Journal of Computational Science 30 (2019) 11–27

Contents lists available at ScienceDirect

Journal of Computational Science

journal homepage: www.elsevier.com/locate/jocs

An adaptive encoding learning for artificial bee colony algorithms

Jingyuan Yang[a,1], Qiaoyong Jiang[a,1], Lei Wang[a,*], Shuai Liu[b,*], Yu-Dong Zhang[c,*], Wei Li[a], Bin Wang[a]

[a] The Faculty of Computer Science and Engineering, Xi'an University of Technology, Xi'an 710048, China
[b] The Faculty of Computer Science, Inner Mongolia University, Hohhot 010012, China
[c] The Department of Informatics, University of Leicester, United Kingdom

ARTICLE INFO

Article history:
Received 13 September 2018
Received in revised form 22 October 2018
Accepted 1 November 2018
Available online 10 November 2018

Keywords:
Artificial bee colony
Variable linkages
Eigen coordinate system
Adaptive encoding learning
Adaptive selection mechanism

ABSTRACT

Recently, the improvements of artificial bee colony (ABCs) have attracted increasing interest in the studies of single optimization problems. However, most existing work of ABC aims to design new solution search equations and is still challenged when solving optimization problems with variable linkages. To overcome this limit, an adaptive encoding learning for ABCs (AEL + ABCs) is proposed in this paper. In AEL + ABCs, the solution search equations are encoded in both natural coordinate system and eigen coordinate system guided by covariance matrix learning. The purpose of the former is to maintain the diversity of population, while the latter aims at directing the evolution of population toward the promising directions by identifying the properties of fitness landscape. In addition, an adaptive selection mechanism is used to achieve a good tradeoff between convergence and diversity. For the comparison purposes, the proposed AEL strategy is applied to eight ABCs and their performance is tested on 30 CEC2014 benchmark functions. Experiment results show that the proposed AEL + ABCs can significantly improve the performance of the state-of-the-art ABCs in the majority of the benchmark functions.

1. Introduction

Artificial bee colony (ABC) algorithm, proposed by Karaboga and Basturk [1] in 2007, has become a hotpot in the field of swarm intelligence computation. Similar to other swarm intelligence algorithms (SIAs), ABC is a population-based optimization algorithm. Due to its simple structure, ease of implementation and outstanding performance, ABC has been widely applied to various optimization problems, such as economic dispatch problem [2], neural network training [3] and clustering [4], and so on. More work can be referred to the literature reviews of ABC in [5–7]. From the literature reviews, it can be seen that ABC has been widely investigated in the past ten years, and a lot of variations have been proposed to further improve its performance. In the following, a brief survey on improved ABCs can be mainly divided into three aspects.

- Introduction of new solution search equations: The solution search equation based on one dimensional variation is the most prominent feature of ABC, which determines the search behavior of ABC. However, the solution search equation used in ABC is good at exploration but poor at exploitation, which may make ABC suffer from slow convergence. In order to overcome this limit, many new solution search equations have been proposed. For example, Zhu and Kwong proposed a gbest-guided ABC (GABC) by introducing the current best solution into the solution search equation [8]. The experiment results show that GABC performs significantly better than ABC on most of cases. Inspired by differential evolution [9], Gao et al. proposed an enhanced ABC (EABC) [10], in which two new solution search equations were developed to generate candidate solutions in the employed bee phase and onlookers phase, respectively. Gao et al. further proposed a modified ABC (MABC) [11], in which a modified solution search equation based on the best solution of the previous iteration to improve the exploitation. Karaboga et al. proposed a new solution search equation for the onlooker bee (qABC), which exploited the neighbors information of the current best solution [12]. Cui et al. proposed a ranking-based solution search equation (RABC), in which the parent food sources were selected according to their rankings [13]. Zhang et al. proposed an improved solution search equation with one-position inheritance mechanism

* Corresponding authors.
E-mail addresses: wangleeei@163.com (L. Wang), cs-liushuai@imu.edu.cn (S. Liu), yudongzhang@ieee.org (Y.-D. Zhang).
[1] The first two authors contributed equally to this work and should be considered co-first authors.

https://doi.org/10.1016/j.jocs.2018.11.001
1877-7503/

(OPIABC) to accelerate the convergence speed of ABC [14]. Li et al. proposed a new ABC variant with memory algorithm (ABCM), in which the memory mechanism was used to guide the solution search equation [15]. Very recently, Pan et al. proposed a hybrid ABC (HABC) [16], in which a new solution search equation was designed based on the best-of-random mutation method. Kumar and Mishra proposed a co-variance guided solution search equation for the onlooker bee (CABC), which was a unification of ABC and statistical covariance [17]. In addition, some techniques with multiple solution search equations were introduced into ABC to enhance its performance, such as ABCVSS [18], MuABC [19], and ABCG [20].

- Hybridization of ABC and auxiliary techniques: Hybridization with auxiliary techniques is a promising way to improve the performance of ABC. For example, Gao et al. improved the performance of ABC by integrating the chaotic map [21], Rosenbrock's rotational direction method [22], orthogonal learning [23], and information learning [24]. In addition, Cui et al. proposed an adaptive approach for the population size for ABC [25], and a ranking-based adaptive ABC for global numerical optimization [26].
- Hybridization of ABC and other meta-heuristic algorithms: Integrating ABC with other optimization algorithms could effectively improve the performance of ABC. For example, Xiang et al. implemented ABC and DE for all solutions [27]. Chen et al. proposed a hybrid ABC with teaching-learning-based optimization (TLBO) [28,29], in which the proposed method adopted three hybrid search phases: teaching-based employed bee phase, learning-based on looker bee phase, and generalized oppositional scout bee phase to improve the efficiency of search. In addition, ABC has been combined with other popular algorithms, such as genetic algorithm (GA) [30] and particle swarm optimization (PSO) [31].

Although most ABCs have achieved good results for most single numerical optimization problems (SNOPs) without variable linkages, they perform poorly in solving SNOPs with variable linkages. The cause of the poor performance is the rotational variability of solution search equations used in ABCs. That is, the performance of ABCs depends mainly on the choice of coordinate system. Unfortunately, existing ABCs are all implemented in the natural coordinate system. This coordinate system is suitable for solving SNOPs without variable linkages, but it is inappropriate for solving SNOPs with variable linkages. The reason is that the interactions among variables have not been correctly identified under the natural coordinate system. As a result, ABCs often lose their original advantages and effectiveness when solving problems with strong linkage among variables.

On the basis of above considerations, a new class of ABCs with adaptive encoding learning, denoted as AEL + ABCs hereafter, are proposed in this paper. In AEL + ABCs, covariance matrix learning strategy is firstly used to extract the distribution information of current population, with the aim of establishing the eigen coordinate system. Subsequently, an adaptive encoding learning is built for ABCs by implementing the solution search equations in both eigen coordinate system and natural coordinate system. Moreover, each coordinate system is selected based on a probability vector which determines the selection ratio, and the probability vector is adaptively adjusted according to the success rate of each coordinate system in a learning period. Finally, extensive experiments are carried on the 30 test functions with 30, 50 and 100 dimensions from CEC2014 to verify the effectiveness of AEL + ABCs.

The main contributions of this paper are summarized as follows:

- AEL provides a simple yet efficient adaptive encoding learning for the solution search equation of ABCs, in which one is encoded in the eigen coordinate system and the other one is encoded in the natural coordinate system. The former aims at loosing the interactions among the variables by identifying the properties of the fitness landscape, which is useful for directing the evolution of the population toward the promising directions. The purpose of the latter is to maintain the superiority of ABCs in the natural coordinate system.
- An adaptive selection mechanism based on the success rate of each coordinate system in a learning period is designed to balance the exploration and exploitation abilities of AEL + ABCs.
- The results of our experimental studies have shown that AEL is an effective method to enhance the performance of existing ABCs.

The remainder of this paper is organized as follows. In Section 2, the basic ABC is briefly introduced. In Section 3, the proposed AEL + ABCs are elaborated. In Section 4, experimental results and an analysis of our comparative study are presented. Finally, Section 5 concludes the paper and suggests possible directions for future study.

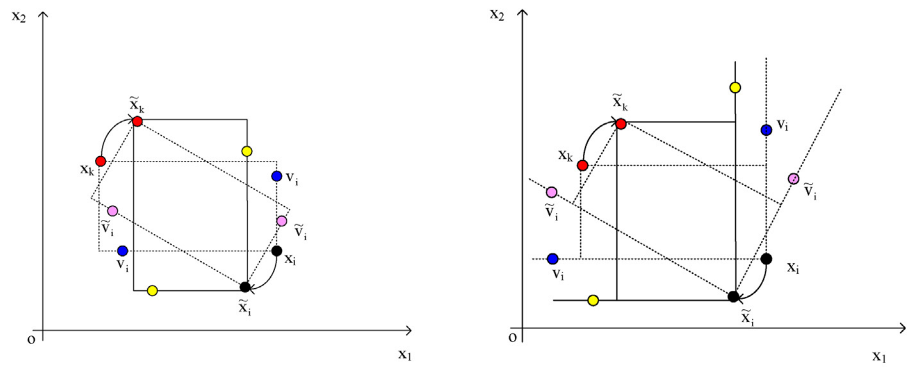
Algorithm 1. Artificial bee colony

Step 1) Initialization:
  Step 1.1) Randomly initialize a population of SN individuals by Eq. (1).
  Step 1.2) Evaluate the objective function value of each individual by Eq. (2).
  Step 1.3) Set $FES = SN$.
Step 2) The employed bee phase:
 For $i = 1, 2, \ldots, SN$ **do**
  Step 2.1) Generate the candidate solution $\mathbf{v}_i$ by Eq. (3).
  Step 2.2) Evaluate $\mathbf{v}_i$ and set $FES = FES + 1$.
  Step 2.3) **If** $f(\mathbf{v}_i) < f(\mathbf{x}_i)$, set $\mathbf{v}_i = \mathbf{x}_i$, $trial_i = 0$, **otherwise** set $trial_i = trial_i + 1$.
Step 3) Calculate the probability values p_i by Eq. (4), and set $t = 0$, $i = 1$.
Step 4) The onlookers phase:
 While $t \le SN$ **do**
  Step 4.1) **If** $rand(0, 1) < p_i$
   Step 4.1.1) Generate the candidate solution $\mathbf{v}_i$ by Eq. (3).
   Step 4.1.2) Evaluate $\mathbf{v}_i$ and set $FES = FES + 1$.
   Step 4.1.3) **If** $f(\mathbf{v}_i) < f(\mathbf{x}_i)$, set $\mathbf{v}_i = \mathbf{x}_i$, $trial_i = 0$, **otherwise** set $trial_i = trial_i + 1$.
   Step 4.1.4) Set $t = t + 1$.
  Step 4.2) Set $i = i + 1$, **If** $i = SN$, set $i = 1$.
Step 5) The scout phase:
   If $max(trial_i) > limit$, replace $\mathbf{x}_i$ with a new solution produced by Eq. (1) and set $FES = FES + 1$.
Step 6) **If** $FES > MaxFEs$, stop and output the best solution, **otherwise** go to **Step 2)**.

2. Artificial bee colony

The ABC is a swarm based meta-heuristic algorithm which is inspired by the intelligent foraging behavior of honey bees. In ABC, the colony of artificial bees contains three different groups of bees: employed bees, onlooker bees, and scout bees. Half of the colony consists of the employed bees, and the remaining half is considered as the onlooker bees. The employed bees are responsible for searching food source and sharing the information about food source with onlooker bees, and the onlooker bees choose good food source from those found by employed bees to further search the foods. When the quality of the food source is not improved within a predetermined number of cycles, the employed bee becomes a scout bee and starts to search for a new food source. In ABC, the position of a food source corresponds to a possible solution to the optimization problem, and the nectar amount of a food source represents the fitness (quality) of the associated solution. The procedure of ABC is shown in algorithm 1. According to the procedure, the initialization stage together with the three search stages are described as follows.

At the initialization stage, ABC creates a randomly distribution population $\mathbf{P}$ of SN solutions (food source positions), where SN is the size of employed bees. Each solution $\mathbf{x}_i = \{x_{i,1}, x_{i,2}, \ldots, x_{i,D}\}$



(a) (b)

Fig. 1. The search equation of ABC and its rotation.

is generated randomly within the search space by Eq. (1) and is evaluated based on the fitness value fit_i by Eq. (2).

$$x_{i,j} = L_j + rand_{i,j}(0,1)\cdot(U_j - L_j) \tag{1}$$

$$fit_i = \begin{cases} \dfrac{1}{1+|f(\mathbf{x}_i)|}, \; if\; f(\mathbf{x}_i) \geq 0, \\ 1+|f(\mathbf{x}_i)|, \; otherwise, \end{cases} \tag{2}$$

where $i \in \{1, 2, \ldots, SN\}$, $j \in \{1, 2, \ldots, D\}$, D is the problem dimension; $x_{i,j}$ is the jth decision variable of $\mathbf{x}_i$, $rand_{i,j}(0, 1)$ is a uniformly distributed random number between 0 and 1, and L_j and U_j are the lower and upper bounds of the jth decision variable, respectively.

At the employed bee stage, each employed bee $\mathbf{x}_i$ is used to produce a candidate solution $\mathbf{v}_i$ using the following search equation.

$$v_{i,j} = \begin{cases} x_{i,j} + \phi_{i,j1}\cdot(x_{i,j} - x_{k,j}), \; ifj = j_1, \\ x_{i,j}, \; otherwise, \end{cases} \tag{3}$$

where $x_{k,j}$ and $v_{i,j}$ denote the jth decision variable of $\mathbf{x}_k$ and $\mathbf{v}_i$ $(k \neq i)$, respectively; $\phi_{i,j1} \in [-1, 1]$ is a random number, and $j_1 \in [1, D]$ is a randomly index. Then, a greedy selection is done between $\mathbf{x}_i$ and $\mathbf{v}_i$.

At the onlooker bee stage, each onlooker bee chooses a position produced by the employed bees to visit, which is based on the probability value p_i of the solution selection as follows.

$$p_i = \frac{fit_i}{\sum_{i=1}^{SN} fit_i} \tag{4}$$

After a solution is selected, an onlooker bee generates a modified solution in the same way as shown in Eq. (3), and the better one will survive to the next generation.

At the scout stage, a predetermined parameter *limit* is introduced to decide if a food source is abandoned. If a food source $\mathbf{x}_i$ cannot be improved through *limit* cycles, then the food source is abandoned and the associated employed bee becomes a scout. After that, the scout randomly creates a new food source by the Eq. (1).

3. Proposed AEL + ABCs

3.1. Motivation

According to the above literature survey, we observed that most of advanced ABCs focused mainly on modifies of the solution search equation but rarely involved the encoding problem of search equation in differential coordinate systems. In general, the problem encoding that the choice of the representation of the optimization problem is extremely important. A proper representation can render any search problem trivial and vice versa. For existing ABCs, the search problem is represented without exception on the natural coordinate system. A lot of numerical experiment results have demonstrated that this representation of the optimization problem is useful for ABCs in solving optimization problems without variable linkages, but harmful in solving optimization problems with variable linkages. Therefore, how to find a proper representation for optimization problems with variable linkages is the key to further improve the performance of ABCs.

This observation motivates us to develop a more efficient encoding learning method for further improving the performance of ABC. Based on these considerations and inspired by the researches in [32–36], we first analyze the rotational variability of the solution search equations in ABCs. Then, an AEL method is proposed by learning the information distribution of the current population. Moreover, AEL is further integrated into eight existing ABCs, and a class of new ABCs, AEL + ABCs for short, are proposed. Finally, systematic experiments are presented to verify the effectiveness of the proposed AEL + ABCs.

3.2. Rotational variance of ABC

In this subsection, the rotational variance characteristic of ABC is analyzed. For convenience's sake, rotation transformation mapping $\psi(\cdot)$ is defined as

$$\psi(\mathbf{x}) = \mathbf{R}\mathrm{x}, \forall \mathbf{x} \in \Omega, \forall \mathbf{R} \in Orth^{+} \tag{5}$$

where $\mathbf{x}$ is an arbitrary solution of search space Ω, and $\mathbf{R} \in Orth^{+}$ is an arbitrary orthogonal (rotation) matrix.

Let $\mathbf{x}$ be a solution obtained by a genetic operator in natural coordinate system. After rotation transformation, the new solution $\tilde{\mathbf{x}}$ is obtained under equal conditions for the genetic operator. If the relation between $\tilde{\mathbf{x}}$ and $\mathbf{x}$ satisfies Eq. (5), that is, $\tilde{\mathbf{x}} = \mathbf{R}\mathrm{x}$, the genetic operator is called rotationally invariant; otherwise, it is rotationally variant [37,38].

In ABC, the search equation is used to generate a new candidate solution, and its rotational variability is summarized in the following Theorem 1.

