

Research Article

A Hybrid Firefly and Multi-Strategy Artificial Bee Colony Algorithm

Ivona Brajević^{1,2,*}, Predrag S. Stanimirović², Shuai Li³, Xinwei Cao⁴

¹Faculty of Applied Management, Economics and Finance, University Business Academy, Jevrejska 24, Belgrade, 11000, Serbia

²Faculty of Sciences and Mathematics, University of Niš, Višegradska 33, Niš, 18000, Serbia

³College of Engineering, Swansea University, Fabian Way, Swansea, SA1 8EN, Wales, United Kingdom

⁴School of Management, Shanghai University, 99 Shangda Road, BaoShan District, Shanghai, 201900, China

ARTICLE INFO

Article History

Received 01 Dec 2019

Accepted 09 Jun 2020

Keywords

Firefly algorithm
 Artificial bee colony
 Multi-strategy
 Hybrid algorithm
 Global optimization

ABSTRACT

Many hard optimization problems have been efficiently solved by two notable swarm intelligence algorithms, artificial bee colony (ABC) and firefly algorithm (FA). In this paper, a collaborative hybrid algorithm based on firefly and multi-strategy artificial bee colony, abbreviated as FA-MABC, is proposed for solving single-objective optimization problems. In the proposed algorithm, FA investigates the search space globally to locate favorable regions of convergence. A novel multi-strategy ABC is employed to perform local search. The proposed algorithm incorporates a diversity measure to help in the switch criteria. The FA-MABC is tested on 40 benchmark functions with diverse complexities. Comparative results with the basic FA, ABC and other recent state-of-the-art metaheuristic algorithms demonstrate the competitive performance of the FA-MABC.

© 2020 The Authors. Published by Atlantis Press SARL.

This is an open access article distributed under the CC BY-NC 4.0 license (<http://creativecommons.org/licenses/by-nc/4.0/>).

1. INTRODUCTION

There are numerous engineering and science problems which can be formulated as global optimization problems. These problems are often difficult to solve due to their diverse properties, such as non-linearity, nondifferentiability, multi-modality and nonseparability. During last decades many metaheuristic algorithms have been proposed for solving hard optimization problems [1–5].

Two major characteristics of metaheuristics are exploitation and exploration [6]. Exploitation is the process of focusing search on a local region by using the information of previously visited good solutions. Exploration allows the algorithm to explore entirely new regions of a search space, often by randomization. Too much exploration increases the chance of finding the optimal solution with better accuracy, but usually tends to decrease the convergence speed of the algorithm [7]. On the other hand, high exploitation tends to increase the convergence rate of the algorithm, but the probability of finding global optimum may be low. Since exploitation and exploration are fundamentally conflicting processes, their balanced combination is essential for successful optimization performance.

Most metaheuristics are based on the swarm intelligence [8]. When solving an optimization problem, these algorithms use multiple interacting intelligent agents, inspired by the collective behavior of social insects, such as bees or ants. Firefly algorithm (FA) [9]

and artificial bee colony (ABC) [10] are two widely used swarm intelligence algorithms originally proposed to solve numerical optimization problems. Both algorithms have simple concepts and easy implementation. The main difference between the FA and ABC is how new solutions are created and then used at each iteration. Earlier studies indicate that the FA has good exploration and exploitation abilities, but it may suffer from high computational complexity and it may show slow convergence speed during the search process [11,12]. On the other hand, the solution search equation of ABC algorithm is good for exploration [13], while the selection mechanism guides the search toward regions of the best solutions. However, solution quality does not improve sufficiently fast when the ABC is applied to solve some complex problems [14].

According to the no free lunch theorem [15], no single algorithm is suitable for solving all optimization problems. Consequently, it is important to know which algorithm performs best on which type of optimization problems. Therefore, original variants of metaheuristics are later modified to improve their performances. The main enhancements are reached by the adaptation of parameters, by modifications of solution search equations and creation of ensemble method which combines multiple search strategies [13,16,17]. Besides these improvements, there are also hybrid algorithms which have a prominent role in enhancing the search capability of algorithms.

