

Modified multiple search cooperative foraging strategy for improved artificial bee colony optimization with robustness analysis

F. Harfouchi¹ · H. Habbi¹ · C. Ozturk² · D. Karaboga²

Published online: 28 June 2017
© Springer-Verlag GmbH Germany 2017

Abstract Considering that extending the concept of bees partitioning into subgroups of foragers to the onlooker phase of the cooperative learning artificial bee colony (CLABC) strategy is not only feasible from algorithmic viewpoint but might reflect real behavioral foraging characteristics of bee swarms, this paper studies whether the performance of CLABC can be enhanced by developing a new model for the proposed cooperative foraging scheme. Relying on this idea, we design a modified cooperative learning artificial bee colony algorithm, referred to as mCLABC. The design procedure is built upon the definition of a partitioning scheme of onlookers allowing the generation of subgroups of foragers that might evolve differently by using specific solution search rules. In order to improve the involving of local and global search at both employed and onlooker levels, the multiple search mechanism is further tuned and scheduled according to a random selection strategy defined on the evolving parameters. Moreover, a detailed performance and robustness study of the proposed algorithm dealing with the analysis of the impact of different structural and parametric settings is conducted on benchmark optimization problems. Superior convergence performance, better solution quality, and strong robustness are the main features of the proposed strategy in comparison with recent ABC variants and advanced methods.

Keywords Swarm intelligence · Artificial bee colony optimization · Search strategy · Cooperative foraging learning

1 Introduction

During past decades, population-based search algorithms have shown great success in solving different kinds of optimization problems. Evolutionary and swarm intelligence methods are common population-based algorithms which have received particular interest in recent years. Extensive efforts are still being made on basic population-based algorithms such as differential evolution (DE) (Storn and Price 1997), evolutionary programming (EP) (Fogel 1995), particle swarm optimization (PSO) (Kennedy and Eberhart 1995), ant colony optimization (ACO) (Dorigo and Stutzle 2004), artificial bee colony optimization (ABC) (Karaboga 2005), in an attempt to improve their essential computational features such as convergence performance and robustness with regard to exploration and exploitation abilities.

The ABC optimization concept relies on a stochastic search mechanism that mimics the natural behavioral foraging process of honeybees as being initially introduced by Karaboga (2005). The ABC model has shown considerable improvements over existing classical and advanced optimization strategies (Karaboga and Akay 2009). It requires only three control parameters to be tuned, i.e., the population size (number of food sources), the exploration parameter called “*limit*,” and the maximum number of generations. In general, it has been proven that ABC achieves efficient global search and, hence, can be seen as a global optimizer. So far, ABC has gained substantial interest with an increasing number of applications to various scientific and engineering problems, such as clustering problems (Karaboga and Ozturk 2011),

Communicated by V. Loia.

✉ H. Habbi
habbi_hacene@hotmail.com

¹ Applied Automation Laboratory, University of Boumerdès, 35000 Boumerdès, Algeria

² Computer Engineering Department, Engineering Faculty, Erciyes University, 38039 Kayseri, Turkey

economic dispatch problem (Secui 2015), training neural networks (Karaboga and Ozturk 2009), vehicle routing problem (Szeto et al. 2015), image processing (Banharnsakun et al. 2011), scheduling problems (Taheri et al. 2013), wireless sensor network (Okdem et al. 2011), optimal filter design (Bose et al. 2014), optimal power flow problem (Yuan et al. 2015; Jadhav and Bamane 2016), fuzzy systems design (Habbi et al. 2015), truss structure design (Sonmez 2011), stock price prediction (Hsieh et al. 2011) and other problems (Saffari et al. 2016; Habbi et al. 2015; Habbi and Boudouaoui 2014; Aydođdu et al. 2016; Habbi 2012; Liang and Lee 2015; Sun et al. 2013). However, like any other population-based techniques, ABC has its own shortcomings: the ABC model has good exploration ability, but its exploitation level needs further improvement, which represents a challenging issue for ABC convergence and performance.

The basic ABC model was subjected to many enhancements aiming to increase its computational performance and robustness. To this end, different modified and hybridized ABC-based algorithms have been developed in order to achieve good balance between exploration and exploitation. Let us mention for instance the work by Zhu and Kwong (2010) where a global-best-guided ABC variant is proposed. Akay and Karaboga (2012) designed a modified ABC algorithm by controlling the frequency of perturbation and introducing the ratio of the variance operator. Gao et al. (2014) developed new search equations to adjust exploration and exploitation capability of the ABC algorithm. Xiang et al. (2014) hybridized DE and ABC with a new search strategy inspired from gbest-guided ABC to enhance the convergence rate of traditional ABC. Kiran and Findik (2014) developed a simple version of the basic ABC algorithm by using direction information regarding the solutions to improve the convergence characteristics of the basic algorithm. Li and Yang (2016), inspired by the biological study of natural honeybees, presented a new ABC variant which imitates a memory mechanism to the artificial bees to memorize their previous successful experiences of foraging behavior. Babaoglu (2015) proposed a distribution-based solution update rule for the basic ABC algorithm, which uses the mean and standard deviation of selected two food sources to obtain a new candidate solution. Alatas (2010) proposed a chaotic ABC algorithm, in which many chaotic maps for parameters adapted from the standard ABC algorithm were introduced to improve its convergence performance. Biswas et al. (2014) presented migratory multi-swarm ABC where subpopulations of swarms employ different search strategies. Harfouchi and Habbi (2016) developed a novel multiple search ABC variant with cooperative learning paradigm. Wang et al. (2014) proposed an integration of update rules for ABC algorithm and analyzed the performance of the proposed method on solving numerical functions. Li et al. (2012) proposed an improved ABC algorithm with the abilities of prediction and

selection by introducing an inertia weight and two acceleration coefficients.

Following up ABC improvement efforts, we presented in Harfouchi and Habbi (2016) a novel variant of ABC model relying on the concept of cooperative foraging learning among bee swarms as highlighted above. In the proposed ABC variant, referred to as CLABC (Cooperative learning artificial bee colony) algorithm, a structural modification of the employed phase is introduced. Employed bees are partitioned into subgroups of foragers which evolve differently with multiple search mechanism. The onlookers are not subjected to this modified behavioral structure and thus are supposed to operate into a unique group as defined in the basic ABC model. From behavioral analysis viewpoint, extending the general concept of foragers partitioning to cover onlookers is not only feasible but might reflect natural features of bee swarms. Relying on this idea, the present paper addresses the problem of designing a modified framework of the CLABC algorithm by introducing structural modifications to better characterize the involving of subgroups of employed and onlooker bees in the foraging process. The proposed optimization strategy also performs with multiple search mechanism that allows mutual and simultaneous exchange of information among the foraging subgroups. Moreover, a detailed performance and robustness study of the proposed ABC variant dealing with the analysis of the impact of different parameter and structural settings is conducted on benchmark and complex optimization problems.

The remaining of this paper is organized as follows. Section 2 introduces the basics of ABC optimization method. The modified cooperative learning artificial bee colony (mCLABC) strategy is described in Sect. 3. To assess its performance, experiments on benchmark numerical and composition test functions are presented in Sect. 4. In Sect. 5, a detailed robustness study is conducted with respect to a number of structural and parameter settings of the proposed method. Finally, conclusions and remarks are given in Sect. 6.

2 Basics of artificial bee colony optimization

As to simulate the foraging behavior of honeybees, the ABC model employs three types of foragers that are categorized into employed bees, onlooker bees and scouts (Karaboga 2005). With regard to natural foraging process of bees, three operational phases are distinguished through which multi-tasks are rigorously performed for profitable search. From behavioral viewpoint, an employed bee searches around the current food source to find a new source position with better nectar amount. If the nectar amount of the discovered position is higher than that of the previous one, the bee saves the new position in her memory and forgets the old one. Each employed bee is associated with a food source

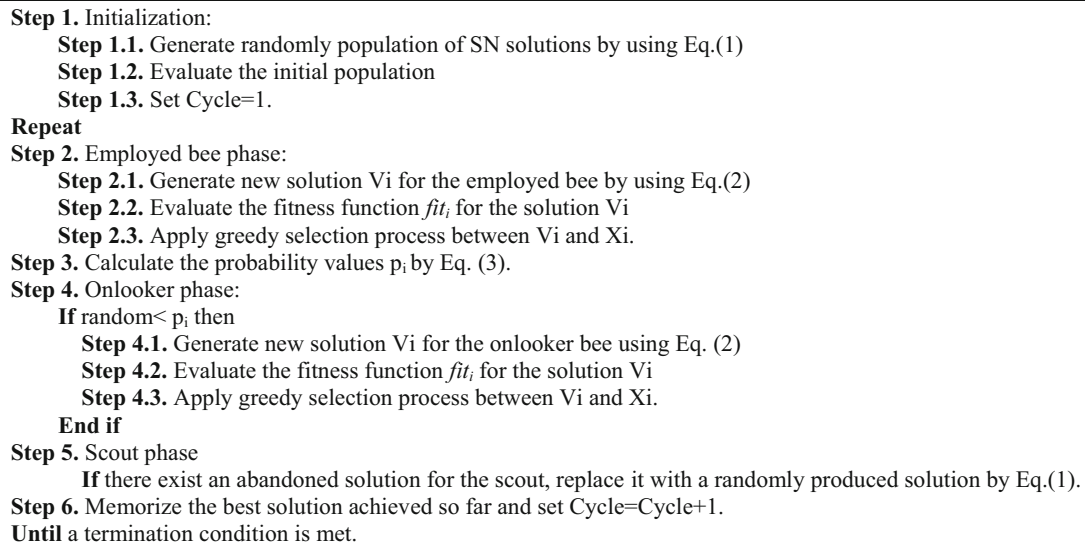


Fig. 1 Pseudo-code of the basic ABC algorithm

which means the number of employed bees is same as the number of food sources. After all employed bees complete their search, they share the gathered information about foraging with onlooker bees which are waiting in the hive. An onlooker bee chooses a food source according to a probabilistic greedy selection mechanism. Therefore, food sources with better profitability will get higher probability to be selected by the onlookers. Similarly, each onlooker bee produces a modification on the position in her memory and checks the nectar amount of the generated candidate source. If a position cannot be improved further through a predetermined limit tolerance called “*limit*”, then that food source is assumed to be abandoned. The corresponding employed bee becomes then a scout. The abandoned position will be replaced with a new food source found by the scout.

From algorithmic viewpoint, the first step of the ABC optimization process consists in generating randomly a population of SN solutions (food source positions) in the admissible search domain by using the following equation:

$$x_i^j = x_{\min}^j + rand(0, 1)(x_{\max}^j - x_{\min}^j) \tag{1}$$

where $i = 1, 2, \dots, SN$, $j = 1, 2, \dots, D$ and D is the number of optimization parameters; x_{\min}^j and x_{\max}^j are the lower and upper bounds of the dimension j , respectively.

After the initialization step, the population of food sources is subjected to repeated cycles of three-phase search process of employed bees, onlooker bees, and scout bees. Employed bees use the following search equation to generate new candidate solutions v_i from the old ones x_i (Karaboga and Basturk 2008):

$$v_i^j = x_i^j + \phi_i^j(x_i^j - x_k^j) \tag{2}$$

where $i = 1, 2, \dots, SN$, $j \in \{1, 2, \dots, D\}$ and $k \in \{1, 2, \dots, SN\}$ are indexes selected randomly, and k different from i . ϕ_i^j is a uniform random number in $[-1, 1]$. Greedy selection between the old and the updated food source position is performed by the employed bee based on fitness value evaluation. This valuable information about the position and the quality of the food sources is shared with the onlooker bees.

Unlike the employed bee, an onlooker bee chooses a food source with a probability that depends on its nectar amount and is calculated as follows:

$$p_i = \frac{fit_i}{\sum_{n=1}^{SN} fit_n} \tag{3}$$

where fit_i indicates the fitness value of the solution i . After having chosen a food source, an onlooker bee determines a new candidate solution by using Eq. (2), and then greedy selection is applied between the new solution and the old one. If a solution cannot be further improved by means of a predefined limit value, it is considered as exhausted and is abandoned. In this case, a new position is randomly determined by a scout bee by using Eq. (1). The three-phase procedure of the basic ABC optimization model is summarized in Fig. 1.

3 The modified cooperative learning strategy for improved ABC optimization

Search equations in population-based algorithms are key operators based on which solution updating is performed. Obviously, the search mechanism guides the optimization

process to optimal or near-optimal solutions in a predefined or randomly chosen direction by updating rules. Evaluating the computational performance of a population-based algorithm might be done in accordance with two important features which are the exploration and the exploitation abilities. The exploration process is related to the ability of carrying out independent search for optimal solution finding, whereas the exploitation process is related to the ability of integrating existing knowledge to improve solutions (Akay and Karaboga 2012). For the sake of efficiency, search equations have to be designed so that exploration and exploitation are adequately balanced. As mentioned above, the basic ABC model uses the search rule defined by Eq. (2) which is suitable for global search. However, ABC might be slow to converge and easily get trapped in local solutions, which is a challenging issue for many population-based search algorithms. To deal with this common drawback, one actual and active research trend in ABC optimization consists in the design of well-established search strategies. The literature reports a number of tentative works on this particular subject (Gao et al. 2012, 2013a; Zhu and Kwong 2010; Gao and Liu 2011; Gao et al. 2014).

With the aim to further improve the performance of ABC optimization, a modified cooperative learning strategy is investigated in the present study. To implement the proposed cooperative concept, structural modifications are introduced into the original CLABC described in Harfouchi and Habbi (2016). The design procedure is built upon the definition of a partitioning scheme of onlookers allowing the generation of subgroups of foragers that might evolve differently by using specific search equations. The updating rules describing the multi-search mechanism have also been adjusted together with some parameter settings. Unlike the original CLABC, the transformed cooperative learning framework which is referred to as mCLABC (modified cooperative learning ABC) involves local and global search rules at both employed and onlooker levels. This might of course impact positively the exploration and the exploitation capabilities of the designed model. Based on this scheme, a random cooperative foraging learning strategy is built upon the combination of a set of local and global search equations through which separate subgroups of foragers evolve simultaneously at both employed or onlooker stages. More precisely, the idea relies on modeling the foraging behavior of swarm honeybees according to the following behavioral concept: "In a bee swarm, employed and onlooker bees could be divided into subgroups of foragers. The subgroups of employed foragers or onlooker might perform food search independently, engage at each level specific search tools differently and also might exchange information about the foraging process mutually".