Theorem 1. *The search equation of ABC is rotationally variant.*

Proof. For the convenience of proof, the new candidate solution $\mathbf{v}_i$ is equally viewed as

$$\begin{aligned} \mathbf{v}_i &= \mathbf{M}_i(\mathbf{x}_i + \varphi_{ij_1}(\mathbf{x}_i - \mathbf{x}_k)) + (\mathbf{E} - \mathbf{M}_i)\mathbf{x}_i \\ &= \varphi_{ij_1}\mathbf{M}_i(\mathbf{x}_i - \mathbf{x}_k) + \mathbf{x}_i \end{aligned} \tag{6}$$

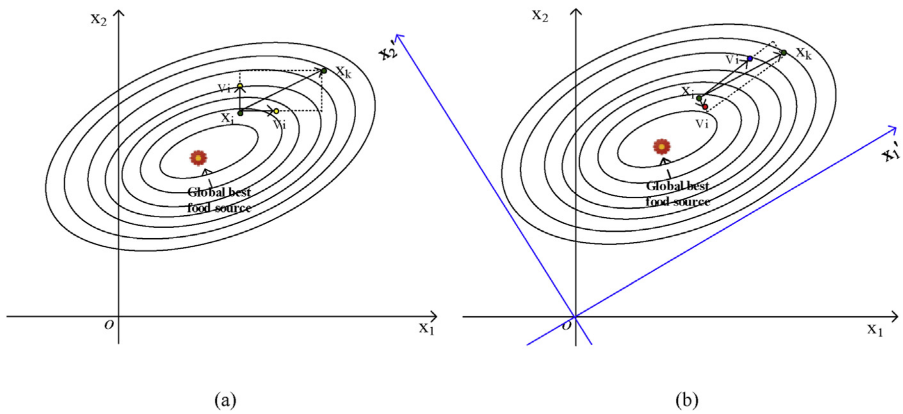


Fig. 2. (a) The search equation of ABC in the natural coordinate system. (b) The search equation of ABC in the eigen coordinate system.

where $\mathbf{M}_i$ is a diagonal matrix with 1 on the diagonal for $j=j_1$; otherwise, it is 0. $\mathbf{E}$ is the identity matrix. Then, the transformed candidate solution is

$$\begin{aligned}\tilde{\mathbf{v}}_i &= \varphi_{ij_1}\tilde{\mathbf{M}}_i(\tilde{\mathbf{x}}_i-\tilde{\mathbf{x}}_k)+\tilde{\mathbf{x}}_i \\ &= \varphi_{ij_1}\tilde{\mathbf{M}}_i\mathbf{R}(\mathbf{x}_i-\mathbf{x}_k)+R\mathbf{x}_i\end{aligned} \tag{7}$$

We assume that the search equation of ABC is rotationally invariant, that is, the transformed new candidate solution $\mathbf{v}_i$ is also expressed as

$$\begin{aligned}\tilde{\mathbf{v}}_i &= \mathbf{R}\mathbf{v}_i \\ &= \varphi_{ij_1}\mathbf{R}\mathbf{M}_i(\mathbf{x}_i-\mathbf{x}_k)+\mathbf{R}\mathrm{x}_i\end{aligned} \tag{8}$$

Comparing the real transformation in Eq. (7) to the required transformation in Eq. (8), the rotational invariance requires

$$\tilde{\mathbf{M}}_i=\mathbf{R}\mathbf{M}_i\mathbf{R}', \forall\mathbf{R}\in Orth^{+} \tag{9}$$

Since the relationship between $\tilde{\mathbf{M}}_i$ and $\mathbf{M}_i$ must hold for $\forall\mathbf{R}\in Orth^{+}$, the solution to Eq. (9) is

$$\tilde{\mathbf{M}}_i=\mathbf{M}_i=r\mathbf{E} \tag{10}$$

where r is an arbitrary real number. Therefore, a contradiction with the definition of $\mathbf{M}_i$ is obtained, which completes the proof. □

In order to intuitively understand the rotational variance of ABC, Fig. 1 (a) and (b) shows the process of the search equation and their rotation in a two-dimensional plane in terms of $\varphi_{ij_1}\in[0,1]$ and $\varphi_{ij_1}\in[-1,0]$, respectively. According to the characteristic of one dimensional mutation operator, it can be deduced that the new candidate solution $\mathbf{v}_i$ may be in one of the blue circles. When the $\mathbf{x}_i$ and $\mathbf{x}_k$ are rotated, the transformed new candidate solution $\tilde{\mathbf{v}}_i$ corresponds to one of the yellow circles and does not correspond to a pink circle. Therefore, the search equation of ABC is rotational variant.

Algorithm 2. The main framework of AEL + ABC.

Step 1) Initialization:
Step 1.1) Randomly initialize a population of SN individuals by Eq. (1).
Step 1.2) Evaluate the objective function value of each individual by Eq. (2).
Step 1.3) Set the selection probability $s_{k,G}$ and the counter $ns_{k,G}$,$k=1, 2$, $G=1$.
Step 1.4) Set learning period LP and $FES=SN$.
Step 2) Get the eigen coordinate system $\mathbf{R}$ by Eq. (13).
Step 3) The employed bee phase:
For $i=1, 2, \ldots, SN$ **do**
Step 3.1) If $rand(0, 1)<s_{1,G}$, generate the candidate solution $\mathbf{v}_{i,G}$ in natural coordinate system by Eq. (3), and set $k=1$, **otherwise** generate the candidate solution $\mathbf{v}_{i,G}$ in eigen coordinate system by Eqs. (14–17) and Eq. (3), and set $k=2$.
Step 3.2) Evaluate $\mathbf{v}_{i,G}$ and set $FES=FES+1$.
Step 3.3) If $f(\mathbf{v}_{i,G})<f(\mathbf{x}_{i,G})$, set $\mathbf{v}_{i,G}=\mathbf{x}_{i,G}$, $ns_{k,G}=ns_{k,G}+1$, $trial_i=0$, **otherwise** set $trial_i=trial_i+1$.
Step 4) Calculate the probability values $p_{i,G}$ by Eq. (4), and set $t=0$, $i=1$.
Step 5) The onlookers phase:
While $t\le SN$ **do**
Step 5.1) If $rand(0, 1)<p_{i,G}$
Step 5.1.1) If $rand(0, 1)<s_{1,G}$, generate the candidate solution $\mathbf{v}_{i,G}$ in natural coordinate system by Eq. (3), and set $k=1$, **otherwise** generate the candidate solution $\mathbf{v}_{i,G}$ in eigen coordinate system by Eqs. (14–17) and Eq. (3), and set $k=2$.
Step 5.1.2) Evaluate $\mathbf{v}_{i,G}$ and set $FES=FES+1$.
Step 5.1.3) If $f(\mathbf{v}_{i,G})<f(\mathbf{x}_{i,G})$, set $\mathbf{v}_{i,G}=\mathbf{x}_{i,G}$, $ns_{k,G}=ns_{k,G}+1$, $trial_i=0$, **otherwise** set $trial_i=trial_i+1$.
Step 5.1.4) Set $t=t+1$.
Step 5.2) Set $i=i+1$, **If** $i=SN$, set $i=1$.
Step 6) The scout phase:
If $max(trial_i)>limit$, replace $\mathbf{x}_i$ with a new solution produced by Eq. (1) and set $FES=FES+1$.
Step 7) Update the selection probability $s_{k,G}(k=1, 2)$ by Eq. (18).
Step 8) $G=G+1$. **If** $FES>MaxFEs$, stop and output the best solution, **otherwise** go to **Step 2)**.

3.3. The main framework of AEL + ABCs

The proposed framework in terms of AEL + ABC is summarized in algorithm 2, from which we can see that the main differences between ABC and AEL + ABC include two main aspects: adaptive encoding learning (Step 2, Step 3.1 and Step 5.1.1) and adaptive selection mechanism (Step 7). The following subsections will detail the two aspects.

- Adaptive encoding learning

According to the above discussion, we have known that the search equation of ABC is rotationally variant, which means that ABC is dependent on the reference coordinate system. Therefore, there exists a specific choice of reference coordinate system in which the optimization problem can be solved easier or better by ABC as compared to some other reference coordinate system. In

Table 1
The description for 30 CEC2014 benchmark functions.

Function	Name	Characteristics	Search space	Optima
F1	Rotated High Conditioned Elliptic Function	Unimodal + Non-separable	$[-100, 100]^D$	100
F2	Rotated Bent Cigar Function	Unimodal + Non-separable	$[-100, 100]^D$	200
F3	Rotated Discus Function	Unimodal + Non-separable	$[-100, 100]^D$	300
F4	Shifted and Rotated Rosenbrock's Function	Simple multimodal + Non-separable	$[-100, 100]^D$	400
F5	Shifted and Rotated Ackley's Function	Simple multimodal + Non-separable	$[-100, 100]^D$	500
F6	Shifted and Rotated Weierstrass Function	Simple multimodal + Non-separable	$[-100, 100]^D$	600
F7	Shifted and Rotated Griewank's Function	Simple multimodal + Non-separable	$[-100, 100]^D$	700
F8	Shifted Rastrigin's Function	Simple multimodal + Non-separable	$[-100, 100]^D$	800
F9	Shifted and Rotated Rastrigin's Function	Simple multimodal + Non-separable	$[-100, 100]^D$	900
F10	Shifted Schwefel's Function	Simple multimodal + Non-separable	$[-100, 100]^D$	1000
F11	Shifted and Rotated Schwefel's Function	Simple multimodal + Non-separable	$[-100, 100]^D$	1100
F12	Shifted and Rotated Katsuura Function	Simple multimodal + Non-separable	$[-100, 100]^D$	1200
F13	Shifted and Rotated HappyCat Function	Simple multimodal + Non-separable	$[-100, 100]^D$	1300
F14	Shifted and Rotated HGBat Function	Simple multimodal + Non-separable	$[-100, 100]^D$	1400
F15	Shifted and Rotated Expanded Griewank's plus Rosenbrock's Function	Simple multimodal + Non-separable	$[-100, 100]^D$	1500
F16	Shifted and Rotated Expanded Scaffer's F6 Function	Simple multimodal + Non-separable	$[-100, 100]^D$	1600
F17	Hybrid Function 1	Complex multimodal + Non-separable	$[-100, 100]^D$	1700
F18	Hybrid Function 2	Complex multimodal + Non-separable	$[-100, 100]^D$	1800
F19	Hybrid Function 3	Complex multimodal + Non-separable	$[-100, 100]^D$	1900
F20	Hybrid Function 4	Complex multimodal + Non-separable	$[-100, 100]^D$	2000
F21	Hybrid Function 5	Complex multimodal + Non-separable	$[-100, 100]^D$	2100
F22	Hybrid Function 6	Complex multimodal + Non-separable	$[-100, 100]^D$	2200
F23	Composition Function 1	Complex multimodal + Non-separable	$[-100, 100]^D$	2300
F24	Composition Function 2	Complex multimodal + Non-separable	$[-100, 100]^D$	2400
F25	Composition Function 3	Complex multimodal + Non-separable	$[-100, 100]^D$	2500
F26	Composition Function 4	Complex multimodal + Non-separable	$[-100, 100]^D$	2600
F27	Composition Function 5	Complex multimodal + Non-separable	$[-100, 100]^D$	2700
F28	Composition Function 6	Complex multimodal + Non-separable	$[-100, 100]^D$	2800
F29	Composition Function 7	Complex multimodal + Non-separable	$[-100, 100]^D$	2900
F30	Composition Function 8	Complex multimodal + Non-separable	$[-100, 100]^D$	3000

general, the performance difference of ABC in different reference coordinate system cannot be quantified, and depends on the characteristics of optimization problem. However, the natural coordinate system used in ABCs cannot identify the relationship between variables well, which results in performing poorly in solving SNOPs with variable linkages. Therefore, how to find a new coordinate system that can effectively reduce the correlation of variable is the key to improve the performance of ABCs in solving SNOPs with variable linkages.

On the basis of above consideration, the covariance matrix composed of variance and covariance is introduced in this work, the main aim of which is to establish an eigen coordinate system with release variable correlation for the ABCs search process. Therefore, during the evolution, systemically utilizing the adaptive encoding

Table 2
The public parameter settings for the compared ABCs.

Parameter settings
Dimension of each function: D = 30/50/100
Population size: SN = 2D
Maximum number of function evaluations ($MaxFEs$) : $MaxFEs$ = 10000D [38]
Independent Number of runs ($NumR$) : $NumR$ = 51 [38]
Learning period (LP) : LP = 50

Table 3
The private parameter settings for the compared ABCs.

Algorithms	Parameter settings
ABC [1]	$limit = 100$
ABCM [15]	$limit = SN * D$, $M = 2$
ABCVSS [18]	$limit = SN * D$
EABC [10]	$limit = 200$, $\mu = 0.3$, $\sigma = 0.3$
GABC [8]	$limit = 200$, $C = 1.5$
MABC [11]	$K = 301$
OPIABC [14]	$limit = 200$
qABC [12]	$limit = SN * D$

Table 4
The statistical results (Mean (Std)) of ABC vs. AEL + ABC, ABCM vs. AEL + ABCM, ABCVSS vs. AEL + ABCVSS and EABC vs. AEL + EABC over 51 independent runs on the CEC2014 benchmarks with 30*D*.