In a hybrid algorithm, two or more algorithms are cooperatively solving an optimization problem [18]. The aim of hybridization is

*Corresponding author. Email: ivona.brajevic@googlemail.com

to combine the benefits of each algorithm, while at the same time it tries to minimize any considerable disadvantage. As a result, hybrid algorithm can achieve better performance than a single algorithm in terms of either computational speed or accuracy. Since various algorithms have their own characteristics which are suitable for different problems, there are a lot of studies that hybridize different metaheuristics in order to produce a valuable synergy from their combination. Recent studies indicate that hybrid approaches are effective for global optimization [19,20].

Motivated by these research studies, in this paper a novel hybrid method based on FA and multi-strategy artificial bee colony (MABC) is proposed to solve complex numerical optimization problems. The proposed hybrid algorithm, abbreviated as FA-MABC, belongs to the category of collaborative hybrids with multi-stage structure [18]. There are two stages in this case. To take advantage of the good search abilities of the FA, it is used in the first stage with the intention to find promising areas of the search space. In order to reduce computational complexity of the FA, the FA variant in which each solution in each iteration can be updated at most once is employed. Output of the FA is then supplied as the input to a novel multi-strategy ABC, which performs a local search. In the proposed MABC two distinct search strategies coexist throughout the search process and participate in creating candidate solutions. The use of the two search strategies which have different advantages so that they support each other during the search and the selection mechanism, enables the proposed algorithm to efficiently investigate the neighborhood of good solutions. Finally, a diversity measure is used in order to determine when to switch from the FA to the MABC.

To evaluate the performance of FA-MABC, it is tested on 12 well-known benchmark functions and on a set of 28 benchmark instances taken from CEC2013 competition. The obtained results are compared to those of the basic FA, ABC and their recent variants, as well as two other successful metaheuristic approaches.

Rest of this paper is organized as follows: The basic FA and ABC are presented in Section 2. Section 3 describes the new approach FA-ABC in detail. The experimental results and the corresponding analysis are presented in Section 4. Section 5 provides concluding remarks.

2. A BRIEF INTRODUCTION OF FA AND ABC

2.1. Firefly Algorithm

The FA, developed by Yang, is a population-based metaheuristics inspired by communication behavior of flashing fireflies [8]. Its mathematical model is proposed according to the following idealized characteristics of the flashing fireflies [21]: (1) all fireflies are unisex, (2) a firefly's attractiveness is proportional to its light intensity or brightness and (3) a firefly's brightness is determined by the landscape of the fitness function.

In the population of fireflies, each firefly represents a candidate solution in the search space. The attractiveness between fireflies can be defined by monotonically decreasing function [8]:

$$\beta = \beta_{min} + (\beta_0 - \beta_{min}) \cdot e^{-\gamma \cdot r_{ij}^2}, \quad (1)$$

where r_{ij} is the distance between firefly i and firefly j , while β_{min} , β_0 and γ are predetermined algorithm parameters: the minimum value of β , the maximum attractiveness value and absorption coefficient, respectively. Distance r_{ij} between two fireflies i and j at positions x_i and x_j is obtained by

$$r_{ij} = \sqrt{\sum_{k=1}^D (x_{i,k} - x_{j,k})^2}, \quad (2)$$

where D is the problem dimension and $x_{i,k}$ and $x_{j,k}$ are the k th dimension value of solutions x_i and x_j . The parameter β_0 represents attractiveness when two fireflies are found at the same point of search space, while the parameter γ characterizes the variation of the attractiveness. It was reported in the literature that for most problems the parameter β_0 can be set to 1 and the parameter γ can take value from 0.01 to 100 [9].