Modeling such a behavioral concept needs first a search mechanism to be defined for each foraging subgroup. A mul-

tipale search mechanism is then adopted which consists in a set of global and local search equations that allow mutual and simultaneous exchange of foraging information between the subgroups of a given bees category. The solution updating equations are defined as follows:

$$v_i^j = x_i^j + \phi_i^j (x_i^j - x_k^j) \quad (4)$$

$$i = 1, 2, \dots, SN, i \neq k, j \in \{1, 2, \dots, D\}$$

$$v_i^j = x_k^j + \phi_i^j (x_k^j - x_{k_1}^j) \quad (5)$$

$$i = 1, 2, \dots, SN, i \neq k \neq k_1, j \in \{1, 2, \dots, D\}$$

$$v_i^j = x_k^j + \phi_i^j (x_{k_1}^j - x_{k_2}^j) \quad (6)$$

$$i = 1, 2, \dots, SN, i \neq k \neq k_1 \neq k_2, \\ j \in \{1, 2, \dots, D\}$$

$$v_i^j = x_{\text{best}}^j + \phi_i^j (x_i^j - x_{\text{mean}}^j) + \phi_i^j (x_k^j - x_{k_1}^j) \quad (7)$$

$$i = 1, 2, \dots, SN, i \neq k \neq k_1, j \in \{1, 2, \dots, D\}$$

$$v_i^j = x_{\text{best}}^j + \phi_i^j (x_k^j - x_{k_1}^j) \quad (8)$$

$$i = 1, 2, \dots, SN, i \neq k \neq k_1, j \in \{1, 2, \dots, D\}$$

$$v_i^j = x_i^j + \phi_i^j (x_i^j - x_{\text{mean}}^j) + \phi_i^j (x_k^j - x_{k_1}^j) \quad (9)$$

$$i = 1, 2, \dots, SN, i \neq k \neq k_1, j \in \{1, 2, \dots, D\}$$

where $\phi_i^j \in [-1, 1]$, $\phi_i^j \in [0, 1]$ are random numbers generated for each dimension at each processing time. x_k^j , $x_{k_1}^j$ and $x_{k_2}^j$ are the j th dimensions of solutions selected randomly from the populations of bee subgroups (subpopulations), k, k_1 and k_2 are not equal to each other and to i , x_{best}^j denotes the j th dimension of the best solution obtained by the subpopulations so far, and x_{mean}^j is the average solution corresponding to the j th dimension for all solutions in the subpopulations.

Here, Eqs. (4)–(6) are suitable for global search and Eqs. (7)–(9) for local search. To incorporate the multiple search mechanism described by Eqs. (4)–(9) into the framework of ABC model, the employed and onlooker foragers are first categorized into a predefined number of subgroups. Subpopulations of potential solutions are then generated iteratively at both employed and onlooker phases. The formed subgroups of employed and onlooker bees will have to evolve differently during the foraging process by engaging specific search strategies which are not similar to those defined for the other subgroups. According to design, the search Eqs. (4) and (5) will be called by the first group of employed and onlooker bees to generate new candidate solutions. Only one of those two search equations is used to update the solution with regard to a specified probability which depends on the current iteration and the maximum number of cycles as defined in Mohamed (2015). The nonlinear decreasing probability rule introduced in Mohamed (2015) has been determined empirically to enhance the global exploration capability of the evolutionary strategy. Following this reasoning and based on several experiments, we adopted a different

Step 1. Initialization:

Step 1.1. Generate randomly population of SN solutions by using Eq.(1)

Step 1.2. Evaluate the initial population

Step 1.3. Set Cycle=1.

Repeat

Step 2. Employed bee phase:

Step 2.1. Send subgroup 1 of employed foragers and perform the process described in Fig. 3 by using Eq. (4) and Eq. (5)

Step 2.2. Send subgroup 2 of employed foragers and perform the process described in Fig. 3 by using Eq. (6) and Eq. (7)

Step 2.3. Send subgroup 3 of employed foragers and perform the process described in Fig. 3 by using Eq. (8) and Eq. (9)

Step 2.4. Apply greedy selection process between V_i and X_i .

Step 3. Calculate the probability values p_i and select the solutions which will be updated by the onlookers.

Step 4. Onlooker bee phase:

Step 4.1. Send subgroup 1 of onlooker foragers and perform the process described in Fig. 4 by using Eq. (4) and Eq. (5)

Step 4.2. Send subgroup 2 of onlooker foragers and perform the process described in Fig. 4 by using Eq. (6) and Eq. (7)

Step 4.3. Send subgroup 3 of onlooker foragers and perform the process described in Fig. 4 by using Eq. (8) and Eq. (9)

Step 5. Scout phase

If there exist an abandoned solution for the scout, replace it with a randomly produced solution by Eq. (1).

Step 6. Memorize the best solution achieved so far and set Cycle=Cycle+1.

Until a termination condition is met.

Fig. 2 The algorithmic framework of the proposed mCLABC algorithm

form of probability rule for our design which is given by the increasing nonlinear form $\gamma = (iter/MCN)^{1/2}$. Based on this definition, we can for instance define the following selective updating rule for a given subgroup of foragers:

$$\begin{aligned}
 & \text{if}(\text{random} \leq \gamma) \text{ then} \\
 & v_i^j = x_i^j + \phi_i^j(x_i^j - x_k^j) \\
 & \text{else} \\
 & v_i^j = x_k^j + \phi_i^j(x_k^j - x_{k1}^j) \\
 & \text{end}
 \end{aligned} \tag{10}$$

where $iter$ is the current generation number and MCN is the maximum number of cycles. Obviously, it is clear that, from Eq. (10), the probability of using one of the two search rules is a function of the iteration number. The nonlinear form of the probability is determined empirically with regard to search process convergence rate which is guided by the above probabilistic strategy. Based on the same definition of probability given above, the random search Eq. (6) and the gbest-guided Eq. (7) are used for the second subgroup of employed and onlooker bees. Similarly, the third subgroup employs Eqs. (8) and (9) to update candidate solutions. Here, search Eq. (8) performs solution updating in a random direction guided by the gbest solution found so far, while search Eq. (9) is mean-oriented. On the other hand, a strategy to select the number of evolving optimization parameters is incorporated in the modified algorithmic framework of mCLABC at both employed and onlooker levels. For this purpose, employed bees are managed to start by changing one random dimension. Then, when the number of iterations reaches $iter = \frac{1}{5}MCN$, the algorithm will have to optimize randomly one or all parameters simultaneously. While reaching the end of the optimization process, i.e., between

$iter = \frac{4}{5}MCN$ and $iter = MCN$, the employed bees will update all parameters of the solution. Subsequently, after completion of the employed bee phase and based on greedy selection, the subgroups of onlookers start to update one random parameter of the solution. Then, when $iter = \frac{1}{5}MCN$, the onlooker bees will have to change randomly one or all parameters until $iter = MCN$. In this way, we attempt to mimic the natural behavior of honeybees which might look highly selective when concerned with the alteration of the features of a given food source. In the absence of tight characterization of such a behavioral foraging concept, we adopted a random selection strategy. Figure 2 describes the general framework of the modified CLABC algorithm. The pseudo-codes describing the foraging process of the associated employed and onlooker subgroups are given in Figs. 3 and 4, respectively.

The proposed multi-search strategy is designed with the possibility to apply all updating rules at any level of the search process. The mechanism is set to ensure more global search in the beginning of the process to increase exploration, and more local search through generations to increase exploitation. Regarding probability definition, search Eqs. (5), (6), and (9) will have higher possibility to be called in the beginning of the optimization process which might ensure diversity. Then, all global Eqs. (4), (5), and (6) and local Eqs. (7), (8), and (9) will be used randomly to intensify the search. Accordingly, both global exploration and local exploitation are balanced at this level. Afterward, and depending on γ values, all search strategies might be employed, but again Eqs. (4), (7) and (8) will have higher probability to be involved which might increase exploitation ability and convergence performance and preserve exploration capability. This strategy of being involved of solutions updating rules in mCLABC algorithm is described in Fig. 5.

```

Employed Subgroup
Step 1. Set  $\gamma = (\text{Cycle}/\text{MaxCycle})^{(1/2)}$ .
Step 2.
For all  $\text{SN}_{\text{employed}}$  food sources visited by the employed bees
Step 2.1. Apply the evolving parameters selecting strategy
If (All parameters are to be altered)
If (random  $\leq \gamma$ )
Step 2.2. Produce a new food source  $V_i$  for the employed bee by using the specified search rule (SRule1)
Else
Step 2.3. Produce a new food source  $V_i$  for the employed bee by using the specified search rule (SRule2)
End if
Else
If (random  $\leq \gamma$ )
Step 2.4. Produce a new food source  $V_i$  for the employed bee by changing one random parameter
using the specified search rule (SRule1)
Else
Step 2.5. Produce a new food source  $V_i$  for the employed bee by changing one random parameter
using the specified search rule (SRule2)
End if
Endif
End for

```

Fig. 3 Pseudo-code of the employed subgroup foraging process

```

Onlooker Subgroup:
Step 1. set  $t=0, i=1$ .
While ( $t \leq \text{SN}_{\text{onlooker}}$ ) do
Step 2.
If (random  $< p_i$ ) then
Step 2.1. Apply the evolving parameters selecting strategy
If (All parameters are to be altered)
If (random  $\leq \gamma$ )
Step 2.1.1. Produce a new food source  $V_i$  for the onlooker bee by using the specified search rule (SRule1)
Else
Step 2.1.2. Produce a new food source  $V_i$  for the onlooker bee by using the specified search rule (SRule2)
End if
Else
If (random  $\leq \gamma$ )
Step 2.1.3. Produce a new food source  $V_i$  for the onlooker bee by changing one random parameter
using the specified search rule (SRule1)
Else
Step 2.1.4. Produce a new food source  $V_i$  for the onlooker bee by changing one random parameter
using the specified search rule (SRule2)
End if
End if
Step 2.1.5. Apply greedy selection process between  $V_i$  and  $X_i$ 
Step 2.1.6.  $t = t+1$ 
End if
Step 2.2. Set  $i=i+1$ , if  $i=(\text{SN}_{\text{onlooker}})+1$ , set  $i=1$ .
End While

```

Fig. 4 Pseudo-code of the onlooker subgroup foraging process

4 Numerical experiments and results

4.1 Experiments on benchmark numerical functions

To demonstrate the computational performance of the proposed mCLABC algorithm, a set of 22 benchmark functions with dimensions $D = 15, 30, 50, 60, 100$ and 200 are used. The test functions are listed in Table 1. They are categorized into the low-, middle- and high-dimensional functions. For instance, the high-dimensional functions are the functions

F1–F20 with $D = 60$ and the functions F21 and F22 with $D = 200$. For all experiments, the maximum number of function evaluations (FEs) is used as termination condition. FEs is taken as 50,000, 100,000, and 200,000 for the low-, middle-, and high-dimensional functions, respectively. Colony size is chosen as $\text{CS} = 40$. The limit value is set to $\text{limit} = D * \text{SN}$, where D is the dimensionality of the problem and SN is the number of food sources.

The compared results are taken directly from their corresponding literature, and thus setting parameters of mCLABC

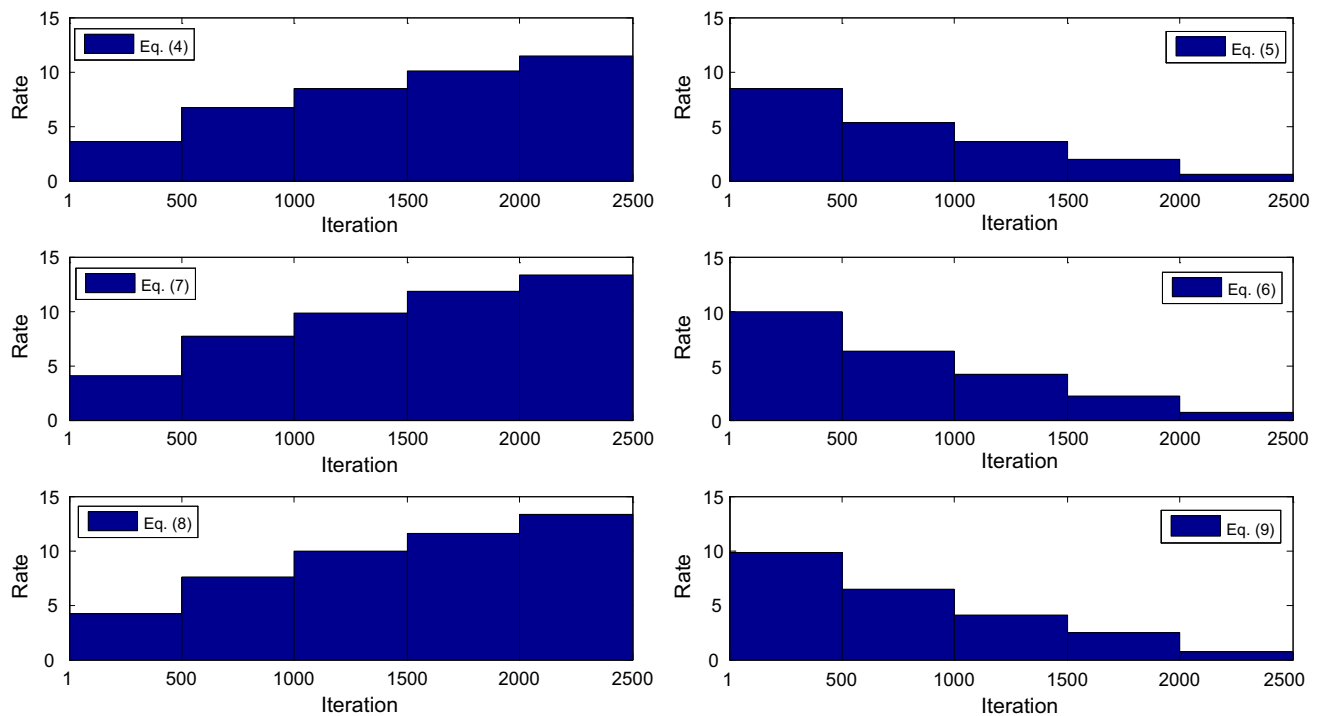


Fig. 5 Rate of being selected of the search equations in mCLABC algorithm

are set same as the compared methods. The best results obtained over running times are marked in bold.