Function	ABC vs.	AEL + ABC	ABCM vs.	AEL + ABCM	ABCVSS vs.	AEL + ABCVSS	EABC vs.	AEL + EABC
F1	6.21E+06‡	**2.79E+04**	4.79E+06‡	**2.00E+04**	6.04E+06‡	**5.79E+04**	7.59E+06‡	**1.90E+05**
	(2.52E+06)	(1.98E+04)	(1.76E+06)	(1.89E+04)	(2.82E+06)	(3.43E+04)	(3.12E+06)	(9.84E+04)
F2	1.46E+02‡	**1.33E−09**	5.42E+01‡	**3.16E−06**	1.48E+02‡	**2.95E−14**	3.84E+03‡	**4.82E+01**
	(2.26E+02)	(1.22E−09)	(6.32E+01)	(2.25E−05)	(3.72E+02)	(9.78E−15)	(9.61E+03)	(1.75E+02)
F3	5.92E+02‡	**2.12E−08**	4.72E+02‡	**9.14E−14**	7.50E+02‡	**5.80E−14**	9.60E+02‡	**5.57E−14**
	(4.59E+02)	(8.89E−08)	(3.81E+02)	(3.61E−14)	(7.54E+02)	(7.96E−15)	(9.44E+02)	(7.96E−15)
F4	2.25E+01‡	**1.40E+01**	2.41E+01‡	**1.45E+01**	2.06E+01‡	**1.47E+01**	2.84E+01‡	**1.46E+01**
	(2.43E+01)	(2.58E+01)	(2.46E+01)	(2.49E+01)	(2.23E+01)	(2.63E+01)	(3.16E+01)	(2.70E+01)
F5	2.02E+01§	2.02E+01	2.01E+01§	2.01E+01	2.02E+01§	2.02E+01	2.02E+01§	2.02E+01
	(3.97E−02)	(3.32E−02)	(2.04E−02)	(1.78E−02)	(3.13E−02)	(2.61E−02)	(4.70E−02)	(4.89E−02)
F6	**1.35E+01**†	1.43E+01	**1.43E+01**§	1.44E+01	1.37E+01§	1.37E+01	1.21E+01§	**1.20E+01**
	(1.72E+00)	(1.46E+00)	(1.60E+00)	(1.61E+00)	(1.32E+00)	(1.35E+00)	(1.54E+00)	(1.52E+00)
F7	4.65E−05‡	**2.16E−06**	8.90E−06‡	**2.19E−06**	4.48E−05‡	**3.26E−06**	9.29E−03‡	**3.21E−07**
	(3.26E−05)	(3.66E−06)	(1.03E−05)	(2.80E−06)	(6.52E−05)	(5.09E−06)	(2.16E−02)	(1.10E−06)
F8	3.37E−13‡	**1.98E−13**	1.14E−13§	1.14E−13	1.18E−13§	1.18E−13	1.09E−13§	**1.00E−13**
	(9.08E−14)	(5.00E−14)	(0.00E+00)	(0.00E+00)	(2.23E−14)	(2.23E−14)	(2.23E−14)	(3.70E−14)
F9	**7.78E+01**†	9.54E+01	**8.34E+01**†	9.27E+01	**5.49E+01**†	6.06E+01	**3.18E+01**†	3.75E+01
	(1.12E+01)	(1.44E+01)	(1.16E+01)	(1.18E+01)	(8.96E+00)	(1.20E+01)	(5.93E+00)	(7.11E+00)
F10	1.20E+00§	**1.09E+00**	**1.44E+00**§	1.57E+00	3.84E−02§	**2.98E−02**	4.95E−01§	**2.59E−01**
	(6.82E−01)	(8.22E−01)	(8.77E−01)	(6.91E−01)	(3.04E−02)	(2.61E−02)	(7.62E−01)	(3.19E−01)
F11	**1.86E+03**§	1.88E+03	**1.91E+03**§	1.94E+03	**1.78E+03**†	1.96E+03	**1.80E+03**§	1.84E+03
	(2.91E+02)	(2.60E+02)	(2.82E+02)	(2.48E+02)	(2.23E+02)	(3.34E+02)	(3.56E+02)	(3.33E+02)
F12	2.21E−01‡	**2.00E−01**	1.67E−01§	**1.65E−01**	1.90E−01§	**1.88E−01**	**2.30E−01**§	2.40E−01
	(5.70E−02)	(4.24E−02)	(2.27E−02)	(2.68E−02)	(3.06E−02)	(3.22E−02)	(5.91E−02)	(6.56E−02)
F13	**2.17E−01**†	2.28E−01	2.26E−01§	**2.20E−01**	**2.18E−01**†	2.49E−01	**1.10E−01**†	2.29E−01
	(3.06E−02)	(3.39E−02)	(2.84E−02)	(3.00E−02)	(2.93E−02)	(4.02E−02)	(2.33E−02)	(3.23E−02)
F14	1.82E−01‡	**1.71E−01**	2.08E−01‡	**1.84E−01**	2.18E−01‡	**1.90E−01**	1.94E−01‡	**1.72E−01**
	(1.78E−02)	(2.00E−02)	(1.69E−02)	(2.21E−02)	(2.47E−02)	(2.09E−02)	(2.17E−02)	(1.90E−02)
F15	9.10E+00‡	**7.75E+00**	8.56E+00‡	**7.37E+00**	6.08E+00§	**6.06E+00**	**3.85E+00**§	4.06E+00
	(1.54E+00)	(1.18E+00)	(1.50E+00)	(1.18E+00)	(1.35E+00)	(1.05E+00)	(6.94E−01)	(7.90E−01)
F16	9.99E+00§	**9.95E+00**	9.79E+00§	**9.77E+00**	**9.53E+00**§	9.59E+00	9.14E+00§	**9.12E+00**
	(4.32E−01)	(2.96E−01)	(3.82E−01)	(3.58E−01)	(3.95E−01)	(3.34E−01)	(4.80E−01)	(4.80E−01)
F17	1.84E+06‡	**4.95E+02**	1.56E+06‡	**5.19E+02**	2.26E+06‡	**4.46E+02**	1.79E+06‡	**5.63E+02**
	(8.68E+05)	(1.53E+02)	(7.03E+05)	(1.20E+02)	(1.04E+06)	(1.54E+02)	(9.20E+05)	(2.59E+02)
F18	1.59E+03‡	**2.33E+01**	4.96E+02‡	**2.35E+01**	3.83E+02‡	**1.68E+01**	1.64E+03‡	**2.75E+01**
	(1.68E+03)	(7.60E+00)	(3.31E+02)	(5.24E+00)	(4.28E+02)	(6.99E+00)	(2.14E+03)	(1.27E+01)
F19	7.21E+00‡	**6.74E+00**	7.21E+00‡	**6.83E+00**	6.84E+00‡	**6.32E+00**	6.77E+00‡	**5.39E+00**
	(6.35E−01)	(6.17E−01)	(6.30E−01)	(5.39E−01)	(7.64E−01)	(6.87E−01)	(9.70E−01)	(8.24E−01)
F20	4.01E+03‡	**2.05E+01**	7.50E+03‡	**2.11E+01**	6.68E+03‡	**1.31E+01**	3.23E+03‡	**1.27E+01**
	(1.88E+03)	(4.86E+00)	(3.18E+03)	(4.60E+00)	(3.00E+03)	(3.56E+00)	(1.65E+03)	(5.08E+00)
F21	2.42E+05‡	**1.62E+02**	1.36E+05‡	**1.74E+02**	2.00E+05‡	**1.64E+02**	3.86E+05‡	**2.23E+02**
	(1.36E+05)	(8.77E+01)	(7.53E+04)	(7.76E+01)	(1.01E+05)	(8.11E+01)	(2.62E+05)	(1.00E+02)
F22	**2.09E+02**§	2.16E+02	2.75E+02§	**2.50E+02**	2.56E+02§	**2.39E+02**	2.53E+02‡	**1.86E+02**
	(8.22E+01)	(7.74E+01)	(1.00E+02)	(9.09E+01)	(1.06E+02)	(7.75E+01)	(1.04E+02)	(7.48E+01)
F23	3.15E+02‡	3.15E+02	3.15E+02‡	3.15E+02	3.15E+02‡	3.15E+02	3.15E+02‡	3.15E+02
	(8.73E−02)	(4.51E−12)	(5.50E−02)	(3.16E−05)	(1.35E−01)	(8.88E−13)	(3.13E−01)	(9.67E−04)
F24	2.27E+02§	**2.26E+02**	2.27E+02§	**2.26E+02**	**2.26E+02**§	2.27E+02	2.25E+02‡	**2.24E+02**
	(6.15E−01)	(4.91E+00)	(3.87E+00)	(5.55E+00)	(5.74E+00)	(3.18E+00)	(1.12E+00)	(8.56E−01)
F25	2.08E+02‡	**2.03E+02**	2.08E+02‡	**2.03E+02**	2.09E+02‡	**2.03E+02**	2.08E+02‡	**2.03E+02**
	(9.72E−01)	(2.95E−01)	(1.23E+00)	(2.99E−01)	(1.13E+00)	(2.72E−01)	(9.93E−01)	(4.36E−01)
F26	1.00E+02‡	1.00E+02	1.00E+02‡	1.00E+02	1.00E+02‡	1.00E+02	1.00E+02§	1.00E+02
	(5.93E−02)	(3.60E−02)	(5.47E−02)	(3.72E−02)	(5.43E−02)	(4.36E−02)	(5.10E−02)	(4.30E−02)
F27	**4.00E+02**†	4.01E+02	4.09E+02‡	**4.06E+02**	4.12E+02‡	**4.05E+02**	4.09E+02‡	**3.93E+02**
	(5.60E+01)	(3.10E+00)	(2.24E+00)	(2.26E+00)	(4.19E+00)	(3.62E+00)	(2.72E+00)	(5.53E+01)
F28	**1.09E+03**§	1.12E+03	1.20E+03§	**1.18E+03**	**9.41E+02**†	9.80E+02	**8.51E+02**§	8.59E+02
	(7.81E+01)	(1.16E+02)	(1.40E+02)	(1.26E+02)	(5.91E+01)	(6.54E+01)	(5.29E+01)	(3.94E+01)
F29	1.03E+03‡	**7.36E+02**	9.92E+02§	**9.78E+02**	1.16E+03‡	**7.15E+02**	1.42E+03‡	**8.01E+02**
	(8.02E+01)	(8.68E+01)	(6.70E+01)	(6.81E+01)	(1.11E+02)	(6.80E−01)	(2.61E+02)	(7.95E+01)
F30	2.93E+03‡	**1.63E+03**	2.64E+03‡	**2.00E+03**	3.09E+03‡	**1.48E+03**	3.85E+03‡	**1.40E+03**
	(7.70E+02)	(3.46E+02)	(6.38E+02)	(3.79E+02)	(8.26E+02)	(2.82E+02)	(1.06E+03)	(2.35E+02)
‡/†/§	19/4/7		17/1/12		17/4/9		18/2/10	

learning of the search equation in the eigen coordinate system may make the SNOPs with variable linkages easier to be solved by loosing the interactions among the variables. In order to intuitively understand the difference between two kinds of coordinate systems, Fig. 2 shows the search equation of ABC implemented in the natural system $x_1 - o - x_2$ (Fig. 2 (a)) and that in the eigen coordinate system $x'_1 - o' - x'_2$ (Fig. 2(b)) for a SNOP with variable linkage. From Fig. 2, we can see that the generated candidate solution v_i in the eigen coordinate system is more promising to find the global best food source, since the solution v_i in the eigen coordinate system looks more close to the optimal position than that in the natural coordinate system.

The adaptive encoding learning includes three core techniques: eigen decomposition of the covariance matrix (Step 1), solution encoding learning in the eigen coordinate system (Step 2) and solution decoding (Step 3). The Step 1 is to built the eigen coordinate system by eigen decomposition of the covariance matrix, the purpose of Step 2 is to implement the search equation in the eigen coordinate system for generating better candidate solution, and the aim of Step 3 is to transform the generated candidate solution into the natural coordinate system for evaluation. The procedure of the adaptive encoding learning in ABC is described as follows.

Step 1. (a) Compute the covariance matrix $cov(\mathbf{P}_{1:SN/2})$ of the top $SN/2$ individuals in the current population as follows:

$$cov(\mathbf{P}_{1:SN/2}) = [cov(i,j)]_{D\times D} \tag{11}$$

where $cov(i,j)$ is the covariance of the ith and the jth dimensions of the top $SN/2$ individuals in the current population. It can be computed as

$$cov(k,j) = \frac{1}{SN/2-1}\sum_{i=1}^{SN/2}(x_{i,k}-\bar{x}_k)(x_{i,j}-\bar{x}_j),$$
$$k = 1, 2, \ldots, D;\ j = 1, 2, \ldots, D, \tag{12}$$

where $\bar{x}_k$ and $\bar{x}_j$ are the mean values of the kth and the jth dimension of the top $SN/2$ individuals in the current population, respectively.

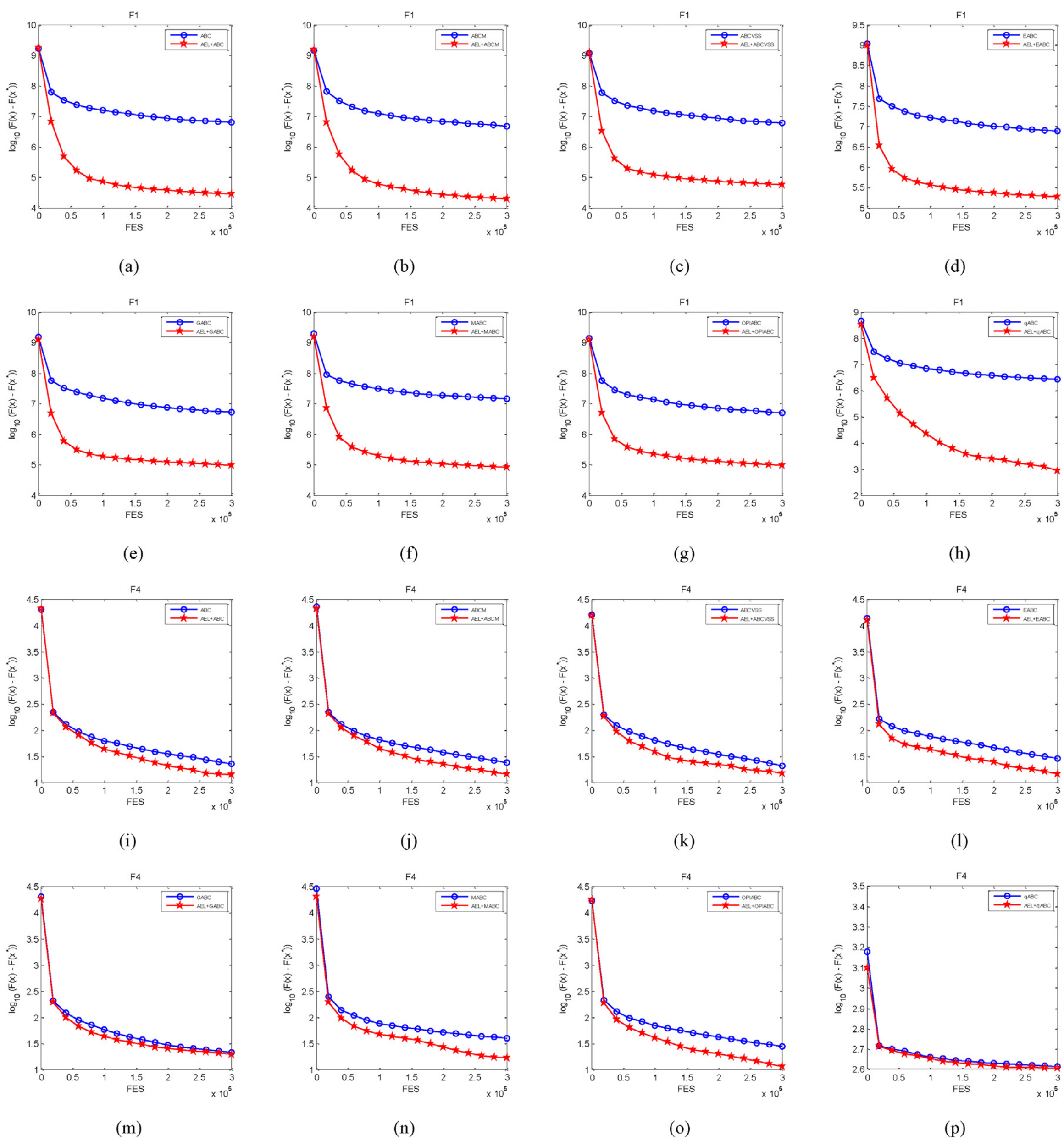


Fig. 3. The mean function error values versus numbers of function evaluations for the eight groups of compared algorithms over 51 independent runs on F1 and F4 with 30D.

(b) Apply eigen decomposition to $cov(\mathbf{P}_{1:SN/2})$ as follows:

$$cov(\mathbf{P}_{1:SN/2}) = \mathbf{R}\Lambda^2\mathbf{R}' \tag{13}$$

where $\mathbf{R}$ is a $D \times D$ orthogonal matrix which represents the eigen coordinate system, and each column of $\mathbf{R}$ is an eigen vector of the covariance matrix $cov(\mathbf{P}_{1:SN/2})$. $\mathbf{R}'$ represents the transformation from eigen coordinate system to natural coordinate system, and Λ is a diagonal matrix composed of eigen values.

Step 2. (a) Encode the target vectors $\mathbf{x}_i$ and $\mathbf{x}_k$ in the eigen coordinate system using $\mathbf{R}$:

$$\tilde{\mathbf{x}}_i = \mathbf{x}_i\mathbf{R} \tag{14}$$

$$\tilde{\mathbf{x}}_k = \mathbf{x}_k\mathbf{R} \tag{15}$$

(b) Apply the search equation to $\tilde{\mathbf{x}}_i$ and $\tilde{\mathbf{x}}_k$, and generate a candidate solution $\tilde{\mathbf{v}}_i$ in the eigen coordinate system:

$$\tilde{v}_{i,j} = \begin{cases} \tilde{x}_{i,j} + \phi_{i,j1}\cdot(\tilde{x}_{i,j} - \tilde{x}_{k,j}),\ if j = j_1, \\ \tilde{x}_{i,j},\ otherwise, \end{cases} \tag{16}$$

Step 3. Decode the candidate solution $\tilde{\mathbf{v}}_k$ using $\mathbf{R}'$:

$$\mathbf{v}_i = \tilde{\mathbf{v}}_i\mathbf{R}' \tag{17}$$

Remark 1. In Step 1, the calculation of covariance matrix is based on the top $SN/2$ individuals instead of all the individuals in the population, which is to construct more reasonable eigen coordinate system by reducing the randomness of the population distribution.

Remark 2. Recently, Kumar and Mishra proposed a covariance guided ABC (CABC) [17] which was a unification of ABC and statistical covariance. In CABC, the covariance matrix was used to construct a new solution search equation $\mathbf{v}_i = m + \sigma B\,D\mathbf{r}$ in onlooker bee phase,

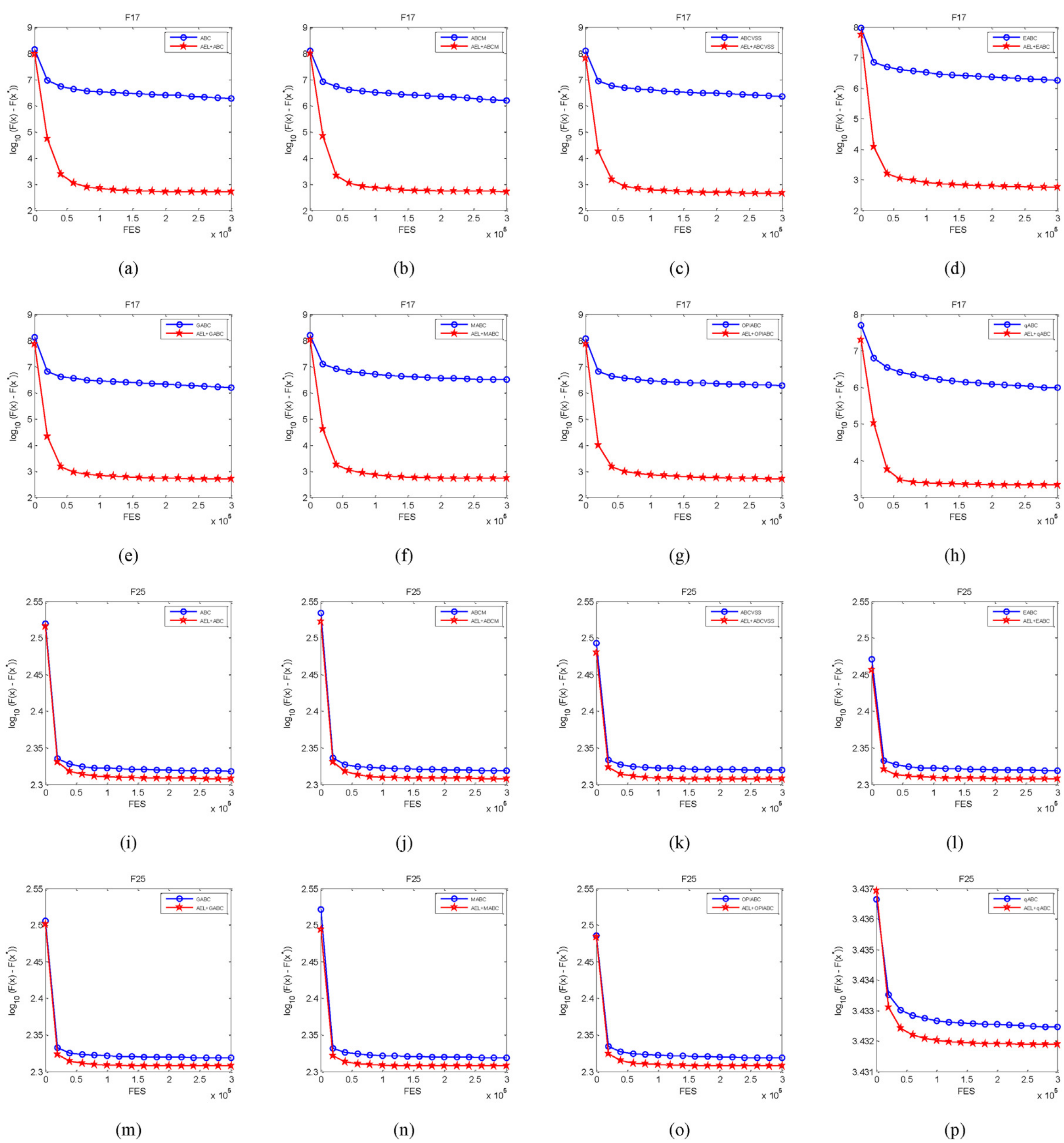


Fig. 4. The mean function error values versus numbers of function evaluations for the eight groups of compared algorithms over 51 independent runs on F17 and F25 with 30*D*.