If the objective function value of x_j is less than x_i , each parameter value $x_{i,k}$ is updated by the following rule described in Ref. [8]:

$$x_{i,k} = x_{i,k} + \beta \cdot (x_{j,k} - x_{i,k}) + \alpha \cdot \left(rand_{i,k} - \frac{1}{2} \right), \quad (3)$$

where $x_{i,k}$ and $x_{j,k}$ are the k th dimension value of solutions x_i and x_j , respectively, and $k = 1, 2, \dots, D$. The second term on the right hand of Eq. (3) is due to the attraction, while the third term is randomization term. In the randomization term, $\alpha \in [0, 1]$ is the randomization parameter and $rand_{i,k}$ is a random number uniformly distributed between 0 and 1.

Later findings indicate the solution quality can be improved by reducing the randomization parameter α with a geometric progression reduction scheme which can be described by [8]

$$\alpha = \alpha_0 \cdot \theta^t, \quad (4)$$

where α_0 is the initial randomization parameter, $\theta \in [0, 1]$ is the randomness reduction constant, $t \in [0, G_{max}]$ is the pseudo time for simulations and G_{max} is the maximum generation number. This step is optional in the FA.

The pseudo code of the basic FA is described in Algorithm 1. The input of Algorithm 1 includes the population size value SP , the maximum number of fitness evaluations $MaxNFEs$, the value of randomization parameter α , the value of parameter β_{min} , the maximum attractiveness value β_0 , the value of absorption coefficient γ and the objective function f . The output of Algorithm 1 is the solution with the smallest objective function value.

2.2. Artificial Bee Colony

ABC algorithm is a metaheuristic technique inspired by the foraging behavior of natural honey bee swarms [10]. In the ABC a colony of artificial bees consists of three groups of bees: employed bees, onlooker bees and scouts. One half of the population of artificial bees are employed bees, while the other half consists of the onlookers and scouts. The number of the employed bees is equal to the number of food sources (possible solutions). Employed bees are all bees that are currently exploiting a food source, meanwhile they share their information about food sources with the onlookers. Onlooker bees select quality food sources from those found by

Algorithm 1: Pseudo-code of the basic FA

```

Create initial population of solutions  $x_i, i = 1, 2, \dots, SP$ ;
Calculate the objective function value of each solution;
 $FES = SP$ ;
while  $FES < MaxNFES$  do
  for  $i = 1$  to  $SP$  do
    for  $j = 1$  to  $SP$  do
      if  $f(x_j) < f(x_i)$  then
        Move  $x_j$  towards  $x_i$  according to Eq. (3);
        Calculate the objective function value of the new  $x_i$ ;
         $FES++$ ;
      end if
    end for
  end for
Rank the fireflies and find the current best;
end while

```

the employed bees and further search the foods around the selected food sources. Employed bees that abandon their unpromising food sources to search for new ones become scout bees.

In the initialization step of the ABC algorithm, the population of solutions is created randomly in the search space. After the initialization step, the employed, onlooker and scout phases are repeated a predefined number of generations. In the end of each generation, the found source so far is memorized.

In the employed phase every solution x_i is updated by

$$v_{i,m} = x_{i,m} + \varphi \cdot (x_{i,m} - x_{i,m}), \quad i = 1, 2, \dots, SP/2, \quad (5)$$

where m is a randomly chosen parameter index, φ is a uniform random number in range $(-1,1)$ and x_i represents the other solution selected randomly from the population. The update process is finalized after greedy selection is applied between x_i and v_i .

In the onlooker phase, the solutions which will be subjected to the update process are selected according to the fitness proportionate selection. The update process in the onlooker phase is the same as in the employed phase. In the scout phase, solutions that do not change over a certain number of trials are again initialized.

2.3. FA and ABC Variants

Nowadays the FA and ABC represent some of the most popular metaheuristic algorithms due to their effective performance and easy implementation.