4.1.1 Comparison of mCLABC with the basic ABC and CLABC

The proposed mCLABC model is first compared with the basic ABC algorithm and the original CLABC algorithm. The results are shown in Tables 2, 3 and 4 in terms of means and standard deviations of the solutions obtained over 30 independent runs. Figure 6 depicts the convergence curves of ABC, CLABC, and mCLABC on some representative test functions. The resulting plots show that mCLABC performs differently and better in the most of cases and achieves faster convergence than ABC and CLABC especially on middle- and high-dimensional test functions. Significant conclusions on the performance of different algorithms can be drawn from the results summarized in Tables 2, 3 and 4.

Indeed, as given in Table 2, the mCLABC and CLABC operate clearly better than ABC. However, CLABC outperforms mCLABC on the low-dimensional unimodal functions F1, F2, F3, F5, F9, and F10, while the mCLABC algorithms shows superiority over CLABC on most of low-dimensional multimodal functions F14, F15, F18, F19, F20, and F22 which are rather complex numerical functions.

From Table 3, it can be seen that mCLABC has significantly improve the optimization results in the majority of middle-dimensional unimodal and multimodal test functions.

The CLABC results outperform the mCLABC on only three test functions F4, F5, and F14 with a slight improved difference for F15.

Moreover, Table 4 shows that the proposed mCLABC exhibits a remarkable high performance in solving different kinds of high-dimensional separable, non-separable, unimodal, and multimodal numerical functions. It is worth noting that both ABC and CLABC fail to overpass the proposed mCLABC in any of those test functions except the F15 test case where the CLABC improved slightly the function mean values.

According to the above analysis and results, it can be concluded that the structural modifications introduced into the CLABC framework contributed to the improvement of the cooperative learning ABC optimization strategy and, at the same time, showed visible superiority in solving middle- and high-dimensional problems.

4.1.2 Comparison of mCLABC with ABC variants

In this numerical experiment, mCLABC is compared to five variants of the ABC algorithm, namely ABC based on modified search strategy (CABC) (Gao et al. 2013a), global-best ABC (ABCbest) (Gao et al. 2012), global-best-guided ABC based on orthogonal learning strategy (OGABC) (Gao et al. 2013a), ABC with multiple search strategies (MuABC) (Gao et al. 2015), and ABC with variable search strategy (ABCVSS) (Kiran et al. 2015). The compared results are

Table 1 Benchmark numerical functions: U unimodal, M multimodal, S separable, N non-separable

Name	No. of Func.	Search range	Type	Function	Accept
Sphere	F1	$[-100, 100]^D$	US	$f_1(x) = \sum_{i=1}^D (x_i)^2$	1×10^{-8}
Elliptic	F2	$[-100, 100]^D$	UN	$f_2(x) = \sum_{i=1}^D (10^6)^{(i-1)/(n-1)} x_i^2$	1×10^{-8}
SumSquares	F3	$[-10, 10]^D$	US	$f_3(x) = \sum_{i=1}^D i x_i^2$	1×10^{-8}
SumPower	F4	$[-10, 10]^D$	MS	$f_4(x) = \sum_{i=1}^D x_i ^{(i+1)}$	1×10^{-8}
Schwefel2.22	F5	$[-10, 10]^D$	UN	$f_5(x) = \sum_{i=1}^D x_i + \prod_{i=1}^D x_i $	1×10^{-8}
Schwefel2.21	F6	$[-100, 100]^D$	UN	$f_6(x) = \max_i \{ x_i , 1 \leq i \leq n\}$	4×10^{-1}
Step	F7	$[-100, 100]^D$	US	$f_7(x) = \sum_{i=1}^D (\lfloor x_i + 0.5 \rfloor)^2$	1×10^{-8}
Quartic	F8	$[-1.28, 1.28]^D$	US	$f_8(x) = \sum_{i=1}^D i x_i^4$	1×10^{-8}
QuarticWN	F9	$[-1.28, 1.28]^D$	US	$f_9(x) = \sum_{i=1}^D i x_i^4 + \text{random}[0, 1)$	1×10^{-1}
Rosenbrock	F10	$[-30, 30]^D$	UN	$f_{10}(x) = \sum_{i=1}^{D-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	5×10^0
Rastrigin	F11	$[-5.12, 5.12]^D$	MS	$f_{11}(x) = \sum_{i=1}^D [x_i^2 - 10 \cos(2\pi x_i) + 10]$	1×10^{-8}
Non-continuous rastrigin	F12	$[-5.12, 5.12]^D$	MS	$f_{12}(x) = \sum_{i=1}^D [y_i^2 - 10 \cos(2\pi y_i) + 10]$ $\begin{cases} x_i x_i < \frac{1}{2} \\ \frac{\text{round}(2x_i)}{2} x_i \geq \frac{1}{2} \end{cases}$	1×10^{-8}
Griewank	F13	$[-600, 600]^D$	MN	$f_{13}(x) = \frac{1}{4000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos(\frac{x_i}{\sqrt{i}}) + 1$	1×10^{-8}
Schaffer	F14	$[-100, 100]^D$	MN	$f_{14}(x) = 0.5 + \frac{\sin^2(\sqrt{x_1^2 + x_2^2}) - 0.5}{(1 + 0.001(x_1^2 + x_2^2))^2}$	1×10^{-1}
Ackley	F15	$[-32, 32]^D$	MN	$f_{15}(x) = -20 \exp\{-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2}\} - \exp\{\frac{1}{D} \sum_{i=1}^D \cos(2\pi x_i)\} + 20 + e$	1×10^{-8}
Penalized1	F16	$[-50, 50]^D$	MN	$f_{16}(x) = \frac{\pi}{D} [10 \sin^2(\pi y_1) + \sum_{i=1}^{D-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] + (y_D - 1)^2] + \sum_{i=1}^D u(x_i, 10, 100, 4) y_i = 1 + \frac{1}{4} (x_i + 1) u_{x_i, a.k.m}$ $= \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a \leq x_i \leq a \\ k(x_i - a)^m & x_i < -a \end{cases}$	1×10^{-8}
Penalized2	F17	$[-50, 50]^D$	MN	$f_{17}(x) = \frac{1}{10} [\sin^2(\pi x_1) + \sum_{i=1}^{D-1} (x_i - 1)^2 [1 + \sin^2(3\pi x_{i+1})] + (x_D - 1)^2 [1 + \sin^2(2\pi x_{i+1})]] + \sum_{i=1}^D u(x_i, 5, 100, 4)$	1×10^{-8}
Alpine	F18	$[-10, 10]^D$	MS	$f_{18}(x) = \sum_{i=1}^D x_i \cdot \sin(x_i) + 0.1 \cdot x_i $	1×10^{-8}
Levy	F19	$[-10, 10]^D$	MN	$f_{19}(x) = \sum_{i=1}^{D-1} (x_i - 1)^2 [1 + \sin^2(3\pi x_{i+1})] + \sin^2(3\pi x_1) + x_D - 1 [1 + \sin^2(3\pi x_D)]$	1×10^{-8}
Weierstrass	F20	$[-0.5, 0.5]^D$	MN	$f_{20}(x) = \sum_{i=1}^D -D \sum_{k=0}^{k_{\max}} [a^k \cos(2\pi b^k (x_i + 0.5))]$ $a = 0.5, b = 3, k_{\max} = 20$	1×10^{-8}
Himmelblau	F21	$[-5, 5]^D$	MS	$f_{21}(x) = \frac{1}{D} \sum_{i=1}^D (x_i^4 - 16x_i^2 + 5x_i)$	-78
Michalewicz	F22	$[0, \pi]^D$	MS	$f_{22}(x) = -\sum_{i=1}^D \sin(x_i) \sin^2\left(\frac{i \cdot x_i^2}{\pi}\right)$	-49, -95, -190

taken directly from Kiran et al. (2015) and Gao et al. (2015). The comparative study is summarized in Table 5 for low-dimensional functions, in Table 6 for middle-dimensional functions and in Table 7 for high-dimensional functions. The results are recorded over 30 independent runs.

Based on the results presented in Table 5, it can be seen that the mCLABC and the compared ABC variants have given same mean function values for F7, F11, F12, F16, F20, and F21. However, mCLABC improves visibly well the results

obtained by the ABC variants on the unimodal functions F1, F2, F3, F4, F5, and F6. For the rest of test functions, mCLABC has better performance than at least three compared algorithms.

For middle-dimensional functions of Table 6, mCLABC shows better results on solving F1, F2, F4, F5, F10 and F15. Same performance with ABC variants is therefore obtained for F7, F11, and F21. For the rest of functions, the mCLABC has equal or better performance than at least three algorithms.

Table 2 Comparison of mCLABC, CLABC, and ABC on solving 15-dimensional functions F1–F20 and 50-dimensional functions F21–F22

Function	ABC		CLABC		mCLABC	
	Mean	Std	Mean	Std	Mean	Std
F1	2.36e−16	6.35e−17	1.03e−66	4.91e−66	1.72e−62	3.04e−62
F2	2.27e−16	3.66e−17	2.73e−63	1.24e−62	1.35e−58	2.37e−58
F3	2.40e−16	4.61e−17	8.89e−70	2.85e−69	5.59e−64	9.97e−64
F4	2.14e−17	8.42e−18	8.31e−135	3.17e−134	7.47e−122	1.89e−121
F5	5.48e−16	7.89e−17	1.50e−38	6.95e−38	5.71e−34	6.82e−34
F6	2.21e−01	7.15e−02	1.56e−03	3.14e−03	1.05e−04	3.95e−04
F7	0	0	0	0	0	0
F8	6.36e−17	1.63e−17	7.52e−122	4.06e−121	2.24e−122	5.44e−122
F9	2.59e−02	9.84e−03	3.15e−03	1.11e−03	3.69e−03	1.52e−03
F10	6.07e−02	7.81e−02	1.30e−03	5.07e−03	1.75e−01	9.34e−01
F11	0	0	0	0	0	0
F12	0	0	0	0	0	0
F13	2.63e−03	4.57e−03	2.69e−04	1.46e−03	1.48e−03	4.50e−03
F14	5.01e−03	4.32e−03	3.89e−03	5.32e−03	2.91e−03	4.53e−03
F15	1.84e−14	3.36e−15	7.14e−15	1.97e−15	7.05e−15	1.85e−15
F16	2.42e−16	4.70e−17	3.14e−32	1.11e−47	3.14e−32	1.11e−47
F17	2.33e−16	4.71e−17	1.34e−32	5.57e−48	1.34e−32	5.57e−48
F18	3.85e−09	5.35e−09	1.60e−12	6.79e−12	9.34e−31	2.24e−30
F19	1.73e−16	6.00e−17	1.34e−31	1.77e−46	1.34e−31	6.68e−47
F20	0	0	1.17e−07	6.42e−07	0	0
F21	−78.33	4.90e−04	−78.33	7.13e−14	−78.33	5.21e−14
F22	−47.22	3.41e−01	−48.30	2.69e+01	−49.11	1.15e−01

Table 3 Comparison of mCLABC, CLABC, and ABC on solving 30-dimensional functions F1–F20 and 100-dimensional functions F21–F22

Function	ABC		CLABC		mCLABC	
	Mean	Std	Mean	Std	Mean	Std
F1	5.90e−16	9.58e−17	3.71e−57	1.89e−56	8.08e−58	1.14e−57
F2	4.98e−16	8.79e−17	2.73e−53	8.74e−53	1.15e−53	3.94e−53
F3	5.53e−16	9.18e−17	7.33e−58	2.86e−57	1.47e−58	1.97e−58
F4	3.39e−17	1.15e−17	3.74e−128	2.60e−127	2.25e−117	1.23e−116
F5	1.31e−15	1.47e−16	3.71e−34	1.55e−33	7.66e−32	9.83e−32
F6	3.86	1.56	3.59	1.21	8.13e−01	5.75e−01
F7	0	0	0	0	0	0
F8	2.17e−16	5.35e−17	2.06e−99	1.05e−98	1.19e−110	7.39e−110
F9	6.02e−02	1.49e−02	8.49e−03	2.90e−03	7.56e−03	3.27e−03
F10	6.25 e−02	9.61e−02	8.12e−03	3.54e−02	5.88e−03	1.66e−02
F11	0	0	0	0	0	0
F12	2.35e−13	1.28e−12	0	0	0	0
F13	9.57e−13	5.24e−12	0	0	0	0
F14	5.73e−03	4.18e−03	1.30e−03	3.36e−03	2.72e−03	4.41e−03
F15	4.40e−14	5.42e−15	1.42e−14	3.56e−15	1.59e−14	3.19e−15
F16	5.22e−16	7.14e−17	1.65e−32	3.19e−33	1.57e−32	5.57e−48
F17	5.16e−16	1.03e−16	1.34e−32	5.57e−48	1.34e−32	5.57e−48
F18	2.45e−07	3.99e−07	2.27e−10	6.45e−10	1.90e−29	4.85e−29
F19	4.43e−16	8.47e−17	2.55e−31	4.55e−31	1.34e−31	6.68e−47
F20	0	0	4.34e−06	1.56e−05	0	0
F21	−78.3308	3.70e−03	−78.3323	1.15e−10	−78.3323	2.64e−13
F22	−93.72	9.01e−01	−95.73	7.46e−01	−97.22	3.11e−01