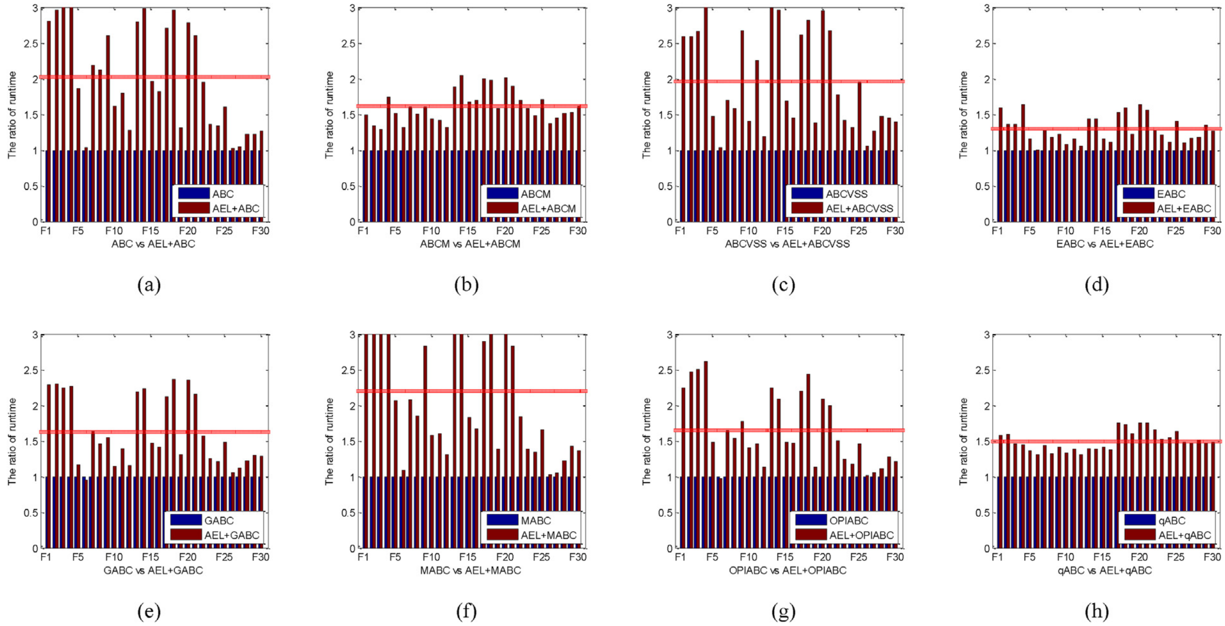


Fig. 5. The ratio of runtime for the eight groups of compared algorithms on the 30 test problems with 30*D*.

where m is the mean solution, σ is a constant of value 0.5, $\mathbf{r}$ is a random vector, matrices B and D are obtained from the covariance matrix. However, in AEL + ABC, the covariance matrix is used to built the eigen coordinate system, which can make the SNOPs with variable linkages easier to be solved by loosing the interactions among the variables. Therefore, there is an essential difference between CABC and AEL + ABC.

- Adaptive selection mechanism

In the framework of AEL + ABC, we use the selection probability s_k to adjust the effect of the adaptive encoding learning on the performance. The main reason is explained as follows. Although the adaptive encoding learning based on the eigen coordinate system is an effective way to make the SNOPs with variable linkages easier to be solved by loosing the interactions among the variables, it is a relatively greedy and deterministic mechanism due to the use of the top $SN/2$ individuals in the population to compute the covariance matrix. As a result, the performance of the algorithm may degrade for some complex optimization problems because of the rapid loss of population diversity. Based on this consideration, the solution search equation is implemented in the natural coordinate system with a probability $(1-s_k)$ to encourage the population diversity. With respect to AEL + ABC, combining these two types of solution search equation can achieve a good tradeoff between convergence and diversity by adaptively adjusting the parameter s_k. The following subsections will detail the adaptive selection mechanism.

Inspired by the adaptive mechanism proposed in [39], we introduced a similar method to AEL + ABC. For each coordinate system at generation G, the AEL + ABC records the number of the candidate solutions ($ns_{k,G}$) generated by the solution search equation under the kth coordinate system ($k = 1, 2$) which can enter the next stage. In addition, an additional parameter, LP, which is called the learning period, is introduced to update the $ns_{k,G}$ dynamically. During its first LP generations, the solution search equation chooses the two coordinate systems with the same probability (i.e. 1/2). When the generation number G is larger than LP, the selection probability $s_{k,G}$ of using the kth coordinate system is calculated as follows:

$$s_{k,G} = \frac{\sum_{g=G-LP}^{G-1} ns_{k,g}}{\sum_{k=1}^{2}\sum_{g=G-LP}^{G-1} ns_{k,g} + \xi} \tag{18}$$

where $k = 1, 2$, $G > LP$, and ξ is a small constant to prevent $ns_{k,g}$ from becoming zero.

Eq. (18) indicates the relative success rate of solution search equation under the kth coordinate system during the previous LP generations. Finally, the roulette wheel method is used to select one coordinate system based on Eq. (18). Clearly, the larger $ns_{k,g}$ is obtained, the larger the selection probability $s_{k,G}$ will be.

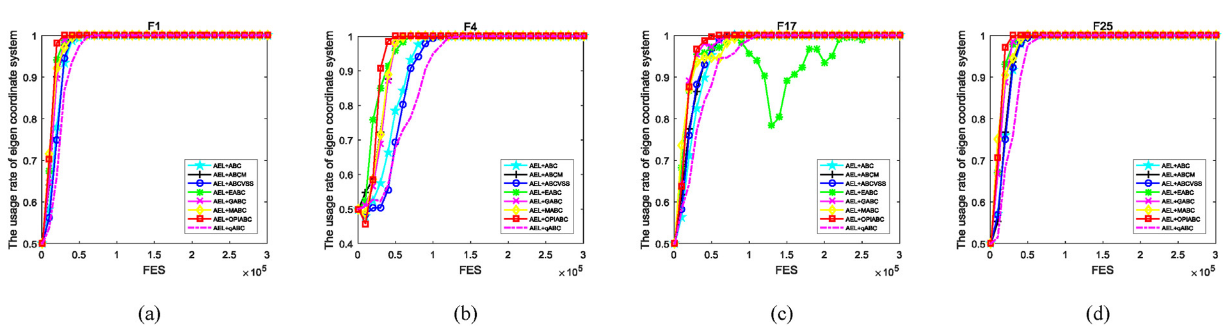


Fig. 6. The variations of the usage rate of eigen coordinate system as the evolution progress on F1, F4, F17 and F25.

Table 5
The statistical results (Mean (Std)) of GABC vs. AEL + GABC, MABC vs. AEL + MABC, OPIABC vs. AEL + OPIABC and qABC vs. AEL + qABC over 51 independent runs on the CEC2014 benchmarks with 30*D*.

Function	GABC vs.	AEL + GABC	MABC vs.	AEL + MABC	OPIABC vs.	AEL + OPIABC	qABC vs.	AEL + qABC
F1	5.07E+06‡	**9.48E+04**	1.40E+07‡	**8.32E+04**	5.02E+06‡	**9.69E+04**	2.60E+06‡	**8.85E+02**
	(2.46E+06)	(7.80E+04)	(4.20E+06)	(4.41E+04)	(2.24E+06)	(5.90E+04)	(1.18E+06)	(1.25E+03)
F2	9.58E+01‡	**6.85E−14**	8.73E+01‡	**2.57E+01**	1.48E+02‡	**1.83E+00**	3.59E+02‡	**2.00E+02**
	(1.89E+02)	(2.38E−13)	(1.35E+02)	(1.69E+02)	(2.04E+02)	(1.03E+01)	(1.84E+02)	(2.55E−07)
F3	3.50E+02‡	**7.58E−14**	1.27E+03‡	**4.59E−11**	3.56E+02‡	**7.58E−14**	9.48E+02‡	**3.00E+02**
	(2.70E+02)	(2.71E−14)	(1.01E+03)	(3.01E−10)	(3.20E+02)	(3.35E−14)	(5.06E+02)	(1.54E−10)
F4	2.10E+01‡	**1.95E+01**	3.95E+01‡	**1.67E+01**	2.79E+01‡	**1.16E+01**	4.13E+02‡	**4.06E+02**
	(2.55E+01)	(2.96E+01)	(2.92E+01)	(2.77E+01)	(2.49E+01)	(2.35E+01)	(2.41E+01)	(1.64E+01)
F5	2.02E+01§	**2.01E+01**	2.01E+01§	2.01E+01	2.03E+01§	**2.02E+01**	5.20E+02‡	5.20E+02
	(3.43E−02)	(4.30E−02)	(2.75E−02)	(2.93E−02)	(4.54E−02)	(5.50E−02)	(2.85E−03)	(1.89E−03)
F6	**1.29E+01**§	1.31E+01	1.42E+01‡	**1.35E+01**	1.28E+01§	**1.24E+01**	**6.14E+02**†	6.15E+02
	(1.33E+00)	(1.33E+00)	(1.32E+00)	(1.46E+00)	(1.36E+00)	(1.26E+00)	(1.68E+00)	(1.58E+00)
F7	2.34E−05‡	**1.02E−06**	4.01E−05 ‡	**4.32E−06**	2.21E−06‡	**8.13E−07**	7.00E+02‡	7.00E+02
	(5.63E−05)	(1.61E−06)	(6.22E−05)	(1.79E−05)	(2.81E−06)	(1.44E−06)	(3.87E−07)	(2.25E−07)
F8	1.65E−13§	**1.60E−13**	**2.59E−13**§	2.67E−13	**2.01E−14**§	3.57E−14	8.00E+02§	8.00E+02
	(5.71E−14)	(5.65E−14)	(5.12E−14)	(5.94E−14)	(4.38E−14)	(5.33E−14)	(3.90E−13)	(3.69E−13)
F9	**5.70E+01**†	6.38E+01	**5.30E+01**§	5.52E+01	**5.25E+01**†	5.74E+01	**9.79E+02**†	9.92E+02
	(7.87E+00)	(8.81E+00)	(7.79E+00)	(8.08E+00)	(7.91E+00)	(7.77E+00)	(1.31E+01)	(1.33E+01)
F10	**3.54E−01**§	4.22E−01	3.99E−01§	**3.26E−01**	**3.37E−02**§	3.40E−02	1.00E+03§	1.00E+03
	(92.78E−01)	(4.35E−01)	(6.76E−01)	(4.95E−01)	(2.48E−02)	(2.85E−02)	(2.07E−02)	(1.83E−02)
F11	**1.70E+03**†	1.81E+03	**1.78E+03**§	1.81E+03	**1.67E+03**§	1.78E+03	**2.96E+03**§	3.05E+03
	(2.17E+02)	(2.54E+02)	(2.52E+02)	(1.88E+02)	(2.72E+02)	(2.19E+02)	(2.56E+02)	(2.27E+02)
F12	2.04E−01§	**1.97E−01**	**1.78E−01**§	1.82E−01	2.94E−01§	**2.75E−01**	1.20E+03†	1.20E+03
	(4.53E−02)	(4.24E−02)	(3.31E−02)	(3.04E−02)	(6.07E−02)	(5.51E−02)	(3.28E−02)	(3.86E−02)
F13	**2.06E−01**†	2.30E−01	**2.64E−01**†	2.89E−01	**2.20E−01**†	2.46E−01	1.30E+03†	1.30E+03
	(3.18E−02)	(3.23E−02)	(4.46E−02)	(3.66E−02)	(2.67E−02)	(3.06E−02)	(3.60E−02)	(3.68E−02)
F14	1.74E−01§	**1.69E−01**	2.51E−01‡	**2.03E−01**	2.15E−01‡	**1.88E−01**	1.40E+03‡	1.40E+03
	(1.93E−02)	(1.70E−02)	(3.58E−02)	(2.30E−02)	(1.86E−02)	(2.34E−02)	(1.99E−02)	(2.26E−02)
F15	5.91E+00§	5.91E+00	5.79E+00§	**5.62E+00**	5.79E+00‡	**5.49E+00**	1.51E+03§	1.51E+03
	(7.02E−01)	(8.51E−01)	(8.47E−01)	(8.80E−01)	(8.84E−01)	(7.12E−01)	(1.39E+00)	(1.40E+00)
F16	9.62E+00‡	**9.45E+00**	9.60E+00§	**9.48E+00**	**9.71E+00**†	9.88E+00	1.61E+03†	1.61E+03
	(3.62E−01)	(3.89E−01)	(3.55E−01)	(3.48E−01)	(3.77E−01)	(2.89E−01)	(5.17E−01)	(4.74E−01)
F17	1.57E+06‡	**4.97E+02**	3.07E+06‡	**5.27E+02**	1.87E+06‡	**5.14E+02**	9.75E+05‡	**2.15E+03**
	(7.70E+05)	(1.32E+02)	(1.85E+06)	(1.87E+02)	(7.98E+05)	(1.55E+02)	(6.54E+05)	(1.71E+02)
F18	2.13E+03‡	**2.53E+01**	6.23E+03‡	**3.36E+01**	6.41E+02‡	**2.03E+01**	2.25E+03‡	**1.83E+03**
	(2.43E+03)	(8.68E+00)	(4.67E+03)	(1.45E+01)	(7.16E+02)	(7.57E+00)	(5.55E+02)	(8.21E+00)
F19	6.75E+00‡	**6.14E+00**	7.20E+00‡	**6.22E+00**	6.65E+00‡	**6.40E+00**	1.91E+03‡	1.91E+03
	(6.40E−01)	(6.91E−01)	(8.35E−01)	(8.28E−01)	(6.30E−01)	(5.82E−01)	(8.98E−01)	(8.28E−01)
F20	2.93E+03‡	**1.60E+01**	7.17E+03‡	**1.68E+01**	4.99E+03‡	**1.78E+01**	8.50E+03‡	**2.02E+03**
	(1.42E+03)	(4.21E+00)	(4.12E+03)	(5.73E+00)	(2.69E+03)	(4.16E+00)	(3.32E+03)	(5.89E+00)
F21	2.13E+05‡	**1.77E+02**	5.66E+05‡	**2.07E+02**	1.99E+05‡	**1.45E+02**	1.84E+05‡	**2.29E+03**
	(1.22E+05)	(7.94E+01)	(2.86E+05)	(9.00E+01)	(9.66E+04)	(8.68E+01)	(1.56E+05)	(9.48E+01)
F22	1.80E+02§	**1.74E+02**	2.60E+02‡	**2.19E+02**	**1.90E+02**§	2.13E+02	2.49E+03§	**2.46E+03**
	(7.51E+01)	(9.40E+01)	(1.14E+02)	(7.58E+01)	(8.40E+01)	(7.75E+01)	(7.83E+01)	(9.12E+01)
F23	3.15E+02‡	3.15E+02	3.16E+02‡	**3.15E+02**	3.15E+02‡	3.15E+02	2.62E+03‡	2.62E+03
	(1.13E−01)	(2.21E−13)	(2.22E−01)	(2.52E−06)	(7.20E−02)	(3.62E−13)	(4.16E−02)	(4.76E−06)
F24	**2.21E+02**§	2.22E+02	**2.24E+02**§	2.26E+02	**2.23E+02**†	2.24E+02	2.63E+03‡	2.63E+03
	(1.15E+01)	(1.21E+01)	(8.86E+00)	(4.80E+00)	(7.80E+00)	(5.06E+00)	(2.56E+00)	(8.05E−01)
F25	2.08E+02‡	**2.03E+02**	2.08E+02‡	**2.03E+02**	2.08E+02‡	**2.03E+02**	2.71E+03‡	**2.70E+03**
	(1.01E+00)	(2.57E−01)	(1.27E+00)	(1.97E−01)	(9.63E−01)	(3.38E−01)	(1.61E+00)	(3.31E−01)
F26	1.00E+02‡	1.00E+02	1.00E+02§	1.00E+02	1.00E+02‡	1.00E+02	2.70E+03‡	2.70E+03
	(5.61E−02)	(3.29E−02)	(5.01E−02)	(4.88E−02)	(5.67E−02)	(3.69E−02)	(5.25E−02)	(5.22E−02)
F27	4.08E+02‡	**4.01E+02**	4.16E+02‡	**4.08E+02**	4.09E+02‡	**4.02E+02**	3.11E+03‡	**3.10E+03**
	(2.09E+00)	(1.43E+00)	(5.79E+00)	(3.60E+00)	(3.07E+00)	(3.36E+00)	(2.94E+00)	(1.52E+00)
F28	**9.22E+02**§	9.42E+02	**8.98E+02**§	9.03E+02	**8.89E+02**§	8.96E+02	**3.98E+03**†	4.09E+03
	(3.70E+01)	(6.03E+01)	(4.45E+01)	(4.38E+01)	(3.55E+01)	(3.47E+01)	(1.53E+02)	(2.39E+02)
F29	1.09E+03‡	**7.15E+02**	1.24E+03‡	**7.42E+02**	1.11E+03‡	**7.16E+02**	3.91E+03‡	**3.64E+03**
	(7.66E+01)	(6.04E−01)	(1.30E+02)	(9.83E+01)	(8.35E+01)	(1.30E+00)	(7.23E+01)	(7.92E+01)
F30	2.93E+03‡	**1.31E+03**	4.36E+03‡	**1.42E+03**	2.95E+03‡	**1.58E+03**	5.23E+03‡	**4.65E+03**
	(8.61E+02)	(2.96E+02)	(1.26E+03)	(4.52E+02)	(7.26E+02)	(2.80E+02)	(5.44E+02)	(4.06E+02)
‡/†/§	17/3/10		18/1/11		18/4/8		19/6/5	

4. Experimental studies

In this section, comprehensive experiments are designed to evaluate the effectiveness of AEL + ABCs. Firstly, AEL + ABCs are tested on 30 benchmark functions with 30*D* from CEC2014 [40], and the comparisons of AEL + ABCs with corresponding ABCs, respectively, are given in detail. Then, AEL + ABCs are further tested on the CEC2014 benchmark functions with 50*D* and 100*D* to study the scalability of AEL + ABCs. Finally, an additional computation burden compared with the corresponding ABCs is discussed. The simulations are carried out on an Intel(R) Core(TM) i7-4790 PC with 3.60GHz CPU and 8GB RAM and Microsoft Windows 7 operating system, and AEL + ABCs are written in Matlab software.