Several prominent FA variants were proposed for solving continuous optimization problems. Novel chaotic improved firefly algorithm (CFA) was presented in Ref. [22]. Different chaotic maps are used to replace the attraction parameters β and γ of FA in order to improve the reliability of the optimization results. To reduce high computational cost of FA, several FA variants have been proposed [11,12]. For instance, in Ref. [11] the FA with random attraction (RaFA) which uses a random attraction model and a Cauchy mutation operator is developed. Also, in Ref. [12] the FA with neighborhood attraction (NaFA) is proposed. In the NaFA each firefly was attracted by other brighter fireflies selected from a limited neighborhood. The FA variant (FADMF) in which the sex of fireflies is distinguished is presented in Ref. [23]. A hybrid algorithm based on genetic algorithm and the FADMF is presented in Ref. [19]. In Ref. [20] a hybrid FA and particle swarm optimization algorithm

which uses three novel operators is proposed. An improved chaotic FA (ICFA) is recently proposed variant of FA which uses chaotic FA as the parent algorithm and introduces a novel search strategy [24].

Although the basic FA was originally developed for solving continuous optimization problems, the extended versions have been also described for the discrete and combinatorial types of problems. Some of these versions were applied to solve scheduling problems, queueing system and traveling salesman problems [21].

There are numerous ABC variants for solving numerical optimization problems. For instance, in Ref. [14] it was noticed that the ABC algorithm needs more parameters to be mutated in the parent solution in order to improve convergence speed. Hence, the ABC search strategy used a novel control parameter that determines how many parameters should be modified for the production of a neighboring solution. Multi-strategy ensemble ABC (MEABC) algorithm is ABC variant in which a pool of distinct solution search strategies coexists throughout the search process and competes to create candidate solutions [13]. A novel ABC algorithm with local and global information interaction (ABCLGII) is a recent ABC variant which proposes two information interaction mechanisms for employed and onlooker bees [25]. The ABC based on the gravity model (ABCG) is a newly proposed variant of ABC which employs an attractive force model for choosing a better neighbor of a current solution to enhance the exploitation ability of ABC [26]. A novel individual-dependent multi-colony ABC algorithm (IDABC) is a recent ABC variant, in which the whole colony is divided into three sub-colonies and three evolution operators are introduced into the corresponding sub-colonies in order to play different roles [27]. An improved ABC based on the multi-strategy fusion (MFABC) is one of the latest ABC variants proposed in Ref. [28] to enhance the search ability of ABC with small population.

Apart from ABC variants for numerical optimization problems, the extended ABC versions also have been described for the discrete and combinatorial types of problems. Some of these versions were applied to solve capacitated vehicle routing problem, the reliability redundancy allocation problem, different versions of scheduling problem, economic load dispatch problem and knapsack problem. Application areas of ABC algorithm include neural networks, image processing, data mining, industrial, mechanical, electrical, electronics, control, civil and software engineering [29].

3. THE PROPOSED APPROACH: FA-MABC

The randomization term of the FA search equation gives an ability to the algorithm to escape from a local optimum in order to search on a global level. In terms of the attraction mechanism, individuals can subdivide themselves into several subgroups, and each group can seek around a local region [30]. These characteristics point out that the FA has good exploration and exploitation abilities. Although the FA has gotten success in different areas, it has some insufficiencies. In the FA the solutions are still changing as the optima are approaching, which may slow down the convergence speed [9]. In addition, the search process of the FA has high computational complexity since each firefly is attracted by all other brighter fireflies in the swarm.

On the other hand, the ABC search strategy has good exploration ability. Thus ABC is very efficient in solving multimodal and

multidimensional basic functions. However, the convergence rate of the ABC is slow when it is applied to solve hybrid complex problems. To successfully solve these problems, the ABC solution search equation needs to be modified [14].

Since the FA and ABC have their own positive features, their combination could produce a valuable synergy. Therefore, a collaborative hybrid algorithm based on firefly and ABC is developed. The FA explores the search space globally to locate the favorable regions of the search space, whereas the novel version of ABC algorithm performs local search. Inspired by earlier mentioned observations related to ABC, we propose a novel multi-strategy ABC to act as the local searcher. In addition, the diversity measure is used to help in the switch criteria. Therefore the proposed FA-MABC has three main components, the global search optimizer, the local search algorithm and the switch criteria. The details of each component and implementation steps of the FA-MABC are presented as follows.