Table 4 Comparison of mCLABC, CLABC, and ABC on solving 60-dimensional functions F1–F20 and 200-dimensional functions F21–F22

Function	ABC		CLABC		mCLABC	
	Mean	Std	Mean	Std	Mean	Std
F1	1.31e−15	1.27e−16	1.14e−46	6.75e−46	2.68e−53	3.21e−53
F2	1.11e−15	1.69e−16	6.60e−43	1.63e−42	8.50e−50	1.25e−49
F3	1.29e−15	1.49e−16	1.04e−47	2.04e−47	4.50e−54	4.47e−54
F4	4.77e−17	1.62e−17	1.09e−105	7.68e−105	4.57e−108	2.25e−107
F5	2.98e−15	1.96e−16	5.97e−29	1.51e−28	1.00e−29	1.56e−29
F6	2.06e+01	4.18e+00	16.50	2.90	7.19e+00	1.55e+00
F7	0	0	0	0	0	0
F8	5.29e−16	8.03e−17	3.16e−81	7.17e−81	9.11e−102	3.49e−101
F9	1.28e−01	2.26e−02	2.42e−02	6.58e−03	1.48e−02	5.13e−03
F10	1.03e−01	2.34e−01	2.28e−02	8.56e−02	1.87e−02	5.72e−02
F11	0	0	0	0	0	0
F12	0	0	0	0	0	0
F13	2.47e−04	1.35e−03	0	0	0	0
F14	4.25e−03	4.09e−03	2.91e−03	4.69e−03	2.59e−03	4.37e−03
F15	1.04e−13	8.97e−15	3.54e−14	5.83e−15	4.10e−14	6.34e−15
F16	1.27e−15	1.16e−16	8.64e−33	2.13e−33	7.85e−33	2.78e−48
F17	1.30e−15	1.51e−16	1.34e−32	5.57e−48	1.34e−32	5.57e−48
F18	2.66e−05	9.86e−05	9.58e−10	1.12e−09	4.87e−28	7.57e−28
F19	1.10e−15	1.44e−16	1.48e−31	2.96e−32	1.34e−31	6.68e−47
F20	1.89e−15	4.91e−15	8.67e−05	3.38e−04	0	0
F21	−78.31	4.33e−02	−78.32	1.20e−02	−78.33	2.32e−12
F22	−186.712	1.48	190	1.48	−192.635	4.35e−01

From Table 7, the compared results of mCLABC and ABC variants demonstrate again the superiority of mCLABC on solving the most of high-dimensional test functions. Better or slightly improved functions means values are therefore produced by the ABC variants on only five test cases. In addition, it is important to note that the optimal solution for the multimodal function F20 has only been reached by the proposed mCLABC.

Summarizing the above statements, it can be found that mCLABC performed very well on the unimodal functions for low-, middle-, and high-dimensional problems. For the multimodal functions, MuABC and ABCVSS showed better results in solving low-dimensional problems, while mCLABC outperforms the other algorithms for the middle and high-dimensional problems.

4.1.3 Comparison of mCLABC and ABC variants by means of statistical tests

In order to assess the convergence and stability of the proposed mCLABC algorithm, experiments involving statistical tests of mCLABC and ABC variants are conducted on numerical functions optimization problems. The results are reported in Tables 8, 9, and 10 in terms of average number of fitness evaluations (AVEN) and the ratio of the number of success-

ful runs (SR). More precisely, AVEN refers to the average number of the fitness evaluations exhausted when an algorithm reaches the threshold defined as “accept”, and SR refers to the ratio of the number of successful runs, over the total number of runs. Boldface indicates the best results among those obtained by mCLABC and three other compared algorithms, namely CABABC, ABCbest, and MuABC. The results of the compared algorithms are taken directly from Gao et al. (2015). As can be seen from Tables 8, 9, and 10, the proposed method shows a comparatively better convergence behavior than CABABC and ABCbest on all cases. The mCLABC can achieve the 100% success rate on all test functions except the Griewank function (F13) with 15 dimensions, it is competitive to the recent introduced multiple search MuABC algorithm which has 100% success rate on all the cases except the Quartic function (F9) with 60 dimensions; however, in order to get the full success rate, MuABC needed more FEs (AVEN) than mCLABC in the most of test cases. The statistical tests results motivate clearly the robustness of the proposed mCLABC method.

4.2 Experiments on CEC 2005 and CEC 2013 composition functions

Now, to further assess the effectiveness of the proposed mCLABC approach, experiments on CEC functions are

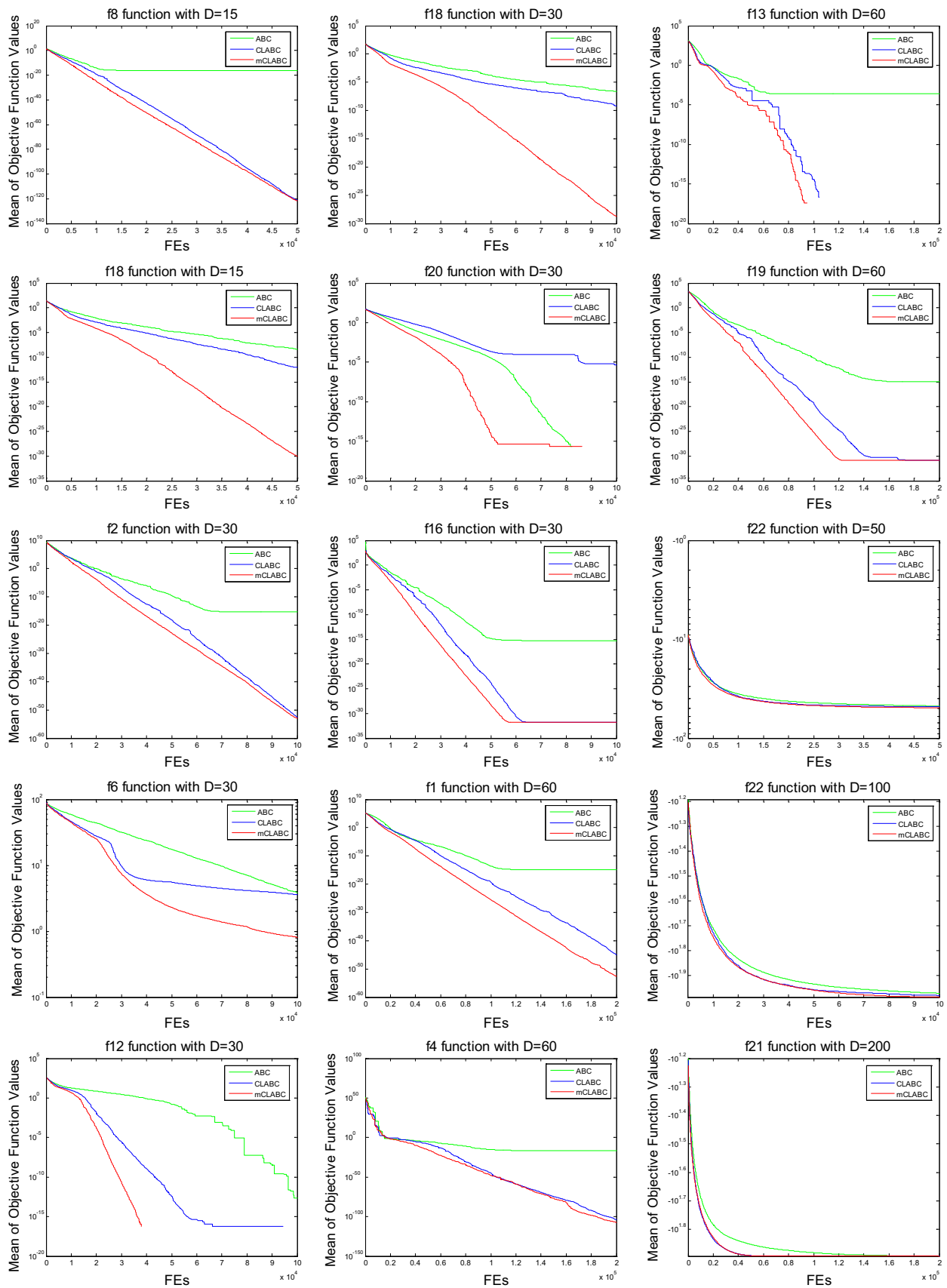


Fig. 6 Convergence curves of basic ABC, CLABC, and mCLABC for selected numerical test functions

Table 5 Comparison of mCLABC and ABCs variants on solving 15-dimensional functions F1–F20 and 50-dimensional functions F21–F22

Function	CABC (Gao et al. 2013a)		ABCbest (Gao et al. 2012)		OGABC (Gao et al. 2013a)		MuABC (Gao et al. 2015)		ABCVSS (Kiran et al. 2015)		mCLABC	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
F1	4.26e-36	4.76e-36	1.27e-34	1.03e-34	3.12e-33	4.85e-33	1.34e-51	1.75e-51	3.01e-61	1.49e-60	1.72e-62	3.04e-62
F2	2.32e-32	3.03e-32	2.59e-31	2.54e-31	2.57e-29	5.56e-29	2.47e-48	2.43e-48	1.21e-53	6.64e-53	1.35e-58	2.37e-58
F3	3.51e-37	6.22e-37	3.17e-36	2.48e-36	2.02e-34	3.04e-34	9.77e-53	5.81e-53	4.91e-60	1.90e-59	5.59e-64	9.97e-64
F4	7.98e-50	2.35e-49	2.00e-56	4.43e-56	1.40e-46	4.72e-46	4.81e-73	7.81e-73	2.67e-63	1.46e-62	7.47e-122	1.89e-121
F5	3.11e-19	3.27e-19	3.92e-19	1.04e-19	1.09e-17	8.70e-18	7.36e-28	5.39e-28	6.14e-29	1.93e-28	5.71e-34	6.82e-34
F6	3.91e-01	1.06e-01	2.06e-01	5.93e-02	5.41e-02	1.54e-02	3.81e-03	1.15e-03	3.13e-03	3.68e-03	1.05e-04	3.95e-04
F7	0	0	0	0	0	0	0	0	0	0	0	0
F9	1.62e-02	7.26e-03	6.70e-03	1.91e-03	3.64e-03	1.33e-03	4.82e-03	1.25e-03	9.28e-03	3.13e-03	3.69e-03	1.52e-03
F10	1.58e-01	1.37e-01	2.48e-00	4.60e-00	5.32e-01	9.43e-01	2.12e-02	6.35e-02	2.57e-01	7.91e-01	1.75e-01	9.34e-01
F11	0	0	0	0	0	0	0	0	0	0	0	0
F12	0	0	0	0	0	0	0	0	0	0	0	0
F13	9.39e-10	4.69e-09	9.34e-03	1.41e-02	9.85e-04	2.75e-03	8.36e-11	1.44e-11	9.04e-04	2.78e-03	1.48e-03	4.50e-03
F15	1.03e-14	3.06e-15	1.06e-14	2.51e-15	3.49e-14	1.21e-14	1.42e-15	2.65e-15	8.88e-15	2.91e-15	7.05e-15	1.85e-15
F16	3.14e-32	1.11e-47	3.14e-32	0	3.14e-32	1.11e-47	3.14e-32	0	3.14e-32	1.11e-47	3.14e-32	1.11e-47
F17	1.34e-32	5.58e-48	1.34e-32	0	1.46e-32	2.42e-33	1.34e-32	0	1.34e-32	5.57e-48	1.34e-32	5.57e-48
F18	4.57e-20	4.71e-20	3.99e-19	4.34e-19	1.26e-16	3.36e-16	3.70e-29	6.41e-29	3.30e-32	1.63e-31	9.34e-31	2.24e-30
F19	1.34e-31	2.23e-47	1.34e-31	0	3.08e-30	8.34e-30	1.34e-31	0	1.34e-31	6.68e-47	1.34e-31	6.68e-47
F20	0	0	0	0	0	0	0	0	0	0	0	0
F21	-7.83e+01	9.59e-10	-78.3323	1.66e-08	-7.83e+01	2.81e-11	-78.3323	4.92e-14	-7.83e+01	1.51e-12	-78.33	5.21e-14
F22	-4.87e+01	2.06e-01	-48.0990	3.59e-01	-4.79e+01	2.72e-01	-49.2637	7.52e-02	-4.94e+01	2.32e-01	-49.11	1.15e-01

Table 6 Comparison of mCLABC and ABCs variants on solving 30-dimensional functions F1–F20 and 100-dimensional functions F21–F22