4.1. Experiment setup

AEL + ABCs are firstly tested on CEC2014 test suit with 30 benchmark functions. These functions are classified into four groups: unimodal functions (F1–F3), simple multimodal functions (F4–F16), hybrid functions (F17–F22), and composition functions

Table 6
The statistical results (Mean (Std)) of ABC vs. AEL + ABC, ABCM vs. AEL + ABCM, ABCVSS vs. AEL + ABCVSS and EABC vs. AEL + EABC over 51 independent runs on the CEC2014 benchmarks with 50*D*.

Function	ABC vs.	AEL + ABC	ABCM vs.	AEL + ABCM	ABCVSS vs.	AEL + ABCVSS	EABC vs.	AEL + EABC
F1	1.26E+07‡	**1.20E+05**	1.04E+07‡	**1.76E+05**	1.12E+07‡	**2.88E+05**	1.34E+07‡	**7.47E+05**
	(3.33E+06)	(6.09E+04)	(2.71E+06)	(1.21E+05)	(3.22E+06)	(1.39E+05)	(4.44E+06)	(3.00E+05)
F2	2.77E+03‡	**1.01E−02**	1.90E+03‡	**3.64E−04**	2.97E+03‡	**7.48E−09**	3.77E+03‡	**8.46E+01**
	(2.22E+03)	(6.55E−03)	(9.28E+02)	(5.02E−04)	(1.99E+03)	(2.80E−08)	(4.14E+03)	(2.06E+02)
F3	5.81E+03‡	**2.68E−06**	6.09E+03‡	**8.84E−06**	5.98E+03‡	**1.52E−07**	5.60E+03‡	**1.18E−13**
	(2.55E+03)	(3.95E−06)	(2.17E+03)	(2.01E−05)	(1.44E+03)	(6.65E−07)	(2.20E+03)	(2.23E−14)
F4	7.96E+01‡	**3.60E+01**	6.43E+01‡	**3.07E+01**	7.34E+01‡	**3.38E+01**	1.02E+02‡	**4.54E+01**
	(1.95E+01)	(2.08E+01)	(2.51E+01)	(2.21E+01)	(2.22E+01)	(2.19E+01)	(1.97E+01)	(3.33E+01)
F5	2.03E+01§	2.03E+01	2.02E+01§	2.02E+01	2.03E+01§	2.03E+01	**2.03E+01**§	2.04E+01
	(4.70E−02)	(4.14E−02)	(1.78E−02)	(2.68E−02)	(3.01E−02)	(3.06E−02)	(4.75E−02)	(4.93E−02)
F6	**3.04E+01**§	3.08E+01	3.09E+01§	**3.07E+01**	**2.91E+01**§	2.94E+01	**2.58E+01**§	2.59E+01
	(2.01E+00)	(1.87E+00)	(1.68E+00)	(1.89E+00)	(1.71E+00)	(1.72E+00)	(2.65E+00)	(2.37E+00)
F7	1.56E−02‡	**3.15E−04**	7.52E−03‡	**1.21E−03**	5.65E−03‡	**2.78E−04**	1.26E−02‡	**1.47E−04**
	(6.66E−03)	(3.36E−04)	(4.88E−03)	(9.18E−04)	(3.84E−03)	(5.18E−04)	(1.69E−02)	(1.76E−04)
F8	**4.56E−06**§	1.53E−04	**1.94E−02**§	1.96E−02	**4.73E−13**§	4.84E−13	1.32E−13§	**1.23E−13**
	(1.43E−05)	(8.78E−04)	(1.38E−01)	(1.39E−01)	(6.17E−14)	(6.76E−14)	(4.18E−14)	(3.09E−14)
F9	**1.84E+02**†	2.13E+02	**1.88E+02**†	1.97E+02	**1.23E+02**†	1.40E+02	**8.86E+01**§	9.41E+01
	(2.08E+01)	(2.04E+01)	(1.58E+01)	(1.82E+01)	(1.56E+01)	(1.95E+01)	(1.35E+01)	(1.15E+01)
F10	**5.28E+00**§	5.69E+00	6.05E+00§	**6.01E+00**	3.69E−02§	**1.74E−02**	5.16E−01§	**4.94E−01**
	(1.21E+00)	(1.62E+00)	(1.28E+00)	(1.49E+00)	(1.03E−01)	(1.50E−02)	(4.68E−01)	(5.04E−01)
F11	**4.04E+03**†	4.32E+03	**4.16E+03**§	4.27E+03	**4.00E+03**†	4.28E+03	4.22E+03§	**4.18E+03**
	(4.43E+02)	(3.55E+02)	(3.57E+02)	(3.40E+02)	(3.87E+02)	(3.09E+02)	(6.57E+02)	(5.30E+02)
F12	2.71E−01‡	**2.46E−01**	**1.98E−01**§	2.02E−01	2.35E−01§	**2.32E−01**	3.22E−01§	**3.21E−01**
	(5.03E−02)	(4.11E−02)	(2.75E−02)	(1.76E−02)	(3.06E−02)	(2.90E−02)	(7.02E−02)	(6.78E−02)
F13	**3.20E−01**†	3.37E−01	**3.08E−01**†	3.50E−01	**3.13E−01**†	3.53E−01	**1.65E−01**†	3.28E−01
	(3.36E−02)	(3.15E−02)	(3.45E−02)	(2.99E−02)	(3.14E−02)	(3.70E−02)	(2.74E−02)	(3.63E−02)
F14	**2.36E−01**§	2.30E−01	2.59E−01§	**2.52E−01**	2.77E−01‡	**2.46E−01**	**2.16E−01**†	2.33E−01
	(2.07E−02)	(2.39E−02)	(2.08E−02)	(1.88E−02)	(2.02E−02)	(2.71E−02)	(2.24E−02)	(2.34E−02)
F15	2.23E+01‡	**1.91E+01**	2.17E+01‡	**1.99E+01**	1.56E+01§	**1.50E+01**	1.08E+01§	**1.07E+01**
	(3.26E+00)	(2.48E+00)	(2.21E+00)	(2.28E+00)	(2.14E+00)	(1.64E+00)	(1.74E+00)	(1.66E+00)
F16	1.86E+01§	**1.85E+01**	**1.84E+01**§	1.85E+01	1.80E+01§	1.80E+01	**1.76E+01**§	1.78E+01
	(4.79E−01)	(3.37E−01)	(3.42E−01)	(3.45E−01)	(3.79E−01)	(3.87E−01)	(5.98E−01)	(5.06E−01)
F17	3.02E+06‡	**1.07E+03**	2.80E+06‡	**1.19E+03**	3.22E+06‡	**1.18E+03**	4.60E+06‡	**1.48E+03**
	(1.39E+06)	(2.31E+02)	(1.04E+06)	(2.16E+02)	(1.11E+06)	(1.83E+02)	(1.81E+06)	(4.40E+02)
F18	2.72E+03‡	**4.68E+01**	1.65E+03‡	**5.03E+01**	4.97E+02‡	**4.52E+01**	8.63E+02‡	**5.94E+01**
	(1.62E+03)	(9.96E+00)	(8.08E+02)	(9.29E+00)	(2.88E+02)	(1.38E+01)	(7.75E+02)	(1.42E+01)
F19	1.57E+01‡	**1.46E+01**	1.56E+01‡	**1.44E+01**	1.71E+01‡	**1.42E+01**	2.70E+01‡	**1.33E+01**
	(2.04E+00)	(1.46E+00)	(1.53E+00)	(1.42E+00)	(1.95E+00)	(1.54E+00)	(1.10E+01)	(1.89E+00)
F20	1.66E+04‡	**6.21E+01**	2.29E+04‡	**6.11E+01**	2.34E+04‡	**4.31E+01**	1.32E+04‡	**4.06E+01**
	(4.18E+03)	(1.08E+01)	(6.17E+03)	(1.19E+01)	(6.59E+03)	(9.25E+00)	(4.72E+03)	(9.94E+00)
F21	2.04E+06‡	**6.44E+02**	1.72E+06‡	**6.60E+02**	1.93E+06‡	**5.95E+02**	2.16E+06‡	**6.35E+02**
	(8.63E+05)	(1.27E+02)	(6.33E+05)	(1.28E+02)	(1.02E+06)	(1.24E+02)	(1.16E+06)	(1.46E+02)
F22	6.46E+02§	**6.21E+02**	6.63E+02§	**6.26E+02**	6.65E+02§	**6.25E+02**	7.84E+02‡	**5.50E+02**
	(1.79E+02)	(1.31E+02)	(1.29E+02)	(1.63E+02)	(1.58E+02)	(1.61E+02)	(1.93E+02)	(1.42E+02)
F23	3.45E+02‡	**3.44E+02**	3.45E+02‡	**3.44E+02**	3.46E+02‡	**3.44E+02**	3.48E+02‡	**3.44E+02**
	(8.17E−01)	(7.12E−07)	(5.22E−01)	(6.29E−03)	(1.37E+00)	(1.01E−05)	(2.03E+00)	(1.86E−01)
F24	2.59E+02‡	**2.57E+02**	2.60E+02‡	**2.59E+02**	2.58E+02§	2.58E+02	2.62E+02‡	**2.58E+02**
	(1.24E+00)	(1.03E+00)	(1.66E+00)	(1.56E+00)	(1.46E+00)	(4.97E−01)	(4.55E+00)	(2.64E+00)
F25	2.17E+02‡	**2.09E+02**	2.16E+02‡	**2.09E+02**	2.16E+02‡	**2.07E+02**	2.17E+02‡	**2.06E+02**
	(1.15E+00)	(1.17E+00)	(1.31E+00)	(1.32E+00)	(1.59E+00)	(1.13E+00)	(1.55E+00)	(4.91E−01)
F26	1.00E+02‡	1.00E+02	1.00E+02‡	1.00E+02	1.00E+02‡	1.00E+02	1.00E+02‡	1.00E+02
	(5.17E−02)	(3.98E−02)	(4.82E−02)	(3.79E−02)	(4.62E−02)	(3.71E−02)	(4.57E−02)	(4.34E−02)
F27	5.37E+02‡	**4.39E+02**	7.07E+02‡	**4.53E+02**	9.19E+02‡	**5.30E+02**	8.78E+02‡	**5.65E+02**
	(2.31E+02)	(1.26E+01)	(3.52E+02)	(1.01E+02)	(2.93E+02)	(1.99E+02)	(2.49E+02)	(2.17E+02)
F28	**2.15E+03**†	2.26E+03	**2.34E+03**§	2.38E+03	**1.54E+03**†	1.61E+03	**1.36E+03**†	1.42E+03
	(1.96E+02)	(3.29E+02)	(2.85E+02)	(3.33E+02)	(9.37E+01)	(1.54E+02)	(4.56E+01)	(1.83E+02)
F29	1.59E+03‡	**8.05E+02**	1.53E+03§	**1.46E+03**	1.78E+03‡	**7.96E+02**	2.51E+03‡	**8.55E+02**
	(2.17E+02)	(4.76E+00)	(2.08E+02)	(1.67E+02)	(3.53E+02)	(1.83E+00)	(4.80E+02)	(6.36E+01)
F30	1.03E+04‡	**9.19E+03**	1.01E+04§	1.01E+04	1.03E+04‡	**9.02E+03**	1.05E+04‡	**8.70E+03**
	(7.96E+02)	(4.33E+02)	(7.19E+02)	(6.94E+02)	(8.29E+02)	(3.89E+02)	(9.32E+02)	(3.54E+02)
‡/†/§	19/4/7		16/2/12		17/4/9		18/3/9	

(F23–F30). The detailed description of the 30 benchmark functions can be found in Table 1 [38]. From Table 1, it can be seen that these benchmark functions are non-separable.

In order to evaluate the performance of AEL + ABCs, the AEL is combined with eight existing ABC algorithms: ABC, ABCM, ABCVSS, EABC, GABC, MABC, OPIABC and qABC. They are termed as AEL + ABC, AEL + ABCM, AEL + ABCVSS, AEL + EABC, AEL + GABC, AEL + MABC, AEL + OPIABC, and AEL + qABC, respectively. The public and private parameter settings of these algorithms are listed in Tables 2 and 3, respectively. In addition, three criteria for evaluation are used to measure the performance of each algorithm.

- Error measure: As an evaluation indicator of calculation accuracy, the function error value $f(\mathbf{x}) - f(\mathbf{x}^*)$ for solution $\mathbf{x}$ is defined, where $\mathbf{x}^*$ is the global optimal solution of the corresponding test problem. The best function error value of each run is recorded when the maximum number of function evaluations is reached. The mean and standard deviation of the best function error values are considered for performance comparison.

Table 7
The statistical results (Mean (Std)) of GABC vs. AEL + GABC, MABC vs. AEL + MABC, OPIABC vs. AEL + OPIABC and qABC vs. AEL + qABC over 51 independent runs on the CEC2014 benchmarks with 50*D*.