3.1. The Global Search Optimizer

In the proposed hybrid algorithm, the global searcher is the variant of FA which uses random attraction model. In this FA variant each solution is compared to another randomly selected solution from the set of promising solutions in the population. Assume that all SP individuals in the population are sorted according to their objective function values. Hence, the first firefly x_1 is the best one, and the SP th firefly x_{SP} is the worst one. In the proposed random attraction model, each solution x_i , randomly selects one solution, x_j , from the set $\{x_1, x_2, \dots, x_{i-1}\}$. Then the update of each parameter value $x_{i,k}$ is determined by

$$x_{i,k}^{t+1} = x_{i,k}^t + \beta \cdot \left(x_{j,k}^t - x_{i,k}^t \right) + \alpha S_k \cdot \left(rand_i - \frac{1}{2} \right), \quad (6)$$

where S_k is the length scale for the k th variable, $rand_i$ is a uniform random number in range $(0,1)$, t denotes the generation number, $j \in \{1, 2, \dots, i-1\}$ and $k = 1, 2, \dots, D$. Our simulations confirmed earlier observations that the search process of FA can be superior if the randomization parameter α is related to the scale of each variable [8,22]. The scaling parameters S_k are calculated by

$$s_k = |u_k - l_k|, \quad (7)$$

where l_k and u_k are the lower and upper bound of the parameter $x_{i,k}$.

In the basic FA, the average number of updates of solution in each generation is $\frac{(SP-1)}{2}$. On the other hand, in the proposed FA variant, each solution can be updated at most once in each generation. The number of solution updates is much lower in the proposed FA variant, than in basic FA.

3.2. The Local Search Algorithm

ABC performance mainly depends on its search equation given by Eq. (5). According to the Eq. (5), the new candidate solution is created by moving the old solution to a randomly selected individual, and the search direction is completely random. Hence the Eq. (5) is random enough for exploration and consequently can provide solutions with plenty of diversity and far from the actual solutions.

Diverse optimization problems require different search strategies depending on the characteristics of problems. It was noticed that the ABC needs to mutate more parameters in the neighborhood of the current solution to successfully solve hybrid complex problems, while the use of standard search strategy given by the Eq. (5) can efficiently solve basic functions [14]. Therefore, the number of parameters in the parent solution which are modified in the update process is very important.

Combining search strategies which have different abilities so that they can complement each other during the search process can achieve better optimization results than a single search strategy as in the basic ABC [13]. Motivated by these findings, an ensemble of multiple solution search strategies for ABC (MABC) is developed to perform local search. In the proposed MABC, two search equations coexist throughout the search process and compete to create better new solutions. The first one is basic ABC search strategy given by Eq. (5). The second search strategy employed in the MABC in order to generate a new candidate solution v_i by using the solution x_i , is described by

$$v_{i,k} = x_{i,m} + \varphi_i \cdot (x_{i,m} - x_{l,m}), \quad (8)$$

where m is a randomly chosen parameter index, φ_i is a random number in range $(-1,1)$, x_l represents the other solution selected randomly from the population and $k = 1, 2, \dots, D$.

It is expected that the basic ABC search equation given by Eq. (5) provides good exploration ability but slower convergence speed, while the employed search strategy of the MABC given by Eq. (8) shows the fast convergence rate.

Algorithm 2: Dynamic regulation for search strategy

```

if  $f(v_i) < f(x_i)$  then
     $x_i = v_i$ ;
    {solution  $x_i$  is updated and its assigned search strategy  $S_i$  is kept for further search}
else
    if  $S_i = S_1$  then
         $S_i = S_2$ ;
    else
         $S_i = S_1$ ;
    end if
    {solution  $x_i$  is kept and its assigned search strategy is replaced for further search}
end if

```

In order to determine how to assign these search strategies to solutions from the population, an encoding method is used. This encoding method is inspired by the study given in Ref. [13]. Let us denote the search strategy given by the Eq. (5) as S_1 and the search strategy given by the Eq. (8) as S_2 . At the beginning of the search, each solution x_i is randomly assigned a search strategy, S_i , from the set $\{S_1, S_2\}$. In the course of search process, the value of S_i is changed according to the quality of the new candidate solution v_i . If the candidate v_i has lower objective function value than its parent x_i , it indicates that the current search equation is appropriate for the search. In that case, the current strategy is kept for the further search and the parent solution is replaced with the candidate solution. Otherwise, it means that the current strategy cannot enhance the quality

of solution and it is replaced. Also, in that case the parent solution is kept for next generation.