Function	CABC (Gao et al. 2013a)		ABCbest (Gao et al. 2012)		OGABC (Gao et al. 2013a)		MuABC (Gao et al. 2015)		ABCVSS (Kiran et al. 2015)		mCLABC	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
F1	5.41e-35	7.02e-35	5.30e-32	3.79e-32	4.69e-38	5.27e-38	3.30e-50	1.77e-50	1.20e-57	5.01e-57	8.08e-58	1.14e-57
F2	4.67e-31	6.68e-31	4.27e-28	6.66e-28	5.38e-34	9.25e-34	3.33e-47	2.63e-47	2.44e-52	1.34e-51	1.15e-53	3.94e-53
F3	7.20e-36	7.79e-36	7.33e-33	3.18e-33	3.42e-38	1.02e-37	2.95e-51	3.87e-51	9.53e-61	4.52e-60	1.47e-58	1.97e-58
F4	3.24e-48	7.87e-48	3.24e-57	7.25e-57	7.36e-57	2.60e-56	3.30e-70	5.72e-70	1.29e-78	5.77e-78	2.25e-117	1.23e-116
F5	1.43e-18	5.68e-19	2.91e-17	1.01e-17	5.33e-20	3.13e-20	1.76e-27	8.16e-28	1.47e-27	7.90e-27	7.66e-32	9.83e-32
F6	6.01e+00	8.80e-01	4.75e+00	1.02e+00	5.84e-01	1.83e-01	2.65e+00	4.91e-02	5.09e-01	3.55e-01	8.13e-01	5.75e-01
F7	0	0	0	0	0	0	0	0	0	0	0	0
F9	4.98e-02	1.78e-02	3.80e-02	2.25e-03	5.09e-03	2.07e-03	2.79e-02	2.51e-03	2.46e-02	9.49e-03	7.56e-03	3.27e-03
F10	1.96e-01	1.36e-01	2.53e+01	3.27e+01	1.10e+00	1.74e+00	1.63e-01	2.46e-01	4.92e-01	1.15e+00	5.88e-03	1.66e-02
F11	0	0	0	0	0	0	0	0	0	0	0	0
F12	0	0	0	0	1.18e-09	5.30e-09	0	0	0	0	0	0
F13	4.93e-04	2.20e-03	6.28e-11	2.71e-11	0	0	4.35e-14	7.51e-14	0	0	0	0
F15	3.07e-14	2.88e-15	6.41e-14	3.17e-15	1.93e-14	4.07e-15	3.19e-14	6.15e-15	2.84e-14	5.02e-15	1.59e-14	3.19e-15
F16	1.57e-32	2.80e-48	1.59e-32	5.77e-34	1.57e-32	2.80e-48	1.57e-32	0	1.57e-32	5.57e-48	1.57e-32	5.57e-48
F17	1.34e-32	2.80e-48	2.97e-32	1.75e-32	1.34 e-32	2.80e-48	1.34 e-32	0	1.34e-32	5.57e-48	1.34e-32	5.57e-48
F18	3.53e-19	2.66e-19	7.02e-17	4.23e-17	7.22e-17	1.90e-16	7.40e-28	1.28e-28	1.30e-29	4.89e-29	1.90e-29	4.85e-29
F19	1.34e-31	0	6.20e-31	4.64e-31	1.34e-31	0	1.34e-31	0	1.34e-31	6.68e-47	1.34e-31	6.68e-47
F20	0	0	1.42e-15	3.17e-15	0	0	0	0	0	0	0	0
F21	-7.83e+01	1.63e-09	-78.3323	4.50e-08	-7.83e+01	8.60e-11	-78.3323	1.06e-13	-7.83e+01	3.80e-13	-78.3323	2.64e-13
F22	-9.56e+01	4.24e-01	-93.4167	6.48e-01	-9.52e+01	6.15e-01	-98.6923	1.24e-01	-9.85e+01	5.68e-01	-97.22	3.11e-01

Table 7 Comparison of mCLABC and ABCs variants on solving 60-dimensional functions F1–F20 and 200-dimensional functions F21–F22

Function	CABC (Gao et al. 2013a)		ABCbest (Gao et al. 2012)		OGABC (Gao et al. 2013a)		MuABC (Gao et al. 2015)		ABCVSS (Kiran et al. 2015)		mCLABC	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
F1	2.63e-34	2.68e-34	1.27e-29	9.42e-30	7.99e-44	9.34e-44	5.73e-49	8.10e-49	5.24e-53	2.86e-52	2.68e-53	3.21e-53
F2	3.43e-30	3.60e-30	5.68e-26	5.14e-26	7.09e-40	6.85e-40	2.76e-46	1.31e-46	1.68e-51	7.53e-51	8.50e-50	1.25e-49
F3	1.64e-34	1.45e-34	4.39e-30	4.08e-30	6.02e-45	5.84e-45	7.36e-50	5.88e-50	4.06e-53	1.58e-52	4.50e-54	4.47e-54
F4	2.65e-47	5.54e-47	1.36e-58	2.75e-58	3.05e-70	3.43e-70	2.47e-68	2.56e-68	8.18e-74	3.55e-73	4.57e-108	2.25e-107
F5	5.19e-18	2.10e-18	5.26e-16	1.67e-16	7.64e-23	9.40e-24	7.73e-27	4.20e-27	1.76e-27	7.06e-27	1.00e-29	1.56e-29
F6	3.99e+01	7.62e+00	3.28e+01	3.64e+00	5.19e+00	1.37e+00	3.13e+01	2.14e+00	6.01e+00	1.04e+00	7.19e+00	1.55e+00
F7	0	0	0	0	0	0	0	0	0	0	0	0
F9	1.99e-01	6.07e-02	1.63e-01	2.93e-02	7.64e-03	1.95e-03	1.16e-01	5.88e-03	5.76e-02	1.16e-02	1.48 e-02	5.13e-03
F10	3.27e-01	6.37e-01	2.38+01	3.15+01	1.49e+00	1.27e+00	2.14e-02	2.19e-02	3.68e-01	8.48e-01	1.87 e-02	5.72e-02
F11	0	0	0	0	2.63e-13	1.31e-12	0	0	0	0	0	0
F12	0	0	0	0	9.78e-14	4.26e-13	0	0	0	0	0	0
F13	1.49e-08	3.31e-08	1.79e-07	3.83e-07	2.95e-04	1.47e-03	7.58e-15	9.77e-15	0	0	0	0
F15	8.31e-14	6.92e-15	9.66e-14	5.94e-15	7.81e-15	1.58e-15	2.28e-14	3.10e-15	7.02e-14	6.44e-15	4.10e-14	6.34e-15
F16	7.85e-33	2.79e-48	8.50e-32	4.14e-32	7.85e-33	2.79e-48	7.85e-33	0	7.85e-33	2.78e-48	7.85e-33	2.78e-48
F17	1.34e-32	5.58e-48	1.68e-30	6.17e-31	1.34e-32	5.58e-48	1.34e-32	0	1.34e-32	5.57e-48	1.34e-32	5.57e-48
F18	4.51e-18	8.82e-18	3.61e-14	3.96e-15	3.95e-16	5.76e-16	7.40e-28	1.28e-29	3.49e-27	1.89e-26	4.87e-28	7.57e-28
F19	1.34e-32	2.23e-47	9.46e-30	1.15e-29	1.34e-32	2.23e-47	1.34e-31	0	1.34e-31	6.68e-47	1.34e-31	6.68e-47
F20	8.52e-15	7.78e-15	2.55e-14	6.35e-15	6.87e-05	1.53e-04	8.36e-16	8.20e-15	9.47e-16	3.61e-15	0	0
F21	-7.83e+01	1.70e-09	-78.3323	8.28e-08	-7.83e+01	1.35e-09	-78.3323	1.42e-14	-7.83e+01	5.12e-14	-78.33	2.32e-12
F22	-1.87e+02	8.33e-01	-181.1403	7.73e-01	-1.90e+02	8.73e-01	-196.65	2.84e-01	-1.96e+02	1.42e+00	-192.635	4.35e-01

Table 8 Convergence speed and successful rate comparisons of ABCs on solving 15-dimensional functions F1–F20 and 50-dimensional functions F21–F22

Function	CABC (Gao et al. 2013a)		ABCbest (Gao et al. 2012)		MuABC (Gao et al. 2015)		mCLABC	
	AVEN	SR	AVEN	SR	AVEN	SR	AVEN	SR
F1	1.56e+04	100	1.62e+04	100	1.11e+04	100	9.60e+03	100
F2	2.07e+04	100	2.05e+04	100	1.41e+04	100	1.25e+04	100
F3	1.42e+04	100	1.44e+04	100	1.02e+04	100	8.52e+03	100
F4	5.66e+03	100	4.87e+03	100	4.14e+03	100	5.32e+03	100
F5	2.41e+04	100	2.42e+04	100	1.70e+04	100	1.41e+04	100
F6	1.16e+04	100	3.09e+03	100	2.71e+03	100	1.74e+03	100
F7	6.02e+03	100	6.19e+03	100	4.10e+03	100	3.85e+03	100
F9	3.56e+04	100	4.93e+03	100	3.81e+03	100	3.54e+03	100
F10	1.42e+04	100	1.87e+04	70	1.03e+04	100	9.57e+03	100
F11	1.80e+04	100	1.87e+04	100	1.42e+04	100	1.17e+04	100
F12	1.90e+04	100	1.91e+04	100	1.49e+04	100	1.23e+04	100
F13	2.65e+04	96	3.14e+04	68	2.24e+04	100	2.57e+04	86.67
F15	2.58e+04	100	2.61e+04	100	1.83e+04	100	1.53e+04	100
F16	1.33e+04	100	1.40e+04	100	9.54e+03	100	8.56e+03	100
F17	1.47e+04	100	1.51e+04	100	1.07e+04	100	9.09e+03	100
F18	2.30e+04	100	2.52e+04	100	1.85e+04	100	1.76e+04	100
F19	1.69e+04	100	1.66e+04	100	1.08e+04	100	1.05e+04	100
F20	2.80e+04	100	2.86e+04	100	1.91e+04	100	1.76e+04	100
F21	1.43e+04	100	1.59e+04	100	1.18e+04	100	1.13e+04	100

Table 9 Convergence speed and successful rate comparisons of ABCs on solving 30-dimensional functions F1–F20 and 100-dimensional functions F21–F22

Function	CABC (Gao et al. 2013a)		ABCbest (Gao et al. 2012)		MuABC (Gao et al. 2015)		mCLABC	
	AVEN	SR	AVEN	SR	AVEN	SR	AVEN	SR
F1	3.30e+04	100	3.53e+04	100	2.38e+04	100	1.99e+04	100
F2	4.39e+04	100	4.61e+04	100	2.93e+04	100	2.59e+04	100
F3	3.09e+04	100	3.37e+04	100	2.17e+04	100	1.86e+04	100
F4	1.22e+04	100	9.80e+03	100	5.62e+03	100	1.41e+04	100
F5	5.07e+04	100	5.46e+04	100	3.44e+04	100	2.99e+04	100
F6	5.32e+04	100	1.67e+04	100	1.39e+04	100	1.18e+04	100
F7	1.34e+04	100	1.46e+04	100	8.90e+03	100	8.76e+03	100
F9	3.63e+04	100	3.30e+04	100	1.77e+04	100	1.80e+04	100
F10	3.52e+04	100	5.51e+04	60	2.27e+04	100	2.54e+04	100
F11	4.15e+04	100	4.34e+04	100	3.92e+04	100	5.56e+04	100
F12	4.24e+04	100	4.28e+04	100	3.25e+04	100	2.60e+04	100
F13	4.45e+04	96	4.26e+04	100	3.15e+04	100	4.49e+04	100
F15	5.32e+04	100	5.75e+04	100	3.72e+04	100	3.16e+04	100
F16	2.78e+04	100	3.05e+04	100	1.91e+04	100	1.74e+04	100
F17	3.11e+04	100	3.36e+04	100	2.21e+04	100	1.91e+04	100
F18	5.03e+04	100	5.61e+04	100	3.91e+04	100	3.85e+04	100
F19	3.51e+04	100	2.46e+04	100	2.26e+04	100	2.14e+04	100
F20	5.83e+04	100	6.20e+04	100	3.93e+04	100	4.02e+04	100
F21	3.09e+04	100	4.50e+04	100	2.35e+04	100	2.36e+04	100
F22	9.45e+04	82	–	0	5.43e+04	100	6.87e+04	100

Table 10 Convergence speed and successful rate comparisons of ABCs on solving 60-dimensional functions F1–F20 and 200-dimensional functions F21–F22

Function	CABC (Gao et al. 2013a)		ABCbest (Gao et al. 2012)		MuABC (Gao et al. 2015)		mCLABC	
	AVEN	SR	AVEN	SR	AVEN	SR	AVEN	SR
F1	6.96e+04	100	7.87e+04	100	4.92e+04	100	4.16e+04	100
F2	9.06e+04	100	1.01e+05	100	6.07e+04	100	5.51e+04	100
F3	6.66e+04	100	7.62e+04	100	4.65e+04	100	3.99e+04	100
F4	2.46e+04	100	2.11e+04	100	1.19e+04	100	3.82e+04	100
F5	1.06e+05	100	1.18e+05	100	7.21e+05	100	6.28e+04	100
F6	1.52e+05	62	1.26e+05	100	1.12e+05	100	4.45e+04	100
F7	2.83e+04	100	3.25e+04	100	1.93e+04	100	1.82e+04	100
F9	–	0	–	0	–	0	4.75e+04	100
F10	8.70e+04	100	1.71e+05	22	5.47e+04	100	7.55e+04	100
F11	8.20e+04	100	9.13e+05	100	7.65e+04	100	1.26e+05	100
F12	8.56e+04	100	9.47e+04	100	724e+04	100	5.56e+04	100
F13	8.64e+04	86	8.99e+04	100	6.29e+04	100	1.03e+04	100
F15	1.10e+05	100	1.21e+05	100	7.56e+04	100	6.62e+04	100
F16	5.62e+04	100	6.46e+04	100	3.93e+04	100	3.60e+04	100
F17	6.50e+04	100	7.46e+04	100	4.39e+04	100	3.96e+04	100
F18	1.03e+05	100	1.26e+05	100	7.94e+04	100	8.36e+04	100
F19	7.42e+04	100	7.47e+04	100	4.46e+04	100	4.34e+04	100
F20	1.20e+05	100	1.09e+05	100	8.14e+04	100	8.69e+04	100
F21	6.38e+04	100	7.26e+04	100	4.69e+04	100	4.96e+04	100
F22	–	0	–	0	1.16e+05	100	1.55e+05	100

conducted in this section. For this purpose, composition functions of CEC2005-type (Suganthan et al. 2005) and CEC2013-type (Liang et al. 2013) are used. For each type, unimodal and multimodal composition functions including basic functions, expanded functions, and hybrid functions are considered.