Function	GABC vs.	AEL + GABC	MABC vs.	AEL + MABC	OPIABC vs.	AEL + OPIABC	qABC vs.	AEL + qABC
F1	1.16E+07‡	**4.36E+05**	1.90E+07‡	**5.08E+05**	1.13E+07‡	**3.88E+05**	5.07E+06‡	**1.21E+04**
	(3.14E+06)	(1.87E+05)	(4.50E+06)	(2.12E+05)	(3.20E+06)	(1.72E+05)	(1.30E+06)	(1.05E+04)
F2	4.03E+03‡	**5.98E−06**	1.76E+04‡	**2.01E+01**	1.69E+03‡	**4.21E+00**	1.05E+03‡	**3.21E+02**
	(3.94E+03)	(4.10E−06)	(9.69E+03)	(5.37E+01)	(1.53E+03)	(1.03E+01)	(7.36E+02)	(1.91E+02)
F3	5.88E+03‡	**1.11E−05**	7.00E+03‡	**3.56E−07**	5.51E+03‡	**8.08E−06**	6.17E+03‡	**3.00E+02**
	(1.87E+03)	(2.90E−05)	(3.42E+03)	(2.31E−06)	(1.87E+03)	(1.68E−05)	(2.28E+03)	(1.97E−08)
F4	6.98E+01‡	**3.35E+01**	8.24E+01‡	**2.80E+01**	8.00E+01‡	**2.45E+01**	4.65E+02‡	**4.40E+02**
	(2.21E+01)	(2.43E+01)	(2.10E+01)	(1.88E+01)	(2.12E+01)	(1.59E+01)	(2.79E+01)	(2.86E+01)
F5	2.03E+01§	2.03E+01	2.03E+01§	2.03E+01	2.04E+01§	2.04E+01	5.20E+02‡	5.20E+02
	(4.26E−02)	(5.30E−02)	(3.29E−02)	(3.40E−02)	(5.07E−02)	(5.63E−02)	(2.66E−03)	(1.98E−03)
F6	2.87E+01§	**2.86E+01**	3.03E+01§	**2.96E+01**	**2.82E+01**§	2.86E+01	**6.27E+02**†	6.31E+02
	(1.95E+00)	(1.94E+00)	(1.88E+00)	(1.76E+00)	(2.01E+00)	(1.77E+00)	(2.48E+00)	(2.67E+00)
F7	5.91E−03‡	**1.50E−04**	4.88E−03‡	**3.48E−04**	2.75E−03‡	**8.10E−05**	7.00E+02‡	7.00E+02
	(3.59E−03)	(2.33E−04)	(2.57E−03)	(7.10E−04)	(1.35E−03)	(1.35E−04)	(8.63E−03)	(4.83E−03)
F8	4.15E−13§	**4.06E−13**	**5.13E−13**§	5.26E−13	**1.63E−13**§	1.81E−13	8.00E+02‡	8.00E+02
	(6.75E−14)	(6.12E−14)	(8.00E−14)	(5.55E−14)	(5.69E−14)	(5.65E−14)	(1.08E−10)	(7.46E−11)
F9	**1.34E+02**§	1.43E+02	**1.20E+02**§	1.23E+02	**1.33E+02**§	1.36E+02	**1.08E+03**†	1.10E+03
	(1.33E+01)	(1.34E+01)	(1.33E+01)	(1.42E+01)	(1.39E+01)	(1.14E+01)	(2.17E+01)	(2.09E+01)
F10	**3.06E+00**†	3.29E+00	**9.24E−02**§	1.10E−01	2.66E−02§	**2.14E−02**	1.00E+03§	1.00E+03
	(1.03E+00)	(1.19E+00)	(1.30E−01)	(2.25E−01)	(1.81E−02)	(1.36E−02)	(1.96E−02)	(2.11E−02)
F11	**3.95E+03**†	4.04E+03	**4.05E+03**§	4.16E+03	4.16E+03§	**4.13E+03**	**5.11E+03**†	5.32E+03
	(3.20E+02)	(3.68E+02)	(3.40E+02)	(3.37E+02)	(3.45E+02)	(3.73E+02)	(4.17E+02)	(3.50E+02)
F12	2.55E−01§	2.55E−01	2.41E−01§	**2.31E−01**	**3.30E−01**§	3.43E−01	1.20E+03†	1.20E+03
	(5.00E−02)	(4.76E−02)	(3.42E−02)	(3.50E−02)	(5.04E−02)	(5.66E−02)	(2.76E−02)	(3.71E−02)
F13	**2.71E−01**§	3.27E−01	**3.02E−01**†	3.56E−01	**3.05E−01**†	3.66E−01	1.30E+03†	1.30E+03
	(2.95E−02)	(2.44E−02)	(3.23E−02)	(4.38E−02)	(3.07E−02)	(3.63E−02)	(2.88E−02)	(3.20E−02)
F14	2.15E−01‡	**1.98E−01**	2.93E−01‡	**2.42E−01**	2.52E−01‡	**2.41E−01**	1.40E+03‡	1.40E+03
	(1.74E−02)	(2.92E−02)	(2.80E−02)	(2.62E−02)	(2.01E−02)	(2.20E−02)	(2.13E−02)	(1.77E−02)
F15	1.65E+01‡	**1.55E+01**	1.53E+01§	**1.48E+01**	1.59E+01§	**1.58E+01**	1.52E+03‡	1.52E+03
	(1.91E+00)	(1.97E+00)	(2.08E+00)	(1.58E+00)	(2.19E+00)	(1.46E+00)	(3.29E+00)	(3.22E+00)
F16	**1.79E+01**§	1.80E+01	1.81E+01§	**1.80E+01**	1.85E+01§	**1.84E+01**	1.62E+03‡	1.62E+03
	(4.88E−01)	(4.40E−01)	(3.03E−01)	(3.38E−01)	(4.79E−01)	(4.81E−01)	(5.61E−01)	(5.84E−01)
F17	3.64E+06‡	**1.38E+03**	7.04E+06‡	**1.22E+03**	3.34E+06‡	**1.25E+03**	1.71E+06‡	**2.87E+03**
	(1.12E+06)	(2.80E+02)	(2.69E+06)	(2.31E+02)	(1.07E+06)	(2.46E+02)	(8.49E+05)	(2.98E+02)
F18	1.92E+03‡	**6.63E+01**	5.91E+03‡	**6.75E+01**	1.30E+03‡	**4.24E+01**	2.69E+03‡	**1.87E+03**
	(1.76E+03)	(1.89E+01)	(3.45E+03)	(1.88E+01)	(8.86E+02)	(9.11E+00)	(8.19E+02)	(1.49E+01)
F19	1.64E+01‡	**1.42E+01**	1.98E+01‡	**1.44E+01**	1.64E+01‡	**1.40E+01**	1.91E+03§	1.91E+03
	(2.26E+00)	(1.43E+00)	(2.09E+00)	(1.45E+00)	(1.92E+00)	(1.22E+00)	(1.96E+00)	(1.40E+00)
F20	1.52E+04‡	**5.24E+01**	3.43E+04‡	**5.60E+01**	2.03E+04‡	**5.40E+01**	2.41E+04‡	**2.06E+03**
	(4.34E+03)	(1.27E+01)	(8.89E+03)	(1.28E+01)	(5.66E+03)	(1.16E+01)	(6.80E+03)	(1.58E+01)
F21	1.87E+06‡	**6.13E+02**	4.09E+06‡	**6.78E+02**	1.90E+06‡	**5.82E+02**	1.33E+06‡	**2.83E+03**
	(6.93E+05)	(1.82E+02)	(1.63E+06)	(2.17E+02)	(7.09E+05)	(1.36E+02)	(6.63E+05)	(1.95E+02)
F22	6.33E+02§	**5.81E+02**	6.91E+02‡	**6.18E+02**	6.32E+02§	**5.92E+02**	2.87E+03§	2.87E+03
	(1.39E+02)	(1.42E+02)	(1.75E+02)	(1.36E+02)	(1.29E+02)	(1.21E+02)	(1.71E+02)	(1.57E+02)
F23	3.45E+02‡	**3.44E+02**	3.46E+02‡	**3.44E+02**	3.45E+02‡	**3.44E+02**	2.64E+03‡	2.64E+03
	(6.26E−01)	(4.29E−06)	(1.35E+00)	(2.21E−05)	(5.25E−01)	(5.31E−04)	(2.92E−02)	(1.73E−04)
F24	2.58E+02§	2.58E+02	2.58E+02‡	**2.57E+02**	2.58E+02‡	**2.57E+02**	2.66E+03§	2.66E+03
	(1.09E+00)	(5.46E−01)	(2.23E+00)	(4.89E−01)	(1.16E+00)	(5.18E−01)	(1.32E+00)	(1.95E+00)
F25	2.17E+02‡	**2.07E+02**	2.17E+02‡	**2.07E+02**	2.16E+02‡	**2.07E+02**	2.71E+03‡	2.71E+03
	(1.20E+00)	(7.23E−01)	(1.77E+00)	(8.44E−01)	(1.24E+00)	(8.71E−01)	(1.65E+00)	(8.41E−01)
F26	1.00E+02‡	1.00E+02	1.00E+02§	1.00E+02	1.00E+02‡	1.00E+02	2.70E+03‡	2.70E+03
	(4.43E−02)	(3.48E−02)	(4.29E−02)	(3.45E−02)	(4.76E−02)	(4.48E−02)	(5.02E−02)	(4.54E−02)
F27	7.96E+02‡	5.11E+02	9.76E+02‡	6.79E+02	6.79E+02‡	5.04E+02	3.28E+03‡	**3.13E+03**
	(3.29E+02)	(2.00E+02)	(2.78E+02)	(2.87E+02)	(2.64E+02)	(1.52E+02)	(2.66E+02)	(1.21E+01)
F28	1.59E+03§	**1.58E+03**	**1.42E+03**§	1.44E+03	**1.52E+03**†	1.55E+03	**4.99E+03**†	5.48E+03
	(1.04E+02)	(1.07E+02)	(6.27E+01)	(6.56E+01)	(7.72E+01)	(6.73E+01)	(2.72E+02)	(3.68E+02)
F29	1.62E+03‡	**7.97E+02**	2.25E+03‡	**1.02E+03**	1.69E+03‡	**8.00E+02**	4.22E+03‡	**4.07E+03**
	(2.54E+02)	(1.87E+00)	(2.87E+02)	(4.75E+02)	(1.66E+02)	(3.09E+00)	(2.38E+02)	(1.82E+02)
F30	1.01E+04‡	**9.17E+03**	1.15E+04‡	**8.95E+03**	9.96E+03‡	**9.11E+03**	1.30E+04§	**1.29E+04**
	(6.97E+02)	(3.81E+02)	(1.09E+03)	(3.79E+02)	(6.44E+02)	(3.95E+02)	(8.60E+02)	(7.84E+02)
‡/†/§	18/2/10		18/1/11		18/2/10		17/8/5	

- Statistics by Wilcoxon test: To identify the significance of difference between two algorithms on single problem in terms of the mean function error values, the statistical tool-Wilcoxon signed-rank test at a 0.05 significance level is conduced. The test result is denoted as "‡/†/§", which means that one algorithm is significance better than, worse than, and similar to its competitor, respectively. To be clear, the significantly better results in terms of the mean function error values between AEL + ABCs and their competitors are highlighted in boldface in tables, and the comparison results are summarized in the last row of each table. To identify differences between pair of algorithms on all problems, the multiproblem Wilcoxon signed-rank test, conduced by the KEEL software [41], is carried out.
- Convergence: The convergence curves for some different types of test functions are plotted to show the mean error performance of the best solution over the total run in the respective experiments.

Table 8
The statistical results (Mean (Std)) of ABC vs. AEL + ABC, ABCM vs. AEL + ABCM, ABCVSS vs. AEL + ABCVSS and EABC vs. AEL + EABC over 51 independent runs on the CEC2014 benchmarks with 100*D*.

Function	ABC vs.	AEL + ABC	ABCM vs.	AEL + ABCM	ABCVSS vs.	AEL + ABCVSS	EABC vs.	AEL + EABC
F1	7.17E+07‡	**7.77E+05**	6.68E+07‡	**9.05E+05**	7.48E+07‡	**1.27E+06**	7.60E+07‡	**2.92E+06**
	(1.37E+07)	(3.03E+05)	(1.33E+07)	(3.44E+05)	(1.43E+07)	(4.70E+05)	(1.79E+07)	(1.08E+06)
F2	6.05E+04‡	**1.04E+04**	9.33E+04‡	**4.81E+03**	3.08E+04‡	**2.35E+00**	1.83E+04‡	**3.09E−02**
	(5.24E+04)	(1.05E+04)	(5.63E+04)	(5.94E+03)	(1.92E+04)	(2.76E+00)	(2.48E+04)	(1.62E−01)
F3	1.40E+04‡	**1.90E−02**	1.50E+04‡	**9.47E−03**	1.44E+04‡	**1.80E−04**	1.48E+04‡	**1.03E−08**
	(3.11E+03)	(6.25E−03)	(3.84E+03)	(3.67E−03)	(4.02E+03)	(1.32E−04)	(3.77E+03)	(3.73E−09)
F4	2.73E+02‡	**1.97E+02**	2.66E+02‡	**1.95E+02**	2.41E+02‡	**1.74E+02**	2.53E+02‡	**1.77E+02**
	(2.14E+01)	(1.68E+01)	(2.14E+01)	(2.30E+01)	(2.43E+01)	(1.85E+01)	(3.63E+01)	(2.82E+01)
F5	2.06E+01‡	2.06E+01	2.05E+01§	2.05E+01	2.06E+01§	2.06E+01	2.07E+01§	2.07E+01
	(4.48E−02)	(4.11E−02)	(2.41E−02)	(2.53E−02)	(2.28E−02)	(3.13E−02)	(4.18E−02)	(3.00E−02)
F6	**8.56E+01**§	8.60E+01	**8.57E+01**§	8.60E+01	**8.06E+01**§	8.14E+01	**7.84E+01**§	7.86E+01
	(2.93E+00)	(2.64E+00)	(2.34E+00)	(2.94E+00)	(2.69E+00)	(3.02E+00)	(3.16E+00)	(3.47E+00)
F7	**1.44E−01**†	1.59E−01	1.22E−01‡	**1.08E−01**	9.51E−03‡	**3.44E−03**	3.00E−02‡	**1.57E−03**
	(3.61E−02)	(2.41E−02)	(3.59E−02)	(2.04E−02)	(3.67E−03)	(2.63E−03)	(3.72E−02)	(1.54E−03)
F8	**8.85E+00**§	9.27E+00	**1.18E+01**§	1.25E+01	**1.27E−12**†	1.61E−12	**3.95E−13**†	4.24E−13
	(1.96E+00)	(1.83E+00)	(2.08E+00)	(2.27E+00)	(1.58E−13)	(9.81E−13)	(6.95E−14)	(5.61E−14)
F9	**6.82E+02**†	7.17E+02	**6.85E+02**†	7.15E+02	**4.76E+02**†	5.08E+02	**3.59E+02**§	3.63E+02
	(3.68E+01)	(4.49E+01)	(4.41E+01)	(3.66E+01)	(3.57E+01)	(4.38E+01)	(3.16E+01)	(3.48E+01)
F10	**1.68E+02**§	1.88E+02	**1.88E+02**†	2.31E+02	**7.11E−03**†	1.67E−02	1.13E+00§	**9.97E−01**
	(8.35E+01)	(8.18E+01)	(9.67E+01)	(8.88E+01)	(7.34E−03)	(1.61E−02)	(6.30E−01)	(6.83E−01)
F11	1.19E+04§	**1.18E+04**	**1.16E+04**†	1.18E+04	**1.16E+04**†	1.19E+04	1.27E+04§	**1.25E+04**
	(5.79E+02)	(5.53E+02)	(4.95E+02)	(4.82E+02)	(6.47E+02)	(6.13E+02)	(8.45E+02)	(8.74E+02)
F12	5.09E−01‡	**4.57E−01**	4.05E−01§	**4.03E−01**	4.56E−01§	**4.47E−01**	**5.72E−01**†	6.12E−01
	(7.61E−02)	(4.70E−02)	(2.53E−02)	(3.50E−02)	(4.07E−02)	(3.52E−02)	(7.40E−02)	(8.01E−02)
F13	**3.56E−01**†	3.89E−01	**3.30E−01**†	3.68E−01	**3.35E−01**†	3.97E−01	**2.38E−01**†	3.90E−01
	(2.88E−02)	(2.24E−02)	(2.46E−02)	(2.01E−02)	(2.62E−02)	(2.63E−02)	(2.03E−02)	(2.81E−02)
F14	2.79E−01§	**2.72E−01**	**2.78E−01**§	2.84E−01	2.97E−01‡	**2.72E−01**	**2.65E−01**§	2.71E−01
	(2.24E−02)	(1.81E−02)	(1.93E−02)	(1.34E−02)	(1.87E−02)	(1.58E−02)	(1.58E−02)	(1.52E−02)
F15	7.57E+01‡	**7.18E+01**	7.79E+01‡	**7.39E+01**	5.88E+01‡	**5.66E+01**	**3.99E+01**†	4.23E+01
	(6.16E+00)	(4.89E+00)	(5.61E+00)	(4.44E+00)	(4.27E+00)	(3.89E+00)	(3.99E+00)	(4.88E+00)
F16	4.17E+01‡	**4.14E+01**	**4.13E+01**§	4.14E+01	**4.07E+01**†	4.09E+01	4.06E+01§	**4.05E+01**
	(5.38E−01)	(4.07E−01)	(4.71E−01)	(3.88E−01)	(5.26E−01)	(3.94E−01)	(6.40E−01)	(4.62E−01)
F17	1.70E+07‡	**4.03E+03**	1.69E+07‡	**4.03E+03**	1.99E+07‡	**4.38E+03**	2.16E+07‡	**6.23E+03**
	(5.02E+06)	(4.68E+02)	(3.73E+06)	(4.79E+02)	(5.03E+06)	(5.95E+02)	(6.34E+06)	(2.27E+03)
F18	1.64E+04‡	**2.19E+02**	7.34E+03‡	**1.90E+02**	8.53E+02‡	**1.51E+02**	8.08E+02‡	**1.66E+02**
	(1.75E+04)	(2.24E+01)	(6.25E+03)	(1.76E+01)	(4.18E+02)	(3.34E+01)	(8.74E+02)	(2.99E+01)
F19	8.64E+01‡	**6.19E+01**	8.24E+01‡	**6.33E+01**	8.25E+01‡	**5.59E+01**	1.03E+02‡	**5.49E+01**
	(9.80E+00)	(1.61E+01)	(1.09E+01)	(1.63E+01)	(5.87E+00)	(1.62E+01)	(1.32E+01)	(1.68E+01)
F20	7.10E+04‡	**2.10E+02**	7.95E+04‡	**2.11E+02**	7.90E+04‡	**1.51E+02**	6.30E+04‡	**1.39E+02**
	(1.18E+04)	(2.92E+01)	(1.23E+04)	(2.72E+01)	(1.23E+04)	(1.90E+01)	(1.02E+04)	(2.02E+01)
F21	1.20E+07‡	**1.80E+03**	1.03E+07‡	**1.91E+03**	1.30E+07‡	**1.66E+03**	1.37E+07‡	**1.65E+03**
	(3.07E+06)	(2.37E+02)	(2.38E+06)	(3.17E+02)	(3.06E+06)	(2.53E+02)	(4.43E+06)	(2.35E+02)
F22	1.95E+03‡	**1.56E+03**	1.85E+03‡	**1.60E+03**	1.96E+03‡	**1.50E+03**	2.19E+03‡	**1.59E+03**
	(2.48E+02)	(2.21E+02)	(1.88E+02)	(2.16E+02)	(2.32E+02)	(2.56E+02)	(3.22E+02)	(2.23E+02)
F23	3.51E+02‡	**3.48E+02**	3.50E+02‡	**3.49E+02**	3.50E+02‡	**3.48E+02**	3.51E+02‡	**3.48E+02**
	(8.83E−01)	(5.17E−03)	(7.37E−01)	(1.77E−01)	(6.34E−01)	(6.17E−05)	(1.64E+00)	(2.66E−02)
F24	3.66E+02§	**3.65E+02**	3.67E+02§	3.67E+02	3.59E+02‡	**3.57E+02**	3.64E+02‡	**3.60E+02**
	(1.13E+00)	(1.31E+00)	(1.17E+00)	(1.31E+00)	(1.70E+00)	(1.10E+00)	(1.95E+00)	(1.90E+00)
F25	2.59E+02‡	**2.52E+02**	2.58E+02‡	**2.52E+02**	2.59E+02‡	**2.41E+02**	2.62E+02‡	**2.27E+02**
	(3.03E+00)	(6.25E+00)	(3.95E+00)	(6.78E+00)	(4.15E+00)	(6.60E+00)	(4.45E+00)	(3.75E+00)
F26	1.48E+02‡	**1.06E+02**	1.25E+02§	**1.04E+02**	1.32E+02‡	**1.16E+02**	**1.55E+02**§	1.79E+02
	(5.29E+01)	(1.94E+01)	(4.47E+01)	(1.16E+01)	(4.85E+01)	(3.64E+01)	(5.34E+01)	(4.14E+01)
F27	1.02E+03‡	**5.60E+02**	1.42E+03‡	**5.44E+02**	1.88E+03‡	**1.05E+03**	1.72E+03‡	**1.42E+03**
	(8.00E+02)	(1.04E+02)	(9.51E+02)	(3.97E+01)	(7.47E+02)	(7.58E+02)	(6.47E+02)	(7.56E+02)
F28	**6.95E+03**†	7.25E+03	**7.08E+03**‡	7.28E+03	**4.07E+03**†	4.75E+03	**3.26E+03**†	3.54E+03
	(7.18E+02)	(6.76E+02)	(7.71E+02)	(7.42E+02)	(4.83E+02)	(6.29E+02)	(2.19E+02)	(5.41E+02)
F29	4.85E+03‡	**3.17E+03**	4.64E+03§	**3.67E+03**	3.26E+03‡	**8.20E+02**	1.05E+04‡	**1.92E+03**
	(1.07E+03)	(8.18E+02)	(9.69E+02)	(9.96E+02)	(8.05E+02)	(7.39E+01)	(4.47E+03)	(3.81E+02)
F30	4.48E+04‡	**8.42E+03**	4.33E+04‡	**1.60E+04**	5.19E+04‡	**8.71E+03**	5.59E+04‡	**9.06E+03**
	(1.16E+04)	(8.77E+02)	(1.02E+04)	(6.08E+03)	(1.13E+04)	(1.15E+03)	(1.01E+04)	(8.53E+02)
‡/†/§	20/4/6		17/4/9		20/7/3		17/5/8	