This encoding method is presented in Algorithm 2. The input of Algorithm 2 includes the parent solution x_i and its assigned search strategy S_i , the candidate solution v_i and the objective function f . The output of Algorithm 2 is the solution x_i and its assigned search strategy S_i which will be used in next generation.

3.3. The Switch Criteria

Differences among individuals of a population are prerequisite for exploration, but too much diversity in each phase of the search, may lead to inefficient search. Population diversity is usually high at the beginning of a search process, and it decreases as the population moves towards the global optimum.

In the implementation of the FA-MABC it is important to know when to switch from the FA to MABC. For this purpose the FA-MABC incorporates the diversity metric as Eq. (9) to measure the population diversity [25].

$$Diversity = \frac{1}{SP} \sum_{i=1}^{SP} \sqrt{\frac{1}{D} \sum_{k=1}^D (x_{i,k} - x'_{i,k})^2}, \quad (9)$$

where SP is number of solutions in the population, D is the dimension of problem and $x'_{i,k}$ is the central position of the whole population. In order to determine when to switch the search to MABC, the population diversity during each generation is measured.

Variable *counter* is set to 0 before the search. Then, in each iteration the population diversity is calculated. If the population diversity in the current iteration is higher than the population diversity from the previous one, we increase *counter* by one. Otherwise, the value of *counter* does not change. When the value of variable *counter* becomes higher than the predefined threshold value (T), it means that too many oscillations in the population diversity are produced. Therefore, it indicates that the solution quality does not improve sufficiently quickly. In that moment the search is switched to the MABC, to help the proposed approach avoid stagnation.

3.4. The Pseudo-Code of the FA-MABC

The FA-MABC uses the boundary constraint handling mechanism which ensures that if variables of a created solution by each of these three search equations go outside of boundaries, a diverse set of values is created. This boundary constraint handling method is given by [16]

$$x_{i,k} = \begin{cases} 2 \cdot l_k - x_{i,k} & , \text{if } x_{i,k} < l_k \\ 2 \cdot u_k - x_{i,k} & , \text{if } x_{i,k} > u_k \\ x_{i,k} & , \text{otherwise} \end{cases} \quad (10)$$

where $x_{i,k}$ is the variable k of the new solution x_i , l_k and u_k are the k th lower bound and upper bound of the parameter $x_{i,k}$.

Algorithm 3: Pseudo-code of the FA-MABC

```

Create initial population of solutions  $x_i, i = 1, 2, \dots, SP$  and calculate objective function value of each solution;
Randomly initialize  $S_i$  for each solution;
Calculate the population diversity  $d_1$  according to Eq. (9);
 $t = 0$ ;
 $counter = 0$ ;
while  $t < G_{max}$  do
  if ( $counter < T$ ) then
    for  $i = 1$  to  $SP$  do
      Randomly select solution  $x_j$ , where  $j \in \{1, 2, \dots, i - 1\}$ ;
      if  $f(x_j) < f(x_i)$  then
        Move  $x_i$  towards  $x_j$  according to Eq. (6);
        Apply control of the boundary conditions on the created solution by Eq. (10) and evaluate it;
      end if
    end for
    Rank the solutions;
    Calculate the population diversity  $d_2$  according to Eq. (9);
    if ( $d_1 < d_2$ ) then
       $counter = counter + 1$ ;
    end if
     $d_1 = d_2$ ;
  else
    for  $i = 1$  to  $SP$  do
      Produce new solution  $v_i$  for the parent solution  $x_i$  according to the search strategy  $S_i$ ;
      Apply control of the boundary conditions on the created solution  $v_i$  by Eq. (10) and evaluate it;
      Update  $x_i$  and  $S_i$  according to Algorithm 2;
    end for
  end if
  Find the current best solution;
   $t = t + 1$ ;
end while