For CEC 2005 test case, the population size is chosen as 40, the maximum number of evaluations is set to 320,000, and the parameter limit is 1000. All functions are tested with dimensionality of 10 and run for 30 times. The optimization results of mCLABC, and the compared ones obtained by binary version ABC based on genetic operators (GB-ABC) (Ozturk et al. 2015), quantum inspired binary PSO (QBPSO) (Chung et al. 2011) and binary version of the ABC (DisABC) (Kashan et al. 2012) are presented in Table 11. From this table, it is easy to notice that mCLABC shows the best performance among the compared algorithms on four unimodal complex composition functions f1, f2, f4, and f5. Here, only the mCLABC can obtain a global optimal solution for f1 and a near-global optimal solution for f4 and f5. For the multimodal functions f6–f17, mCLABC is much better than DisABC, QBPSO, and GB-ABC on f6, f7, f9, and f11–f17 test functions. The mCLABC was able to reach the optimum for two multimodal functions f9 and f15. Besides, as far as we know, there is no algorithm that was able to find global or

near-global optimal solution for any of the composition test functions f1–f17 with the same performance as the proposed mCLABC does.

Besides, mCLABC is compared with recent differential evolution (DE) algorithms on solving CEC 2005 optimization problems. Table 12 shows the obtained results for 30-dimensional composition functions processed over 300,000 evaluations (FEs) with population size set to 60, a parameter limit value of 900 and run for 25 times independently. The comparative study is conducted with reference to the experiments and settings reported in Wang et al. (2014), where the results of some other powerful DEs are provided. More precisely, the performance of mCLABC is compared with DE algorithm based on self-adaptive parameter control (jDE) (Brest et al. 2006), self-adaptive DE (SaDE) (Qin et al. 2009), composite DE (CoDE) (Wang et al. 2011), and DE based on covariance matrix learning and bimodal distribution parameter setting (CoBiDE) (Wang et al. 2014). As can be noticed, comparable results were achieved on most of the CEC 2005 functions with visible improvements on multimodal composition functions. The proposed mCLABC found better results than the compared DE variants on optimizing the basic multimodal functions f6–f12, the expanded multimodal functions f13–f14 and the hybrid composition functions f15, f18–f20, f22, and f25. Our strategy performed

Table 11 Experimental results of ABC variants, QBPSO, and mCLABC on 10-dimensional CEC 2005 test functions

Function	DisABC (Kashan et al. 2012)		QBPSO (Chung et al. 2011)		GB-ABC (Ozturk et al. 2015)		mCLABC	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std
f1	-103.56	142.39	-334.94	78.78	-336.54	112.75	-450	0
f2	-74.43	170.08	-270.45	173.75	-280.57	151.46	-450	2.35e-28
f3	-44.47	161.15	-291.45	153.19	-303.56	145.54	7080.96	22996.50
f4	-64.12	154.92	-332.52	122.57	-341.57	96.855	-450	1.39e-28
f5	83.19	155.51	-115.02	166.90	-136.87	177.21	-309.99	7.32e-13
f6	740.04	183.25	577.09	138.99	533.92	127.21	390	4.17e-25
f7	-149.06	15.47	-171.42	8.82	-167.25	9.06	-179.84	0.163
f8	-119.66	0.08	-119.79	0.07	-119.73	0.09	-119.75	0.057
f9	-309.69	0.07	-309.81	0.06	-309.67	0.11	-330	0
f10	-309.67	0.07	-309.78	0.08	-309.66	0.12	102.64	5.896
f11	110.29	0.09	110.19	0.09	110.32	0.10	93.21	0.999
f12	-439.67	0.08	-439.81	0.07	-439.75	0.09	-456.33	7.15
f13	-109.68	0.08	-109.81	0.09	-109.68	0.12	-129.92	0.074
f14	-279.66	0.08	-279.75	0.08	-279.64	0.09	-297.50	0.437
f15	472.64	48.95	435.10	174.28	412.57	189.24	120	0
f16	472.43	67.26	449.07	160.37	411.04	164.88	240.195	11.99
f17	472.22	59.78	398.99	183.57	367.16	165.15	238.45	10.99

same or better than jDE in 20 tests, SaDE in 19 tests, CoDE in 15 tests, and CoBiDE in 16 tests. Moreover, mCLABC outperforms or is same as the adaptive differential evolution with optional external archive algorithm, named JADE (Wang et al. 2014), in 18 test cases. The results of JADE are not included in Table 12 for the sake of brevity only. Additionally, among all optimization results of JADE, jDE, SaDE, CoDE, and CoBiDE, a noticeable achievement of the proposed mCLABC giving an average value of $9.04e-08$ and a standard deviation value of $2.55e-07$ of the multimodal f6 error values was obtained over 25 independent runs. For this particular test case, the recent DE variant referred to as CoBiDE algorithm (Wang et al. 2014) showed a mean value of $4.13e-02$ and a standard deviation of $9.21e-02$. On the other hand, similar performance to at least one compared new DE algorithm on five test functions is reached by using the modified cooperative learning ABC algorithm.

For CEC 2013 test case, the proposed algorithm is compared to four ABC variants and two DE variants, i.e., ABC with Powell's local search (PABC) (Gao et al. 2013b), modified ABC (MABC) (Akay and Karaboga 2012), compact ABC (comABC) (Dao et al. 2014), enhanced compact ABC (EcABC) (Banitalebi et al. 2015), persistent compact DE (cDE) (Mininno et al. 2011), and memetic compact DE (McDE) (Neri and Mininno 2010) by considering 28 composition functions with $D = 50, 100$. In the experiments, we use the same parameter settings as those of the compared methods reported in Banitalebi et al. (2015). The number of

FEs is set to $5000 * D$. All the results are obtained from 50 independent runs. The experimental results are provided in Tables 13 and 14. As can be seen from Table 13 ($D = 50$) and Table 14 ($D = 100$), the proposed mCLABC was able to get better solutions than the other compared algorithms on 21 test functions with $D = 50$ and 23 test functions with $D = 100$. Specifically, the performance of mCLABC is clearly the best among the six algorithms for the unimodal functions f1–f5. The mCLABC outperforms all the compared algorithms on solving f1, f3, and f5, while EcABC shows better results on only f2 and f4. In addition, it can be seen that only mCLABC can find the near-optimum of f1 and f5 with dimensionality of 100. Moreover, comparison results on the set of multimodal composition functions f6–f20 show us that the EcABC performs better for f16 and f6 (with $D = 50$), while MABC and McDE are the best for f15 with $D = 50$ and $D = 100$, respectively. For the rest of the compared multimodal test functions, mCLABC is visibly superior. Besides, with regard to composition functions f21–f28, PABC outperforms mCLABC for functions f21 and f22 with $D = 50$, and EcABC looks better than mCLABC for only f27 with $D = 100$. However, the performance of mCLABC is same or better than the compared algorithms for the rest of composition functions.

Based on the above comparisons, the superiority of mCLABC on solving complex CEC high-dimensional problems is then clearly justified. In fact, reaching global or near-global optimums of some composition test functions by the proposed method is a great finding.

Table 12 Experimental results of DEs and mCLABC on 30-dimensional CEC 2005 test functions

Fun	jDE (Brest et al. 2006)		SaDE (Qin et al. 2009)		CoDE (Wang et al. 2011)		CoBiDE (Wang et al. 2014)		mCLABC	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
f1	0	0	0	0	0	0	0	0	0	0
f2	1.11e-06	1.96e-06	8.26e-06	1.65e-05	1.69e-15	3.95e-15	1.60e-12	2.90e-12	1.74e-13	3.37e-13
f3	1.98e+05	1.10e+05	4.27e+05	2.08e+05	1.05e+05	6.25e+04	7.26e+04	5.64e+04	1.95e+05	1.38e+05
f4	4.40e-02	1.26e-01	1.77e+02	2.67e+02	5.81e-03	1.38e-02	1.16e-03	2.74e-03	6.89e+02	1.14e+03
f5	5.11e+02	4.40e+02	3.25e+03	5.90e+02	3.31e+02	3.44e+02	8.03e+01	1.51e+02	3.07e+03	6.17e+02
f6	2.35e+01	2.50e+01	5.31e+01	3.25e+01	1.60e-01	7.85e-01	4.13e-02	9.21e-02	9.04e-08	2.55e-07
f7	1.18e-02	7.78e-03	1.57e-02	1.38e-02	7.46e-03	8.55e-03	1.77e-03	3.73e-03	2.37e-02	1.48e-02
f8	2.09e+01	4.86e-02	2.09e+01	4.95e-02	2.01e+01	1.41e-01	2.07e+01	3.75e-01	2.09e+01	6.69e-02
f9	0	0	2.39e-01	4.33e-01	0	0	0	0	0	0
f10	5.54e+01	8.46e+00	4.72e+01	1.01e+01	4.15e+01	1.16e+01	4.41e+01	1.29e+01	1.04e+02	4.47e+01
f11	2.79e+01	1.61e+00	1.65e+01	2.42e+00	1.18e+01	3.40e+00	5.62e+00	2.19e+00	2.36e+01	3.03e+00
f12	8.63e+03	8.31e+03	3.02e+03	2.33e+03	3.05e+03	3.80e+03	2.94e+03	3.93e+03	2.41e+03	2.55e+03
f13	1.66e+00	1.35e-01	3.94e+00	2.81e-01	1.57e+00	3.27e-01	2.64e+00	1.13e+00	8.56e-01	1.60e-01
f14	1.30e+01	2.00e-01	1.26e+01	2.83e-01	1.23e+01	4.81e-01	1.23e+01	4.90e-01	1.23e+01	3.18e-01
f15	3.77e+02	8.02e+01	3.76e+02	7.83e+01	3.88e+02	6.85e+01	4.04e+02	5.03e+01	1.96e+01	3.66e+01
f16	7.94e+01	2.96e+01	8.57e+01	6.94e+01	7.37e+01	5.13e+01	7.38e+01	3.66e+01	2.02e+02	4.19e+01
f17	1.37e+02	3.80e+01	7.83e+01	3.76e+01	6.67e+01	2.12e+01	7.25e+01	2.02e+01	1.18e+02	3.79e+01
f18	9.04e+02	1.08e+01	8.68e+02	6.23e+01	9.04e+02	1.04e+00	9.03e+02	1.05e+01	8.52e+02	6.43e+00
f19	9.04e+02	1.11e+00	8.74e+02	6.22e+01	9.04e+02	9.42e-01	9.03e+02	1.04e+01	8.51e+02	6.12e+00
f20	9.04e+02	1.10e+00	8.78e+02	6.03e+01	9.04e+02	9.01e-01	9.04e+02	5.95e-01	8.50e+02	1.15e+01
f21	5.00e+02	4.80e-13	5.52e+02	1.82e+02	5.00e+02	4.88e-13	5.00e+02	4.62e-13	5.00e+02	4.98e-13
f22	8.75e+02	1.91e+01	9.36e+02	1.83e+01	8.63e+02	2.43e+01	8.62e+02	2.80e+01	7.02e+02	1.66e+02
f23	5.34e+02	2.77e-04	5.34e+02	3.57e-03	5.34e+02	4.12e-04	5.34e+02	1.30e-04	5.34e+02	3.20e-04
f24	2.00e+02	2.85e-14	2.00e+02	6.20e-13	2.00e+02	2.85e-14	2.00e+02	2.85e-14	2.00e+02	1.17e-13
f25	2.11e+02	7.32e-01	2.14e+02	2.00e+00	2.11e+02	9.02e-01	2.10e+02	7.73e-01	2.03e+02	8.89e+00

5 Study of the robustness of the proposed optimization strategy

The demonstrated performance of the proposed mCLABC strategy is achieved on the basis of the optimization process as described by the algorithmic structure, which is given in Fig. 2. Obviously, some settings need to be set in a convenient way for satisfactory results. The adequacy of the settings might result from the study of the robustness and performance of the mCLABC model with respect to a number of design parameters, in particular, the number of subgroups of employed and onlooker foragers, the multi-search strategy, and the mechanism of selecting the number of food features to be altered by the foragers simultaneously at each processing time.

To complete the study, we investigated the effects of all of these design parameters by conducting a number of experiments on five benchmark numerical functions selected as F1, F10, F11, F13 and F15 from Table 1. For this purpose, the test functions are set with dimension 30 and the maximum

number of FEs is chosen as 100,000. The experiments are repeated 30 times.

5.1 Subgroups partitioning effect

The proposed mCLABC strategy employs three subgroups of foragers at both employed and onlooker phases. For analysis purpose, five test cases introducing different structural modifications on the original mCLABC are considered. For clarity, the test case denoted as mCLABC (EBS, OBS) stands for mCLABC strategy with EBS employed subgroups and OBS onlooker subgroups. In the first test case, employed bees are categorized into three subgroups while the onlookers are kept in one group. The second test operates with one group of employed and three subgroups of onlookers. The remaining test cases use equal number of subgroups (2, 4 or 6) of both employed and onlooker bees. The search strategies of the compared variants are chosen same as the original mCLABC.