4.2. Benchmark comparisons

To evaluate the contribution of the proposed AEL on ABCs, eight groups of experiments (i.e., ABC vs. AEL + ABC, ABCM vs. AEL + ABCM, ABCVSS vs. AEL + ABCVSS, EABC vs. AEL + EABC, GABC vs. AEL + GABC, MABC vs. AEL + MABC, OPIABC vs. AEL + OPIABC and qABC vs. AEL + qABC) are designed. Tables 3 and 4 give the statistics summarizing the performance comparisons. The results in row ‡/†/§ show that AEL + ABCs significantly outperform the corresponding ABCs algorithm on most of benchmark problems. Specifically, AEL + ABC, AEL + ABCM, AEL + ABCVSS, AEL + EABC, AEL + GABC, AEL + MABC, AEL + OPIABC and AEL + qABC significantly outperform ABC, ABCM, ABCVSS, EABC, GABC, MABC, OPIABC and qABC on 19, 17, 17, 18, 17, 18, 18 and 19 out of 30 benchmark problems, respectively. By contrast, ABC, ABCM, ABCVSS, EABC, GABC, MABC, OPIABC and qABC significantly outperform AEL + ABC, AEL + ABCM, AEL + ABCVSS, AEL + EABC, AEL + GABC, AEL + MABC, AEL + OPIABC and AEL + qABC only on 4, 1, 4, 2, 3, 1, 4 and 6 out of 30 benchmark problems, respectively. In additional, the statistical results of multiproblem Wilcoxon signed-rank test summarized

Table 9
The statistical results (Mean (Std)) of GABC vs. AEL + GABC, MABC vs. AEL + MABC, OPIABC vs. AEL + OPIABC and qABC vs. AEL + qABC over 51 independent runs on the CEC2014 benchmarks with 100*D*.

Function	GABC vs.	AEL + GABC	MABC vs.	AEL + MABC	OPIABC vs.	AEL + OPIABC	qABC vs.	AEL + qABC
F1	7.12E+07‡	**1.52E+06**	1.16E+08‡	**2.77E+06**	7.90E+07‡	**2.07E+06**	2.27E+07‡	**1.81E+05**
	(1.47E+07)	(6.29E+05)	(1.74E+07)	(8.95E+05)	(1.28E+07)	(8.71E+05)	(6.19E+06)	(7.10E+04)
F2	3.28E+04‡	**5.54E+01**	6.12E+05‡	**1.00E+01**	2.81E+04‡	**1.81E+02**	**7.17E+03**§	1.11E+04
	(2.06E+04)	(6.21E+01)	(3.51E+05)	(1.80E+01)	(1.17E+04)	(3.49E+02)	(5.74E+03)	(1.05E+04)
F3	1.45E+04‡	**8.75E−04**	1.96E+04‡	**6.02E−06**	1.48E+04‡	**8.44E−04**	1.34E+04‡	**3.00E+02**
	(3.65E+03)	(2.29E−04)	(4.78E+03)	(1.87E−06)	(3.86E+03)	(8.86E−04)	(3.30E+03)	(1.35E−05)
F4	2.43E+02‡	**1.76E+02**	2.41E+02‡	**1.75E+02**	2.54E+02‡	**1.67E+02**	5.91E+02§	**5.87E+02**
	(2.49E+01)	(2.06E+01)	(2.21E+01)	(1.86E+01)	(2.05E+01)	(1.39E+01)	(3.56E+01)	(3.42E+01)
F5	2.06E+01§	2.06E+01	2.06E+01§	2.06E+01	2.07E+01§	2.07E+01	5.20E+02§	5.20E+02
	(4.63E−02)	(4.53E−02)	(3.64E−02)	(3.12E−02)	(3.95E−02)	(3.54E−02)	(2.84E−03)	(3.04E−03)
F6	**7.92E+01**†	8.10E+01	8.20E+01§	**8.11E+01**	**8.12E+01**†	8.24E+01	**6.79E+02**†	6.81E+02
	(3.23E+00)	(2.06E+00)	(2.69E+00)	(3.72E+00)	(2.16E+00)	(1.90E+00)	(4.00E+00)	(4.17E+00)
F7	**1.35E−02**†	1.74E−02	1.01E−02‡	**2.05E−03**	7.67E−03‡	**2.31E−03**	7.00E+02‡	7.00E+02
	(4.25E−03)	(4.66E−03)	(3.80E−03)	(2.78E−03)	(2.19E−03)	(1.59E−03)	(1.16E−02)	(4.94E−03)
F8	**2.27E+00**†	2.52E+00	**1.42E−12**§	1.46E−12	**1.31E−12**†	1.35E−12	8.00E+02§	8.00E+02
	(6.02E−01)	(7.67E−01)	(9.74E−14)	(9.10E−14)	(7.32E−14)	(8.58E−14)	(5.34E−08)	(6.57E−08)
F9	**5.06E+02**†	5.26E+02	**4.34E+02**†	4.50E+02	**5.10E+02**†	5.26E+02	**1.55E+03**†	1.60E+03
	(3.41E+01)	(3.50E+01)	(2.60E+01)	(2.56E+01)	(3.63E+01)	(3.29E+01)	(5.46E+01)	(4.49E+01)
F10	**1.50E+01**§	1.53E+01	4.26E−02§	**4.10E−02**	6.96E−03§	**5.81E−03**	1.00E+03§	1.00E+03
	(3.44E+00)	(3.07E+00)	(1.67E−02)	(1.63E−02)	(5.76E−03)	(5.15E−03)	(1.69E−02)	(2.01E−02)
F11	**1.16E+04**§	1.17E+04	**1.21E+04**§	1.22E+04	**1.20E+04**§	1.22E+04	**1.23E+04**§	1.25E+04
	(5.85E+02)	(5.98E+02)	(6.23E+02)	(6.28E+02)	(6.19E+02)	(5.27E+02)	(8.63E+02)	(7.66E+02)
F12	**4.78E−01**§	4.82E−01	**4.48E−01**§	4.55E−01	6.15E−01§	**6.09E−01**	1.20E+03†	1.20E+03
	(5.26E−02)	(6.53E−02)	(4.37E−02)	(4.60E−02)	(7.40E−02)	(6.13E−02)	(3.39E−02)	(6.67E−02)
F13	**3.42E−01**†	4.00E−01	**3.63E−01**†	4.10E−01	**3.51E−01**†	4.09E−01	1.30E+03†	1.30E+03
	(2.44E−02)	(2.92E−02)	(2.34E−02)	(2.63E−02)	(2.59E−02)	(2.84E−02)	(2.22E−02)	(2.48E−02)
F14	2.69E−01‡	**2.52E−01**	3.12E−01‡	**2.73E−01**	2.97E−01‡	**2.84E−01**	1.40E+03‡	1.40E+03
	(1.87E−02)	(2.13E−02)	(2.56E−02)	(1.98E−02)	(2.29E−02)	(2.35E−02)	(1.12E−02)	(1.08E−02)
F15	6.23E+01§	**6.10E+01**	5.92E+01‡	**5.64E+01**	6.16E+01‡	**5.96E+01**	**1.57E+03**†	1.58E+03
	(4.67E+00)	(3.83E+00)	(5.20E+00)	(3.59E+00)	(4.14E+00)	(3.76E+00)	(1.31E+01)	(1.64E+01)
F16	**4.08E+01**§	4.09E+01	4.10E+01§	4.10E+01	4.16E+01§	4.16E+01	1.64E+03†	1.64E+03
	(4.91E−01)	(4.89E−01)	(4.83E−01)	(3.99E−01)	(3.36E−01)	(3.46E−01)	(8.13E−01)	(6.57E−01)
F17	1.84E+07‡	**4.51E+03**	3.39E+07‡	**4.89E+03**	1.74E+07‡	**4.84E+03**	5.33E+06‡	**5.33E+03**
	(4.73E+06)	(5.14E+02)	(6.94E+06)	(5.31E+02)	(4.36E+06)	(6.49E+02)	(1.71E+06)	(7.24E+02)
F18	8.70E+03‡	**2.17E+02**	1.24E+05‡	**1.82E+02**	5.03E+03‡	**1.57E+02**	3.13E+03‡	**2.15E+03**
	(1.39E+04)	(4.68E+01)	(7.33E+04)	(3.00E+01)	(3.35E+03)	(1.51E+01)	(1.05E+03)	(7.07E+01)
F19	8.19E+01‡	**5.70E+01**	9.57E+01‡	**6.35E+01**	7.98E+01‡	**5.94E+01**	1.97E+03‡	**1.96E+03**
	(8.73E+00)	(1.71E+01)	(1.16E+01)	(1.45E+01)	(8.89E+00)	(1.62E+01)	(1.85E+01)	(1.66E+01)
F20	6.67E+04‡	**1.89E+02**	1.06E+05‡	**1.88E+02**	7.10E+04‡	**1.85E+02**	8.08E+04‡	**2.21E+03**
	(1.41E+04)	(2.72E+01)	(1.58E+04)	(2.66E+01)	(1.19E+04)	(2.46E+01)	(1.33E+04)	(4.65E+01)
F21	1.14E+07‡	**1.75E+03**	2.31E+07‡	**2.57E+03**	1.24E+07‡	**1.79E+03**	4.11E+06‡	**4.20E+03**
	(2.44E+06)	(2.54E+02)	(5.47E+06)	(9.33E+02)	(2.95E+06)	(2.62E+02)	(1.31E+06)	(5.05E+02)
F22	1.87E+03‡	**1.59E+03**	2.11E+03‡	**1.61E+03**	1.76E+03‡	**1.58E+03**	3.94E+03‡	**3.85E+03**
	(2.27E+02)	(2.08E+02)	(2.75E+02)	(2.45E+02)	(1.94E+02)	(2.55E+02)	(2.43E+02)	(2.09E+02)
F23	3.49E+02‡	**3.48E+02**	3.50E+02‡	**3.48E+02**	3.49E+02‡	**3.48E+02**	2.65E+03‡	2.65E+03
	(5.47E−01)	(3.27E−04)	(7.20E−01)	(6.04E−06)	(4.21E−01)	(1.07E−04)	(3.24E−01)	(1.49E−03)
F24	3.61E+02‡	**3.59E+02**	3.61E+02‡	**3.58E+02**	3.58E+02‡	**3.56E+02**	2.75E+03§	2.75E+03
	(1.56E+00)	(1.24E+00)	(1.14E+00)	(1.79E+00)	(1.26E+00)	(5.56E−01)	(9.72E+00)	(1.00E+01)
F25	2.58E+02‡	**2.39E+02**	2.59E+02‡	**2.36E+02**	2.59E+02‡	**2.35E+02**	2.75E+03‡	**2.74E+03**
	(3.52E+00)	(5.75E+00)	(4.06E+00)	(4.02E+00)	(2.91E+00)	(3.80E+00)	(5.73E+00)	(6.14E+00)
F26	1.43E+02‡	**1.22E+02**	**1.03E+02**†	1.14E+02	1.47E+02‡	**1.34E+02**	2.74E+03§	**2.73E+03**
	(5.19E+01)	(4.13E+01)	(1.53E+01)	(3.28E+01)	(5.33E+01)	(4.72E+01)	(4.98E+01)	(4.47E+01)
F27	1.80E+03‡	**9.63E+02**	2.16E+03‡	**1.92E+03**	1.36E+03‡	**8.52E+02**	4.08E+03‡	**3.34E+03**
	(7.90E+02)	(7.52E+02)	(6.07E+02)	(7.30E+02)	(8.71E+02)	(6.29E+02)	(8.75E+02)	(4.20E+02)
F28	**4.72E+03**†	4.97E+03	**3.25E+03**†	3.53E+03	**4.06E+03**†	4.29E+03	**9.54E+03**†	1.01E+04
	(6.04E+02)	(6.71E+02)	(2.82E+02)	(3.80E+02)	(3.68E+02)	(5.13E+02)	(7.36E+02)	(8.58E+02)
F29	2.74E+03‡	**1.25E+03**	5.52E+03‡	**2.65E+03**	2.88E+03‡	**1.03E+03**	5.97E+03‡	**5.14E+03**
	(4.69E+02)	(3.70E+02)	(9.11E+02)	(1.14E+03)	(5.44E+02)	(1.40E+02)	(7.91E+02)	(2.88E+02)
F30	4.24E+04‡	**8.11E+03**	7.22E+04‡	**8.59E+03**	4.29E+04‡	**8.83E+03**	2.77E+04‡	**1.15E+04**
	(1.01E+04)	(8.19E+02)	(1.45E+04)	(6.74E+02)	(8.91E+03)	(7.06E+02)	(4.79E+03)	(1.08E+03)
‡/†/§	18/6/6		19/4/7		20/5/5		15/7/8	

in Table 9 indicate that AEL + ABCs obtain higher R^{+} values than R^{-} values in all cases. Further, the *p*-values of all cases are less than 0.05, which shows that AEL + ABCs are significantly better than the corresponding ABCs.

With respect to the characteristics of benchmark problems, a close inspection of the numerical values shows that AEL + ABCs can find better results for most of cases in different types of test functions. In the case of unimodal functions (F1–F3), AEL + ABCs greatly improve the performance of ABCs, and can obtain at least one order of magnitude improvement for F1–F3. In the case of other three kinds of multimodal functions (F4–F16, F17–F22 and F23–F30), AEL + ABCs also obtain better performance on most of benchmark problems. Since AEL + ABCs consider both fitness landscape and diversity information, a better balance between exploration and exploitation can be achieved. Therefore, AEL + ABCs can obtain better results for most of cases in unimodal functions, simple multimodal functions, hybrid functions, and composition functions.

In additional, the convergence curves of four different types of test functions (unimodal function-F1, simple multimodal function-F4, hybrid function-F17 and composition function-F25) are plotted in Figs. 3 and 4 for the eight groups of compared algorithms. It can be seen that AEL + ABCs could obtain the smaller mean error metric values for F1, F4, F17 and F25 within 300,000 function evaluations, which suggests that the proposed AEL can help accelerate the convergence speed of ABCs.

In summary, the above experimental results show that AEL + ABCs have a significant advantage on the calculation accuracy and convergence speed for most of benchmark problems than those of ABCs, and the major contribution to performance improvement is from the proposed AEL.

4.3. Scalability to higher dimensionality

In order to study the scalability, AEL + ABCs are further tested on the 30 benchmark problems at 50*D* and 100*D*, respectively. From the statistical results of mean function error values summarized in Tables 5–8, it can be seen that AEL + ABCs have shown very good scalability to the problem dimension. That is, the performance of AEL + ABCs do not deteriorate seriously as the problem dimension increases. The results in row ‡/†/§ show that AEL + ABCs significantly outperform the corresponding ABCs on most of benchmark problems at 50*D* and 100*D*. Moreover, the statistical results of multiproblem Wilcoxon signed-rank test provided in Table 9 show that AEL + ABCs obtain higher R^+ values than R^- values in all cases. According to the Wilcoxon test $\alpha = 0.05$, there are significant differences in eight cases between AEL + ABCs and the corresponding ABCs. The results of Table 9 show that AEL + ABCs are also significantly better than the corresponding ABCs in the case of 50*D* and 100*D*. Therefore, we can conclude that AEL brings benefit to the eight ABC algorithms for solving the higher dimensionality benchmark problem at 50*D* and 100*D*.

It is worthy of note that these benchmark problems with different dimensions are employed just to study the scalability of AEL + ABCs, and the main aim of this work is not to develop AEL + ABCs to solve large scale global optimization problem. In order to improve the performance of AEL + ABCs on large scale global optimization problem, some advanced techniques, such as cooperative co-evolution [42], may be introduced into AEL + ABCs. This important work is out of the scope of this paper and needs to be studied independently in the future (Table 10).

4.4. Computation cost of AEL + ABCs

As discussed in Section 3, AEL + ABCs have additional computation burden compared with the corresponding ABCs. Here, the ratio of the mean runtime between AEL + ABCs and the corresponding ABCs on the 30 benchmark problems in CEC2014 at 30*D* is shown in Fig. 5. It is clear that the computation cost of AEL + ABCs are higher than those of the corresponding ABCs. Specifically, the average runtime of AEL + ABC, AEL + ABCM, ABC+ABCVSS, AEL + EABC, AEL + GABC, AEL + MABC, AEL + OPIABC and AEL + qABC on the 30 test problems is approximately 2.0231, 1.6124, 1.9657, 1.2958, 1.6232, 2.1968, 1.6493 and 1.4979 times longer than those of ABC, ABCM, ABCVSS, EABC, GABC, MABC, OPIABC and qABC, respectively. The results show that AEL + ABCs obtain the better performance at the expense of computation cost to some extent.

