999) Integrating genetic algorithms, Tabu Search, and simulated annealing for the unit commitment problem. *IEEE Trans Power Syst* 14(3):829–836
9. Nakano Y, Takagi H (2009) Influence of quantization noise in fitness on the performance of interactive PSO. In: *IEEE congress on evolutionary computation (CEC2009)*. Trondheim, Norway, pp 2146–2422
10. Pei Y, Takagi H (2011) Accelerating evolutionary computation with elite obtained in projected one-dimensional spaces. In: *Proceedings of the 5th international conference on genetic and evolutionary computing (ICGEC2011)*. Kimmen Taiwan, Xiamen China, pp 89–92, 29 Aug, 1 Sept 2011
11. Pei Y, Takagi H (2011) A novel traveling salesman problem solution by accelerated evolutionary computation with approximated cost matrix in an industrial application. In: *Proceedings of the 3rd international conference on soft computing and pattern recognition (SoCPaR2011)*. Dalian, China, pp 39–44, 14–16 Oct 2011
12. Pei Y, Takagi H (2011) A survey on accelerating evolutionary computation approaches. In: *Proceedings of the 3rd international conference on soft computing and pattern recognition (SoC-PaR2011)*. Dalian, China, pp 201–206, 14–16 Oct 2011
13. Pei Y, Takagi H (2011) Comparative evaluations of evolutionary computation with elite obtained in reduced dimensional spaces. In: *Proceedings of the 3rd international conference on intelligent networking and collaborative systems (INCoS2011)*. Fukuoka, Japan, pp 35–40, 30 Nov, 2 Dec 2011
14. Pei Y, Takagi H (2013) Triple and quadruple comparison-based interactive differential evolution and differential evolution. In: *Proceedings of the foundations of genetic algorithms XII (FOGA) 2013 workshop*. Adelaide, Australia, pp 173–182, 16–20 Jan 2013)
15. Pei Y, Takagi H (2012) Fourier analysis of the fitness landscape for evolutionary search acceleration. In: *Proceedings of the 2012 IEEE congress on evolutionary computation (IEEE CEC 2012)*. Brisbane, Australia, pp 2934–2940, June 2012)
16. Pei Y, Takagi H (2012) Comparative study on fitness landscape approximation with Fourier transform. In: *Proceedings of the sixth international conference on genetic and evolutionary computation (ICGEC2012)*. Kitakyushu, Japan, pp 400–403, 25–28 Aug 2012
17. Price K, Storn R, Lampinen J (2005) *Differential evolution: a practical approach to global optimization*. Springer, Berlin
18. Qin AK, Huang VL, Suganthan PN (2009) Differential evolution algorithm with strategy adaptation for global numerical optimization. *IEEE Trans Evol Comput* 13(2):398–417
19. Suganthan P, Hansen N, Liang J, Deb K, Chen Y, Auger A, Tiwari S (2005) Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization, Technical Report. Nanyang Technological University, Singapore, <http://www.ntu.edu.sg/home/EPNSugan>
20. Qin AK, Huang VL, Suganthan PN (2009) JADE: adaptive differential evolution with external archive. *IEEE Trans Evol Comput* 13(5):945–958
21. Takagi H, Pallez D (2009) Paired comparison-based interactive differential evolution. In: *The first world congress on nature and biologically inspired computing (NaBIC2009)*. Coimbatore, India, pp 375–480, 9–11 Dec 2009
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. Takagi H (2001) Interactive evolutionary computation: fusion of the capabilities of EC optimization and human evaluation. *Proc IEEE* 89(9):1275–1296
24. Wang Y, Cai ZX, Guo GQ, Zhou YR (2007) Multiobjective optimization and hybrid evolutionary algorithm to solve constrained optimization problems. *IEEE Trans Syst Man Cybern Part B Cybern* 37(3):560–575
25. Whitley D, Mathias K, Rana S, Dzubera J (1995) Building better test functions. In: *Proceedings of the 6th international conference on genetic algorithms (ICGA95)*. Pittsburgh, PA, pp 239–246
26. Yao X, Liu Y, Lin G (1999) Evolutionary programming made faster. *IEEE Trans Evol Comput* 3(2):82–102
27. Zhou ZZ, Ong YS, Nair pB, Keane AJ, Lum KY (2007) Combining global and local surrogate models to accelerate evolutionary optimization. *IEEE Trans Syst Man Cybern Part C Appl Rev* 37(1):66–76