The results are listed in Table 15 where it appears clearly that the mCLABC is better than mCLABC(3,1)

Table 13 Performance of mCLABC on 50-dimensional CEC 2013 test functions

Fun	PABC (Gao et al. 2013b)		MABC (Akay and Karaboga 2012)		comABC (Dao et al. 2014)		EcABC (Banitalebi et al. 2015)		mCLABC	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
f1	1.364e-12	1.49e-13	1.606e-12	9.18e-13	1.062e-01	6.65e-01	1.591e-12	1.84e-13	2.274e-13	1.45e-13
f2	1.299e+07	9.72e+06	2.080e+07	6.04e+06	5.645e+08	4.10e+07	5.832e+06	3.11e+05	5.784e+06	4.02e+06
f3	1.474e+09	8.25e+08	1.083e+09	2.13e+09	2.899e+11	9.53e+10	6.312e+08	4.35e+07	1.561e+08	2.31e+08
f4	1.443e+05	4.32e+04	1.578e+05	1.94e+04	7.308e+04	8.28e+03	3.398e+04	2.25e+03	1.150e+05	1.89e+04
f5	1.250e-12	8.31e-13	1.477e-12	8.23e-12	1.107e+03	3.64e+02	6.063e-04	2.30e-05	1.137e-13	3.77e-13
f6	4.408e+02	3.99e+01	4.674e+01	7.01e+00	2.244e+03	7.36e+02	3.968e+01	9.39e-01	4.285e+01	5.07e-01
f7	1.636e+02	6.56e+01	1.498e+02	5.16e+01	1.475e+02	8.05e+02	1.856e+02	6.22e+01	1.056e+02	1.46e+01
f8	2.117e+01	9.84e+00	2.123e+01	2.46e+00	2.114e+01	5.21e+00	2.106e+01	3.14e+00	2.115e+01	3.58e-02
f9	6.013e+01	1.06e+00	6.156e+01	4.87e+00	7.791e+01	2.23e+01	7.141e+01	1.44e+01	5.663e+01	4.05e+00
f10	2.338e+00	9.28e-02	2.713e+00	4.92e-01	2.663e+12	4.58e+02	9.705e-02	4.46e-03	8.630e-02	5.80e-02
f11	2.273e-13	1.98e-14	4.705e-13	8.46e-14	7.757e+02	5.73e+01	2.855e+02	9.63e+00	5.684e-14	3.04e-14
f12	3.841e+02	9.51e+01	8.257e+02	1.74e+02	8.840e+02	6.48e+01	8.327e+02	7.35e+01	1.959e+02	5.10e+01
f13	6.861e+02	4.22e+01	7.038e+02	4.18e+01	8.049e+02	6.12e+01	7.561e+02	8.01e+01	3.023e+02	4.45e+01
f14	1.165e+01	4.17e+00	3.409e+00	3.91e-01	1.057e+04	1.18e+04	3.733e+03	1.07e+02	2.124e+00	2.31e+00
f15	8.880e+03	6.01e+02	6.489e+03	2.56e+02	1.404e+04	7.85e+03	6.504e+03	9.17e+02	8.241e+03	6.90e+02
f16	1.967e+00	5.39e+00	1.524e+00	7.92e-01	3.648e+00	5.62e+00	1.201e+00	6.02e-01	2.194e+00	3.44e-01
f17	5.150e+01	1.71e+00	5.104e+01	3.07e+01	1.025e+03	2.03e+03	5.982e+02	4.08e+01	5.079e+01	1.37e-02
f18	5.612e+02	4.04e+01	7.895e+02	2.42e+01	1.237e+03	8.91e+02	1.381e+03	4.38e+02	2.223e+02	3.24e+01
f19	2.272e+00	1.75e-01	9.302e+01	3.76e+02	5.088e+05	3.48e+01	2.504e+01	5.21e+00	7.095e-01	1.99e-01
f20	8.665e+01	8.81e+00	2.457e+01	1.51e+00	2.503e+01	3.73e+00	2.051e+01	6.24e+00	2.217e+01	8.89e-01
f21	2.002e+02	3.26e+01	2.017e+03	9.87e+02	4.509e+03	2.84e+03	1.122e+03	3.67e+02	3.883e+02	3.28e+02
f22	5.996e+01	1.72e+00	2.714e+03	6.23e+02	1.030e+04	7.32e+03	2.562e+03	8.85e+02	4.098e+01	3.01e+01
f23	1.018e+04	4.65e+03	8.459e+04	6.86e+03	1.552e+04	5.08e+03	9.586e+03	2.61e+02	2.316e+03	9.70e+02
f24	3.451e+02	1.06e+01	3.865e+02	2.32e+01	4.453e+02	4.19e+01	3.787e+02	1.36e+01	3.315e+02	1.85e+01
f25	3.763e+02	8.51e+01	4.257e+02	1.58e+01	4.192e+02	1.74e+02	3.866e+02	9.87e+00	3.570e+02	5.64e+00
f26	2.015e+02	3.56e+01	2.022e+02	3.56e+01	5.083e+02	3.67e+01	2.076e+02	1.97e+01	2.003e+02	1.05e-01
f27	1.946e+03	6.98e+02	2.207e+03	1.85e+03	2.524e+03	2.44e+02	1.892e+03	8.90e+01	1.584e+03	4.59e+02
f28	3.782e+02	3.58e+01	4.109e+03	2.81e+02	7.632e+03	2.86e+03	2.116e+03	7.79e+02	4.000e+02	1.16e-12

and mCLABC(6,6) for the multimodal functions F11 and F13, respectively, and again performs in the same way as mCLABC(1,3), mCLABC(2,2) and mCLABC(4,4) for those two multimodal functions (F11 and F13). Also, mCLABC(6,6) outperforms mCLABC on F10 and F15, but easily traps into local optimum when solving the multimodal function F13. Overall, it should be noticed that increasing the number of equal employed and onlooker partitions from 3 to 6 will not have significant improvement on the results.

5.2 Food features alteration effect

During foraging process, employed or onlooker bees modify the features of the food source being exploited. Thus, the number of altered features at a given processing time might

affect considerably the process of food sources exploitation. This in turn might influence the quality of the results. To analyze this aspect, four food features alteration strategies are tested. The first strategy labeled as “Single” adopts an updating mechanism that alters one random parameter only for all processing times. The second referred to as “Full” modifies all dimensions of a food source simultaneously. The strategy “Random” selects randomly the number of simultaneous evolving parameters in the range $[1, D]$, while the strategy “Combined” combines the mechanisms of the above strategies according to the following rule: employed bees are managed to start by changing one random dimension. Then, when the number of iterations reaches $iter = MCN/5$, the algorithm will have to select randomly the number of parameters to be optimized from $[1, D]$. While reaching the end

Table 14 Performance of mCLABC on 100-dimensional CEC 2013 test functions

Fun	cDE (Mininno et al. 2011)		McDE (Neri and Mininno 2010)		comABC (Dao et al. 2014)		EcABC (Banitalebi et al. 2015)		mCLABC	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
f1	4.063e-02	8.14e-03	1.077e+05	9.02e+04	3.150e+00	4.17e-01	1.294e-02	2.78e-03	2.273e-13	4.97e-13
f2	1.986e+08	9.13e+07	1.169e+09	4.89e+08	1.840e+09	2.93e+08	5.617e+06	9.57e+06	1.337e+07	5.80e+06
f3	5.179e+10	3.71e+10	1.790e+16	3.37e+16	4.740e+14	3.69e+13	3.058e+09	1.57e+09	1.538e+09	1.20e+09
f4	1.071e+05	7.15e+04	1.879e+05	2.41e+04	2.548e+05	4.03e+04	7.721e+03	4.85e+04	2.171e+05	3.27e+04
f5	4.209e-02	9.33e-02	1.336e+04	9.56e+03	5.688e-01	5.75e-01	1.755e-02	9.59e-01	5.684e-13	8.15e-13
f6	5.598e+03	3.92e+02	2.735e+04	1.68e+04	1.436e+04	6.49e+05	1.548e+02	6.78e+02	1.886e+02	2.33e+01
f7	8.156e+04	7.06e+03	1.452e+04	7.31e+03	3.175e+06	4.51e+06	3.893e+05	2.76e+05	1.407e+02	2.11e+01
f8	2.129e+01	6.94e+00	3.311e+01	6.44e+00	2.323e+01	2.96e+00	2.132e+01	8.23e+00	2.129e+01	3.32e-02
f9	1.397e+02	4.38e+02	1.605e+02	3.68e+01	4.640e+02	1.83e+01	1.495e+02	7.95e+01	1.335e+02	5.58e+00
f10	2.786e+03	6.46e+02	7.107e+03	6.25e+03	1.356e+04	7.85e+03	5.998e+02	1.86e+02	8.44e-02	6.58e-02
f11	1.021e+03	6.79e+02	9.425e+02	7.75e+02	8.959e+02	4.86e+02	2.147e+02	7.09e+02	5.684e-14	1.32e-13
f12	1.239e+03	4.98e+03	2.063e+03	5.08e+02	3.762e+03	3.06e+02	1.994e+03	6.55e+01	5.978e+02	7.07e+01
f13	1.921e+03	2.55e+02	2.267e+03	7.94e+02	2.685e+03	8.71e+03	2.713e+03	5.85e+02	8.179e+02	8.10e+01
f14	9.034e+03	5.05e+03	1.014e+04	8.01e+03	1.391e+04	2.78e+04	1.051e+03	5.47e+02	3.532e+00	1.59e+00
f15	1.746e+04	8.48e+03	1.126e+04	3.52e+03	2.058e+04	9.39e+03	1.934e+04	1.38e+02	1.701e+04	1.19e+03
f16	3.106e+00	2.54e+00	3.636e+00	8.75e-01	4.351e+00	5.54e-01	1.797e+00	9.29e-01	2.696e+00	3.17e-01
f17	1.221e+03	6.16e+02	2.561e+03	3.01e+02	2.404e+03	2.07e+02	1.064e+03	3.49e+01	1.016e+02	4.06e-02
f18	3.085e+03	4.73e+03	2.389e+03	8.41e+02	2.558e+03	2.31e+03	2.721e+03	5.85e+02	5.789 e+02	7.13e+01
f19	3.006e+02	5.49e+01	1.253e+05	1.75e+04	6.744e+05	2.25e+04	8.191e+01	7.57e+00	1.898e+00	3.89e-01
f20	5.00e+01	0.00e+00	5.00e+01	0.00e+00	5.00e+01	0.00e+00	5.00e+01	0.00e+00	5.00e+01	0.00e+00
f21	4.267e+02	9.34e+01	7.739e+03	1.84e+02	3.067e+03	4.21e+02	4.261e+02	7.79e+01	3.820e+02	3.88e+01
f22	1.095e+04	4.69e+04	1.346e+04	4.38e+03	1.777e+04	9.04e+03	2.082e+04	7.94e+03	9.780e+01	4.75e+01
f23	2.279e+04	3.11e+03	3.345e+04	2.58e+03	3.442e+04	4.08e+03	2.214e+04	6.01e+03	2.308e+04	1.66e+03
f24	7.651e+02	6.89e+01	7.355e+02	6.02e+02	6.791e+02	7.11e+01	6.163e+02	7.48e+00	5.274e+02	3.43e+01
f25	7.820e+02	4.50e+02	6.107e+02	3.21e+01	6.476e+02	1.17e+01	6.332e+02	1.52e+01	5.574e+02	1.12e+01
f26	6.624e+02	9.61e+01	7.052e+02	5.07e+01	7.445e+02	4.24e+01	2.076e+02	2.59e+01	2.009e+02	2.72e-01
f27	2.521e+03	4.31e+02	4.542e+03	2.62e+02	5.128e+03	8.01e+02	3.204e+03	2.63e+02	3.571e+03	8.43e+02
f28	1.508e+04	3.49e+03	1.446e+04	5.78e+03	1.815e+04	4.83e+03	1.417e+04	5.79e+03	3.222e+03	7.98e+02

of the optimization process, i.e., between $iter = \frac{4}{5}MCN$ and $iter = MCN$, the employed bees will update all parameters of the solution. Subsequently, after completion of the employed bee phase, the three subgroups of onlookers will start to update one random parameter of the solution. Then, when $iter = MCN/5$, the onlooker bees will have to select randomly the number of parameters to be optimized from $[1, D]$. The search strategies in those test cases are same as mCLABC.

The results are compared in Table 16. As can be seen, mCLABC that alters all parameters simultaneously (full) is unable to outperform the original mCLABC on any test function. The mCLABC with “Single” strategy and mCLABC with “Random” strategy show slightly improved or same results as the mCLABC on only one test function. The

mCLABC with combined strategy overpasses the mCLABC on F1 and F15, while the mCLABC is better on F10, F11, and F13, and finds the global optimal solution for F11 and F13.

This analysis proves that the process used in mCLABC to select the number of food features to be altered simultaneously at each iteration provides better results. The strategy to select the number of parameters to be optimized by processing time operates as a scheduling mechanism that contributes to the enhancement of the overall mCLABC performance.

5.3 Search strategy effect

The effect of solution updating rules on the mCLABC algorithm is now investigated through three different variants

Table 15 Performance analysis of mCLABC by means of foragers subgroups partitioning

Fun	mCLABC (3,1)		mCLABC (1,3)		mCLABC (2,2)		mCLABC (4,4)		mCLABC (6,6)		mCLABC	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
F1	1.50e-55	2.88e-55	1.06e-50	2.81e-50	6.38e-54	1.08e-53	9.54e-57	2.15e-56	8.77e-57	1.11e-56	8.08e-58	1.14e-57
F10	2.96e-02	6.02e-02	1.37e-01	7.40e-01	1.66e-01	7.25e-01	7.37e-03	3.02 e-02	4.91e-03	2.06e-02	5.88e-03	1.66e-02
F11	1.18e-16	6.49e-16	0	0	0	0	0	0	0	0	0	0
F13	0	0	0	0	0	0	0	0	3.29e-04	1.80e-03	0	0
F15	1.81e-14	3.97e-15	1.81e-14	4.18e-15	2.24e-14	5.35e-15	1.47e-14	3.53e-15	1.49e-14	2.08e-15	1.59e-14	3.19e-15

Table 16 Performance analysis of mCLABC by means of food features alteration strategy

Function	Food feature alteration strategy type of mCLABC									
	Single		Full		Random		Combined		Proposed	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
F1	3.60e-57	1.23e-56	5.85e-18	2.01e-17	9.07e-74	4.52e-73	4.93e-75	2.20e-74	8.08e-58	1.14e-57
F10	2.77e-01	7.46e-01	42.29	27.71	14.26	20.02	4.79e-01	1.92	5.88e-03	1.66e-02
F11	0	0	32.07	8.86	7.66	2.54	2.37e-16	7.71e-16	0	0
F13	1.53e-09	8.35e-09	3.31e-01	4.76e-01	2.44e-02	2.66e-02	7.40e-18	4.05e-17	0	0
F15	3.30e-14	3.90e-15	7.49	1.68	1.20	8.97e-01	1.08e-14	4.00e-15	1.59e-14	3.19e-15

Table 17 Performance analysis of mCLABC by means of search updating rule strategy

Function	Multi-search strategy of mCLABC							
	SS1		SS2		SS3		Proposed	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std
F1	4.91e-57	8.57e-57	1.11e-56	1.82e-56	5.21e-56	2.10e-55	8.08e-58	1.14e-57
F10	1.05e-01	5.38e-01	2.80e-01	1.48e+00	2.79e-01	1.04e+00	5.88e-03	1.66e-02
F11	0	0	0	0	0	0	0	0
F13	0	0	4.93e-04	1.88e-03	2.47e-04	1.35e-03	0	0
F15	1.58e-14	4.32e-15	1.44e-14	3.54e-15	1.27e-14	3.77e-15	1.59e-14	3.19e-15

using multiple search strategies labeled as SS1 (search strategy 1), SS2 (search strategy 2) and SS3 (search strategy 3). The SS1 uses three global search equations, i.e., Eqs. (4), (5) and (6) for employed subgroups 1, 2 and 3, respectively, while onlooker foragers subgroups 1, 2 and 3 employ the local search equations Eqs. (7), (8), and (9), respectively. The search mechanism will call the whole solution updating rules at each iteration. Similarly, according to the search mechanism defined in SS2, a search equation from the set described above could be called more than one time in a given processing time or even could not be used at all. For SS3, the Eqs. (4), (7) and (8) are used by the employed subgroups 1, 2 and 3, respectively and the Eqs. (5), (6) and (9) by the onlooker subgroups 1, 2 and 3, respectively. The strategy of selecting the evolving parameters in those test cases is same as mCLABC. The results are shown in Table 17.