```

The pseudo-code of the FA-MABC is described in Algorithm 3. The input of Algorithm 3 includes the population size value SP , the maximum generation number G_{max} , the value of randomization parameter α , the value of parameter β_{min} , the maximum attractiveness value β_0 , the value of absorption coefficient γ , the value of parameter T and the objective function f . The output of Algorithm 3 is the solution with the smallest objective function value.

To solve a particular problem f , assume that $O(f)$ is the computational time complexity of evaluating its function value. It can be noticed that in each generation of the FA-MABC, each individual from the population can be updated at most once. Since the maximum number of generations is set to G_{max} , the computational time complexity of the FA-MABC is $O(G_{max} \cdot SP \cdot f)$.

4. EXPERIMENTAL STUDY

To investigate the performance of the FA-MABC, 12 well-known scalable benchmark functions with 30 and 100 variables and a set of 28 benchmarks taken from CEC2013 competition are used. The brief descriptions of the 12 well-known benchmark functions are listed in Table 1. When the objective function value of the best solution obtained by an algorithm in a run is less than the corresponding acceptable value, the run is regarded as a successful

run. Presented benchmark functions can be divided into two categories according to their features: unimodal functions ($f_1 - f_6$) and multimodal ($f_7 - f_{12}$). Additionally, as the number of dimensions increases, the search space of a problem grows exponentially and its characteristics may change. More details about the description of these benchmark problems can be found in Ref. [25]. The benchmark instances from the CEC2013 cover a wide range of problem properties, such as multi-modality, nonseparation, non-symmetrization and rotation. These 28 benchmarks include 5 unimodal functions ($F1 - F5$), 15 basic multimodal functions ($F6 - F20$) and 8 composition functions ($F21 - F28$). More detailed descriptions of CEC2013 benchmarks can be found in Ref. [31].

In order to evaluate the performance of the FA-MABC, it is compared to the basic FA, ABC and recent variants of the FA and ABC. In these experiments two types of comparisons were examined. First type of comparison is a direct comparison, where the basic FA, ABC and FA-MABC were implemented and their corresponding performances are investigated. In the second type of comparison, the results of other prominent variants of FA and ABC were taken from the specialized literature and compared with those achieved by FA-MABC. The last experiment is used to discuss how the proposed algorithmic components affect the performance of FA-MABC.

4.1. Comparison with the Basic FA and ABC

In this subsection, the FA-MABC is compared with the basic FA and ABC on 12 scalable benchmark functions listed in Table 1 with 30 and 100 variables. Each algorithm was implemented in Java programming language on a PC with Intel(R) Core(TM) i5-4460 3.2GHz processor with 16GB of RAM and Windows OS.

In the proposed FA-MABC, values of the parameter G_{max} are 7500 and 25000 for benchmark functions with 30 and 100 variables respectively. Hence, the maximal number of function evaluations, $MaxNFEs$, is $5000 D$. Beside the control parameter G_{max} , the FA-MABC has few other control parameters that greatly influence its performance. The preliminary testing of the FA-MABC was done in order to get good combinations of parameter values. A value of SP equal to 20 was found to be a good choice for all tests carried out in this paper. Our tests confirmed earlier findings that the randomization parameter α in range $(0, 1)$, the initial attractiveness β_0

Table 1 Benchmark functions used in the experiments, where D is the problem dimension.