In order to further reduce the computation cost of AEL + ABCs, the adaptive encoding learning of each individual can be implemented approximately. For example, it can be implemented every several generations instead of every generation.

5. Discussion

As mentioned previously, the proposed AEL + ABCs include two different coordinate systems: natural coordinate system and eigen coordinate system. The adaptive selection mechanism is designed to control the computational resource assigned to the eigen coordinate system. In order to investigate the usage rate of eigen coordinate system, we plot the variations of adaptive selection probability of eigen coordinate system as the evolution progress in term of four different types of test functions (i.e., F1, F4, F17 and F25), which is shown in Fig. 6. These test problems involve unimodal function, simple multimodal function, hybrid function and composition function. The dimension was set to 30 for the four test functions and the maximum number of FES was set to 300,000.

From Fig. 6, we can observe that the usage rate of eigen coordinate system presents an increasing trend as the evolution progress. Thus it may be known that the eigen coordinate system plays the key role for ABCs in solving problems with variable linkages.

Table 10
The statistical results of performance comparisons of eight groups of algorithms for CEC2014 benchmarks at 30*D*, 50*D* and 100*D*.

Algorithms at 30*D*	R^+	R^-	*p*-value	0.05
AEL + ABC vs. ABC	310	125	4.38E−02	Yes
AEL + ABCM vs. ABCM	382	83	2.03E−03	Yes
AEL + ABCVSS vs. ABCVSS	342	93	6.87E−03	Yes
AEL + EABC vs. EABC	359.5	75.5	2.06E−03	Yes
AEL + GABC vs. GABC	336.5	98.5	9.77E−03	Yes
AEL + MABC vs. MABC	384	81	1.72E−03	Yes
AEL + OPIABC vs. OPIABC	389.5	75.5	1.16E−03	Yes
AEL + qABC vs. qABC	327	108	1.71E−02	Yes
Algorithms at 50*D*	R^+	R^-	*p*-value	0.05
AEL + ABC vs. ABC	377.5	87.5	2.77E−03	Yes
AEL + ABCM vs. ABCM	353.5	81.5	3.08E−03	Yes
AEL + ABCVSS vs. ABCVSS	379	86	2.50E−03	Yes
AEL + EABC vs. EABC	368	67	1.03E−03	Yes
AEL + GABC vs. GABC	396	69	6.97E−04	Yes
AEL + MABC vs. MABC	395.5	69.5	7.71E−04	Yes
AEL + OPIABC vs. OPIABC	351	114	1.41E−02	Yes
AEL + qABC vs. qABC	308.5	126.5	4.73E−02	Yes
Algorithms at 100*D*	R^+	R^-	*p*-value	0.05
AEL + ABC vs. ABC	373	62	7.43E−04	Yes
AEL + ABCM vs. ABCM	361.5	103.5	7.58E−03	Yes
AEL + ABCVSS vs. ABCVSS	358	77	2.24E−03	Yes
AEL + EABC vs. EABC	361.5	73.5	1.73E−03	Yes
AEL + GABC vs. GABC	375	90	3.27E−03	Yes
AEL + MABC vs. MABC	382.5	82.5	1.97E−03	Yes
AEL + OPIABC vs. OPIABC	351	114	1.41E−02	Yes
AEL + qABC vs. qABC	332	133	3.48E−02	Yes

6. Conclusions

In the family of swarm intelligence, ABCs have been proposed during the last few decades. Limited by the rotational variability of the search equation, the performance of most ABCs degrades in solving optimization problems with variable linkages. To overcome this limit, an adaptive encoding learning for ABCs, AEL + ABCs for short, is proposed in this paper. The AEL provides a simple yet efficient adaptive encoding learning method for the solution search equation, in which one is encoded in the eigen coordinate system and the other one is encoded in the natural coordinate system. In the process of evolution, the two coordinate systems are adjusted using an adaptive selection mechanism based on the success rate of each coordinate system in a learning period, which effectively balances the exploration and exploitation abilities of AEL + ABCs.

To evaluate the effectiveness of the proposed AEL on ABCs, eight groups of experiments, including ABC vs. AEL + ABC, ABCM vs. AEL + ABCM, ABCVSS vs. AEL + ABCVSS, EABC vs. AEL + EABC, GABC vs. AEL + GABC, MABC vs. AEL + MABC, OPIABC vs. AEL + OPIABC and qABC vs. AEL + qABC were conducted through 30 benchmark functions with different dimensions. The experimental results show that AEL + ABCs perform significantly better than their corresponding ABCs in the majority of the functions, and the major contribution to performance improvement is from the proposed AEL.

In the future, we will investigate the application of AEL + ABCs to solving other challenging research problems, such as large scale global optimization problems, multi-objective optimization problems, many-objective optimization problems, and complex real-world problems [43–45].

The source codes of AEL + ABC, AEL + ABCM, AEL + ABCVSS, AEL + EABC, AEL + GABC, AEL + MABC, AEL + OPIABC, and AEL + qABC are written in Matlab software and can be obtained from the first author upon request.

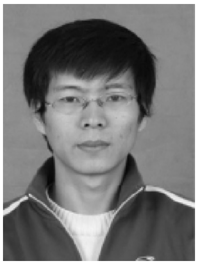
Acknowledgements

The authors would like to thank Dr. D. Karaboga, Dr. M.S. Kiran and Dr. S.Y. Yuen for providing the source codes of qABC, ABCVSS and OPIABC, respectively. This work is partly supported by the National Natural Science Foundation of China under Project Code (61803301, 61773314), and the Doctoral Foundation of Xi'an University of Technology (112-451116017).

References

[1] D. Karaboga, B. Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, J. Global Optim. 39 (November (3)) (2007) 459–471.
[2] D.C. Secui, A new modified artificial bee colony algorithm for the economic dispatch problem, Energy Convers. Manage. 89 (January) (2015) 43–62.
[3] D. Karaboga, E. Kaya, An adaptive and hybrid artificial bee colony algorithm (aABC) for ANFIS training, Appl. Soft Comput. 49 (December) (2016) 423–436.
[4] D. Karaboga, C. Ozturk, A novel clustering approach: Artificial Bee Colony (ABC) algorithm, Appl. Soft Comput. 11 (January (1)) (2011) 652–657.
[5] D. Karaboga, B. Akay, A survey: algorithms simulating bee swarm intelligence, Artif. Intell. Rev. 31 (June) (2009) 61–85.
[6] D. Karaboga, B. Gorkemli, C. Ozturk, et al., A comprehensive survey: artificial bee colony (ABC) algorithm and applications, Artif. Intell. Rev. 42 (June (1)) (2014) 284–294.
[7] A. Rajasekhar, N. Lynn, S. Das, et al., Computing with the collective intelligence of honey bees - a survey, Swarm Evol. Comput. 32 (February) (2017) 25–48.
[8] G.P. Zhu, S. Kwong, Gbest-guided artificial bee colony algorithm for numerical function optimization, Appl. Math. Comput. 217 (December (7)) (2010) 3166–3173.
[9] R. Storn, K. Price, Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces, J. Global Optim. 11 (December (4)) (1997) 341–359.
[10] W.F. Gao, S.Y. Liu, L.L. Huang, Enhancing artificial bee colony algorithm using more information-based search equations, Inf. Sci. 270 (June (1)) (2014) 112–133.
[11] W.F. Gao, S.Y. Liu, A modified artificial bee colony algorithm, Comput. Oper. Res. 39 (March (3)) (2012) 687–697.
[12] D. Karaboga, B. Gorkemli, A quick artificial bee colony (qABC) algorithm and its performance on optimization problems, Appl. Soft Comput. 23 (October) (2014) 227–238.
[13] L.Z. Cui, G.H. Li, X.Z. Wang, et al., A ranking-based adaptive artificial bee colony algorithm for global numerical optimization, Inf. Sci. 417 (November) (2017) 169–185.
[14] X. Zhang, S.Y. Yuen, Improving artificial bee colony with one-position inheritance mechanism, Memet. Comput. 5 (September (3)) (2013) 187–211.
[15] X.N. Li, G.F. Yang, Artificial bee colony algorithm with memory, Appl. Soft Comput. 41 (April) (2016) 362–372.
[16] X.Q. Pan, Y. Lu, N. Sun, et al., A hybrid artificial bee colony algorithm with modified search model for numerical optimization, Cluster Comput. (2017), http://dx.doi.org/10.1007/s10586-017-1343-0.
[17] D. Kumar, K.K. Mishra, Co-variance guided artificial bee colony, Appl. Soft Comput. 70 (September) (2018) 86–107.
[18] M.S. Kiran, H. Hakli, M. Gunduz, et al., Artificial bee colony algorithm with variable search strategy for continuous optimization, Inf. Sci. 300 (April) (2015) 140–157.
[19] W.F. Gao, L.L. Huang, S.Y. Liu, et al., Artificial bee colony algorithm with multiple search strategies, Appl. Math. Comput. 271 (November) (2015) 269–287.
[20] W.L. Xiang, X.L. Meng, Y.Z. Li, et al., An improved artificial bee colony algorithm based on the gravity model, Inf. Sci. 429 (March) (2018) 49–71.
[21] B. Alatas, Chaotic bee colony algorithms for global numerical optimization, Expert Syst. Appl. 37 (August (8)) (2010) 5682–5687.
[22] F. Kang, J.J. Li, Z.Y. Ma, Rosenbrock artificial bee colony algorithm for accurate global optimization of numerical functions, Inf. Sci. 181 (August (16)) (2011) 3508–3531.
[23] W.F. Gao, S.Y. Liu, L.L. Huang, A novel artificial bee colony algorithm based on modified search equation and orthogonal learning, IEEE Trans. Cybern. 43 (June (3)) (2013) 1011–1024.
[24] W.F. Gao, L.L. Huang, S.Y. Liu, et al., Artificial bee colony algorithm based on information learning, IEEE Trans. Cybern. 45 (December (12)) (2015) 2827–2839.
[25] L.Z. Cui, G.H. Li, Z.X. Zhu, et al., A novel artificial bee colony algorithm with an adaptive population size for numerical function optimization, Inf. Sci. 414 (November) (2017) 53–67.
[26] L.Z. Cui, G.H. Li, X.Z. Wang, et al., A ranking-based adaptive artificial bee colony algorithm for global numerical optimization, Inf. Sci. 417 (November) (2017) 169–185.
[27] W.L. Xiang, S.F. Ma, M.Q. An, hABCDE: a hybrid evolutionary algorithm based on artificial bee colony algorithm and differential evolution, Appl. Math. Comput. 238 (July) (2014) 370–386.
[28] R.V. Rao, V.J. Savsani, D.P. Vakharia, Teaching-learning-based optimization: a novel method for constrained mechanical design optimization problems, Comput.-Aid. Des. 43 (March (3)) (2011) 303–315.
[29] X. Chen, B. Xu, C.L. Mei, et al., Teaching-learning-based artificial bee colony for solar photovoltaic parameter estimation, Appl. Energy 212 (February) (2018) 1578–1588.
[30] X.H. Yan, Y.L. Zhu, W.P. Zou, et al., A new approach for data clustering using hybrid artificial bee colony algorithm, Neurocomputing 97 (November) (2012) 241–250.
[31] M.S. Klran, M. Gündüz, A recombination-based hybridization of particle swarm optimization and artificial bee colony algorithm for continuous optimization problems, Appl. Soft Comput. 13 (April (4)) (2013) 2188–2203.
[32] N. Hansen, A. Ostermeier, Completely derandomized self-adaptation in evolution strategies, Evol. Comput. 9 (June (2)) (2001) 159–195.
[33] X. Chen, H. Tianfield, W.L. Du, et al., Biogeography-based optimization with covariance matrix based migration, Appl. Soft Comput. 45 (August) (2016) 71–85.
[34] Y. Wang, H.X. Li, T.W. Huang, L. Li, Differential evolution based on covariance matrix learning and bimodal distribution parameter setting, Appl. Soft Comput. 18 (May) (2014) 232–247.
[35] S.M. Guo, C.C. Yang, Enhancing differential evolution utilizing eigenvector-based crossover operator, IEEE Trans. Evol. Comput. 19 (February (1)) (2015) 31–49.
[36] Q.Y. Jiang, L. Wang, J.T. Cheng, et al., Multi-objective differential evolution with dynamic covariance matrix learning for multi-objective optimization problems with variable linkages, Knowl.-Based Syst. 121 (April) (2017) 111–128.
[37] M.N. Ras, D.N. Wilke, A.A. Groenwold, S. Kok, On rotationally invariant continuous-parameter genetic algorithms, Adv. Eng. Softw. 78 (December) (2014) 52–59.
[38] Q.Y. Jiang, L. Wang, X.H. Hei, et al., ARAE-SOM+BCO: An enhanced artificial raindrop algorithm using self-organizing map and binomial crossover operator, Neurocomputing 275 (January) (2018) 2716–2739.
[39] A.K. Qin, V.L. Huang, P.N. Suganthan, Differential evolution algorithm with strategy adaptation for global numerical optimization, IEEE Trans. Evol. Comput. 13 (September (2)) (2009) 398–417.
[40] J.J. Liang, B.Y. Qu, P.N. Suganthan, Problem definitions and evaluation criteria for the CEC2014 special session and competition on single objective real-parameter numerical optimization, Technical Report 201311, Computational Intelligence Laboratory, Zhenzhou University, Zhenzhou, China and Nanyang Technological University, Singapore, 2013, December.
[41] J. Alcalá-Fdez, L. Sánchez, S. García, et al., KEEL: a software tool to assess evolutionary algorithms for data mining problems, Soft Comput. 13 (February (3)) (2009) 307–318.
[42] M.N. Omidvar, M. Yang, Y. Mei, et al., DG2: A faster and more accurate differential grouping for large-scale black-box optimization, IEEE Trans. Evol. Comput. 21 (December (6)) (2017) 929–942.
[43] S.H. Wang, P. Phillips, Y.X. Sui, et al., Classification of Alzheimer's disease based on eight-layer convolutional neural network with leaky rectified linear unit and max pooling, J. Med. Syst. 42 (May (85)) (2018) 1–11.
[44] S.H. Wang, Y.Y. Jiang, X.X. Hou, et al., Cerebral micro-bleed detection based on the convolution neural network with rank based average pooling, IEEE Access 5 (August) (2017) 16576–16583.
[45] S.H. Wang, T.M. Zhan, Y. Chen, et al., Multiple sclerosis detection based on biorthogonal wavelet transform, RBF kernel principal component analysis, and logistic regression, IEEE Access 4 (2016) 7567–7576.

Jingyuan Yang received her B.S. degree in computer science and technology from Xi'an University of Technology, Xi'an, China, in 2016. She is currently pursuing her M.S. degree in computer architecture from Xi'an University of Technology, Xi'an, China. Her current research interests include evolutionary algorithms, deep learning, and neural networks.

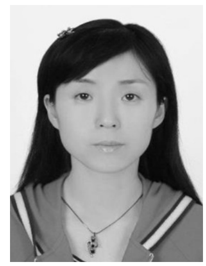


Qiaoyong Jiang received the B.S. degree in applied mathematics from Wenzhou University, Wenzhou, China, in 2008, the M.S. degree in computational mathematics from Beifang University of Nationalities, Yinchuan, China, in 2011, his Ph.D. degree in pattern recognition and intelligent systems from Xi'an University of Technology, Xi'an, China. He is currently a lecturer with the Faculty of Computer Science and Engineering, Xi'an University of Technology, Xi'an, China. His current research interests include meta-heuristics, evolutionary computation and multi-objective optimization.

Lei Wang received his B.S. degree and M.S. degree in computer science and technology from Xi'an University of Technology, Xi'an, China, in 1994 and 1997, respectively, and his Ph.D. degree from in electronic science and technology from Xidian University, Xi'an, China, in 2001. He is currently a professor with the Faculty of Computer Science and Engineering, Xi'an University of Technology, Xi'an, China. His current research interests include evolutionary algorithms, neural networks and data mining.

Shuai Liu received his bachelors (2002), masters (2004) and doctoral (2011) degrees from the College of Computer Science and Technology, Jilin University. He now works as a Professor in the College of Computer Science, Inner Mongolia University. He has published more than 20 papers in Elsevier, Springer, Oxford and other publishers. He is now interested in image processing, computer vision and fractals. He now acts as an (associate/guest) editor in IEEE Access, Mobile Networks and Application and so on. He also acts as are viewer of IEEE Transactions on Fuzzy Systems, IEEE Transactions on Image Processing, Information Sciences, Applied Mathematics and Computation, IEEE Access, and so on.

Yu-Dong Zhang received her B.S. and M.S. degree in degree in information engineering from Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2004 and 2007, respectively, and his Ph.D. degree from in signal and information processing from Southeast University, Nanjing, China, in 2010. He is currently a professor with the Department of Informatics, University of Leicester, United Kingdom. His current research interests include evolutionary algorithms, deep learning and data mining.



Wei Li received her B.S. degree in computer education from Shaanxi Normal University, Xi'an, China, in 1995, and her M.S. degree and Ph.D. degree in computer science and technology from Xi'an University of Technology, Xi'an, China, in 2001 and 2016, respectively. She is currently an associate professor with the Faculty of Computer Science and Engineering, Xi'an University of Technology, Xian, China. Her current research interests include evolutionary algorithms, neural networks, and data mining.

Bin Wang received his B.S. degree in measurement control technology and instrument from Xi'an University of Technology, Xi'an, China, in 1994, respectively, and his M.S. degree from in computer science and technology from Xidian University, Xi'an, China, in 2004. He is currently an associate professor with the Faculty of Computer Science and Engineering, Xi'an University of Technology, Xi'an, China. His current research interests include evolutionary algorithms, neural networks, and data mining.