According to the obtained results, it can be concluded that whether SS1, SS2 or SS3 is used, the mCLABC algorithm shows superiority on functions F1 and F10. On the other hand, mCLABC and mCLABC with SS1 were able to determine the global optimum of functions F11 and F13. The proposed mCLABC gives generally better results on all test functions except for the case F15 where the solutions are slightly different.

From this thorough analysis and based on the results shown in Tables 15, 16 and 17, it is important to notice that the design parameters of the cooperative learning ABC optimization strategy influence differently the quality of the obtained solutions. Although the original mCLABC shows superior-

ity in the most of test cases, the analysis of the impact of structural and parameter settings being introduced through different tests demonstrate the robustness and the merits of the proposed optimization strategy.

6 Conclusion

Aiming to further improve the performance of ABC optimization, a modified cooperative foraging learning strategy based on multiple solution search rules is developed in the present study. The original scheme of generating subgroups of employed foragers adopted in the CLABC framework is extended to the onlooker phase with modified structural and parametric settings. Since the feasibility of the approach can be fully motivated upon natural behavioral principles of bee swarms, it was of major importance for us to consider its integration to the algorithmic structure of the original CLABC algorithm with making all necessary tunings and modifications so that higher computational performance and robustness can be achieved. The modified cooperative foraging strategy, referred to as mCLABC, involves local and global search rules through which separate subgroups of foragers evolve differently and simultaneously at both employed or onlooker stages with allowing mutual exchange of information about the foraging process. To assess the extent of improvement, the proposed mCLABC algorithm is compared to ABC and CLABC algorithms, and to other recent ABCs variants on solving numerical and composition func-

tions optimization problems. From different simulations, it can be concluded that the mCLABC improved considerably well the quality of the solutions and the convergence performance in the most of numerical experiments. Moreover, the superiority of the proposed mCLABC algorithm on solving complex CEC high-dimensional problems was visibly justified, where reaching global or near-global optimums of some composition test functions represents a great finding. To complete the study, a robustness analysis is conducted in order to evaluate the impact of different parametric and structural settings on the performance of the proposed strategy, namely the partitioning scheme of foragers, the search strategy and the food sources alteration mechanism. Future works will be devoted to the parallelization of the proposed algorithm with application to intelligent systems design.

Compliance with ethical standards

Conflict of interest Mrs F. Harfouchi declares that she has no conflict of interest. Prof. H. Habbi declares that he has no conflict of interest. Dr. C. Ozturk declares that he has no conflict of interest. Prof. D. Karaboga declares that he has no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

- Akay B, Karaboga D (2012) A modified artificial bee colony algorithm for real-parameter optimization. *Inf Sci* 192:120–142
- Alatas B (2010) Chaotic bee colony algorithms for global numerical optimization. *Expert Syst Appl* 37(08):5682–5687
- Aydođdu İ, Akin A, Saka MP (2016) Design optimization of real world steel space frames using artificial bee colony algorithm with Levy flight distribution. *Adv Eng Softw* 92:1–14
- Babaoglu I (2015) Artificial bee colony algorithm with distribution-based update rule. *Appl Soft Comput* 34:851–861
- Banharsakun A, Achalakul T, Sirinaovakul B (2011) The best-so-far selection in artificial bee colony algorithm. *Appl Soft Comput* 11:2888–2901
- Banitalebi A, Abd Aziz MI, Bahar A, Abdul Aziz Z (2015) Enhanced compact artificial bee colony. *Inf Sci* 298:491–511
- Biswas S, Das S, Debchoudhury S, Kundu S (2014) Co-evolving bee colonies by forager migration: a multi-swarm based artificial bee colony algorithm for global search space. *Appl Math Comput* 232:216–234
- Bose D, Biswas S, Vasilakos AV, Laha S (2014) Optimal filter desing using an improved artificial bee colony algorithm. *Inf Sci* 281:443–461
- Brest J, Greiner S, Boskovic B, Mernik M, Zumer V (2006) Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems. *IEEE Trans Evol Comput* 10(6):646–657
- Chung CY, Han Y, Kit-Po W (2011) An advanced quantum-inspired evolutionary algorithm for unit commitment. *IEEE Trans Power Syst* 26:847–854
- Dao TK, Chu SC, Nguyen TT, Shieh CS, Horng MF (2014) Compact artificial bee colony. In: Ali M, Pan JS, Chen SM, Horng MF (eds) *Modern Advances in Applied Intelligence*. IEA/AIE 2014. Lecture Notes in Computer Science, vol 8481. Springer, Cham
- Dorigo M, Stutzle T (2004) *Ant colony optimization*. MIT Press, Cambridge
- Fogel DB (1995) *Evolutionary computation: toward a new philosophy of machine intelligence*. IEEE Press, New York
- Gao W, Liu S (2011) Improved artificial bee colony algorithm for global optimization. *Inf Process Lett* 111(17):871–882
- Gao W, Liu S, Huang L (2012) A global best artificial bee colony algorithm for global optimization. *J Comput Appl Math* 236(11):2741–2753
- Gao WF, Liu SY, Huang LL (2013a) A novel artificial bee colony algorithm based on modified search equation and orthogonal learning. *IEEE Trans Cybern* 43(3):1011–1024
- Gao W, Lui S, Huang L (2013b) A novel artificial bee colony algorithm with powell's method. *Appl Soft Comput* 13(9):3763–3775
- Gao WF, Liu SY, Huang LL (2014) Enhancing artificial bee colony algorithm using more information-based search equations. *Inf Sci* 270(20):112–133
- Gao WF, Huang LL, Liu SY, Chan FTS, Dai C, Shan X (2015) Artificial bee colony algorithm with multiple search strategies. *Appl Math Comput* 271:269–287
- Habbi H (2012) Artificial bee colony optimization algorithm for TS-type fuzzy systems learning. In: 25th international conference of European chapter on combinatorial optimization, Antalya, Turkey
- Habbi H, Boudouaoui Y (2014) Hybrid artificial bee colony and least squares method for rule-based systems learning. *Waset Int J Comput Control Quantum Inf Eng* 08:1968–1971
- Habbi H, Boudouaoui Y, Ozturk C, Karaboga D (2015) Fuzzy rule-based modeling of thermal heat exchanger dynamics through swarm bee colony optimization. In: International conference on advanced technology and sciences, ICAT'2015
- Habbi H, Boudouaoui Y, Karaboga D, Ozturk C (2015) Self-generated fuzzy systems design using artificial bee colony optimization. *Inf Sci* 295:145–159
- Harfouchi F, Habbi H (2016) A cooperative learning artificial bee colony algorithm with multiple search mechanisms. *Int J Hybrid Intell Syst* 13:113–124
- Hsieh TJ, Hsiao HF, Yeh WC (2011) Forecasting stock markets using wavelet transforms and recurrent neural networks: an integrated system based on artificial bee colony algorithm. *Appl Soft Comput* 11(02):2510–2525
- Jadhav HT, Bamane PD (2016) Temperature dependent optimal power flow using g-best guided artificial bee colony algorithm. *Electr Power Energy Syst* 77:77–90
- Karaboga D (2005) An idea based on honey swarm for numerical optimization. Technical report-TR06. Erciyes University, Engineering Faculty, Computer Engineering Department
- Karaboga D, Akay B (2009) A comparative study of artificial bee colony algorithm. *Appl Math Comput* 214:108–132
- Karaboga D, Basturk B (2008) On the performance of artificial bee colony (ABC) algorithm. *Appl Soft Comput* 8:687–697
- Karaboga D, Ozturk C (2009) Neural networks training by artificial bee colony algorithm on pattern classification. *Neural Netw World* 19(3):279–292
- Karaboga D, Ozturk C (2011) A novel clustering approach: artificial bee colony (ABC) algorithm. *Appl Soft Comput* 11:652–657
- Kashan MH, Nahavandi N, Kashan AH (2012) DisABC: a new artificial bee colony algorithm for binary optimization. *Appl Soft Comput* 12:342–352
- Kennedy J, Eberhart RC (1995) Particle swarm optimization. *Proceeding of the IEEE international conference on neural networks*. Perth, Australia, pp 1942–1948
- Kiran MS, Findik O (2014) A directed artificial bee colony algorithm. *Appl Soft Comput* 26:454–462
- Kiran MS, Hakli H, Gunduz M, Uguz H (2015) Artificial bee colony algorithm with variable search strategy for continuous optimization. *Inf Sci* 300:140–157

- Li X, Yang G (2016) Artificial bee colony algorithm with memory. *Appl Soft Comput* 41:362–372
- Li G, Niu P, Xiao X (2012) Development and investigation of efficient artificial bee colony algorithm for numerical function optimization. *Appl Soft Comput* 12(01):320–332
- Liang JJ, Qu BY, Suganthan PN, Hernández-Díaz AG (2013) Problem definitions and evaluation criteria for the CEC 2013 special session on real-parameter optimization. Technical report 201212, Computational Intelligence Laboratory, Zhengzhou University and technical report, Nanyang Technological University, Singapore
- Liang JH, Lee CH (2015) Efficient collision-free path-planning of multiple mobile robot system using efficient artificial bee colony algorithm. *Adv Eng Softw* 79:47–56
- Mininno E, Cupertino F, Naso D (2011) Compact differential evolution. *IEEE Trans Evol Comput* 15(1):203–219
- Mohamed AW (2015) An improved differential evolution algorithm with triangular mutation for global numerical optimization. *Comput Ind Eng* 85:359–375
- Neri F, Mininno E (2010) Memetic differential evolution for cartesian robot control. *IEEE Comput Intell Mag* 5(2):54–65
- Okdem S, Karaboga D, Ozturk C (2011) An application of wireless sensor network routing based on artificial bee colony algorithm. *IEEE Congr Evol Comput* 326–330
- Ozturk C, Hancer E, Karaboga D (2015) A novel artificial bee colony algorithm based on genetic operators. *Inf Sci* 297:154–170
- Qin AK, Huang VL, Suganthan PN (2009) Differential evolution algorithm with strategy adaptation for global numerical optimization. *IEEE Trans Evol Comput* 13(2):398–417
- Saffari H, Sadeghi S, Khoshzat M, Mehregan P (2016) Thermodynamic analysis and optimization of a geothermal Kalina cycle system using artificial bee colony algorithm. *Renew Energy* 89:154–167
- Secui DC (2015) A new modified artificial bee colony algorithm for the economic dispatch problem. *Energy Convers Manag* 89:43–62
- Sonmez M (2011) Artificial bee colony algorithm for optimization of truss structures. *Appl Soft Comput* 11(02):2406–2418
- Storn R, Price K (1997) Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *J Glob Optim* 11:341–359
- Sun H, Luş H, Betti R (2013) Identification of structural models using a modified artificial bee colony algorithm. *Comput Struct* 116:59–74
- Suganthan PN, Hansen N, Liang JJ, Deb K, Chen YP, Auger A, Tiwari S (2005) Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization. Nanyang Technol. Univ., Singapore, and IIT Kanpur, Kanpur, India, KanGAL report #2005005
- Szeto WY, Wu YZ, Ho SC (2015) An artificial bee colony algorithm for the capacitated vehicle routing problem. *Eur J Oper Res* 215:126–135
- Taheri J, Lee YC, Zomaya AY, Siegel HJ (2013) A Bee Colony based optimization approach for simultaneous job scheduling and data replication on grid environments. *Comput. Oper. Res.* 40(6):1564–1578
- Wang Y, Cai Z, Zhang Q (2011) Differential evolution with composite trial vector generation strategies and control parameters. *IEEE Trans Evol Comput* 15(1):55–66
- Wang H, Wu Z, Rahnamayan S, Sun H, Liu Y, Pan JS (2014) Multi-strategy ensemble artificial bee colony algorithm. *Inf Sci* 279:587–603
- Wang Y, Li HX, Huang T, Li L (2014) Differential evolution based on covariance matrix learning and bimodal distribution parameter setting. *Appl Soft Comput* 18:232–247
- Xiang W, Ma S, An M (2014) hABCDE: a hybrid evolutionary algorithm based on artificial bee colony algorithm and differential evolution. *Appl Math Comput* 238:370–386
- Yuan X, Wang P, Yuan Y, Huang Y, Zhang X (2015) A new quantum inspired chaotic artificial bee colony algorithm for optimal power flow problem. *Energy Convers Manag* 100:1–9
- Zhu G, Kwong S (2010) Gbest-guided artificial bee colony algorithm for numerical function optimization. *Appl Math Comput* 217:3166–3173