Function	Name	Search Range	Min.	Accept
f_1	Sphera	$[-100, 100]^D$	0	1e-8
f_2	Schwefel 2.22	$[-10, 10]^D$	0	1e-8
f_3	Schwefel 2.21	$[-100, 100]^D$	0	1e+0
f_4	Step	$[-100, 100]^D$	0	1e-8
f_5	Quartic	$[-1.28, 1.28]^D$	0	1e-1
f_6	Rosenbrock	$[-5, 10]^D$	0	1e-1
f_7	Rastrigin	$[-5.12, 5.12]^D$	0	1e-8
f_8	Griewank	$[-600, 600]^D$	0	1e-8
f_9	Schwefel2.26	$[-500, 500]^D$	0	1e-8
f_{10}	Ackley	$[-32, 32]^D$	0	1e-8
f_{11}	Penalized1	$[-50, 50]^D$	0	1e-8
f_{12}	Penalized2	$[-50, 50]^D$	0	1e-8

of 1 and the parameter γ from 0.01 to 100 can be used for most of problems [9]. Hence, the FA-MABC employs the following settings: α is 0.8, β_0 is 1.0, β_{min} is 0.2 and γ is 1. Also, it was empirically determined that the value $0.01 \cdot G_{max}$ of the parameter T is suitable for the FA-MABC.

In the FA and ABC, the $MaxNFEs$ is set to $5000 D$ in order to provide the fair comparison between these algorithms. In Ref. [32] it was found that the size of population from 10 to 25 is sufficient for most applications of the FA. Therefore, in the FA the SP value is set to 20. In the case of the FA, the values of specific control parameters are α_0 is 0.2, β_0 is 1.0, β_{min} is 0.2, γ is 1 and θ is $(10^{-4}/0.9)^{1/G_{max}}$. In the ABC the value of parameter $limit$ is set to $(SP \cdot D)/2$. The FA and ABC use the same specific parameter settings as those used in the original papers, i.e., in Ref. [8] and in Ref. [10] respectively. All compared algorithms are conducted by 30 independent runs for each benchmark problem.

The mean and corresponding standard deviation values of 30 independent runs are used to determine the quality or accuracy of the solutions achieved by the FA, ABC and FA-MABC. The convergence speed of each approach is compared by the metric AVEN. This metric records the average number of function evaluations needed to reach the acceptable value, which is used to evaluate the convergence speed. The robustness or reliability of each algorithm is compared by measuring the success rate (SR%). This rate denotes the ratio of successful runs in the 30 independent runs. The convergence speed is faster if the value of AVEN is smaller, while the robustness of an algorithm is better if the value of SR is greater. Wilcoxon's rank sum test at a 0.05 significance level was applied between the compared metaheuristic algorithm and the FA-MABC. The result of the test is presented as +/-/-, which means that the corresponding algorithm is significantly better than, statistically similar to, and significantly worse than the FA-MABC.

The statistical results of the comparisons on the benchmarks with 30 and 100 dimensions are summarized in Tables 2 and 3. These results include the obtained mean best function values and corresponding standard deviations, AVEN and SR results. The best results are indicated in bold.

Results from Tables 2 and 3 clearly show that the FA-MABC is significantly better than the basic ABC and FA on all test functions in terms of solution accuracy, robustness and convergence rate. Exceptions are problems f_4 and f_7 at each tested dimension, where the ABC and FA-MABC reached the same optimization results. Additionally, the superiority of the FA-MABC is not affected by the growth of the dimensions of search space.

4.2. Comparison with the Other Variants of FA and ABC

In this subsection, the FA-MABC is compared with six recent FA and ABC variants, i.e., RaFA [11], NaFA [12], ICFA [24], ABCLGII [25], ABCG [26] and MFABC [28] on 12 test functions with 30 variables. These 12 approaches were previously tested to solve same benchmarks and had a very good results.

Results reported by other FA and ABC variants were taken from the specialized literature and compared with those achieved by the FA-MABC. The results obtained by the RaFA and NaFA are taken

Table 2 | Comparison results of the FA, ABC and FA-MABC on 12 benchmark functions with 30D. The best results are indicated in bold.

Fun.	Metric	FA	ABC	FA-MABC
f_1	Mean(std.)	6.28e-05(1.02e-05)-	4.95e-16(7.47e-17)-	3.97e-286(0.00e+00)
	AVEN(SR)	NaN(0)	32233(100)	6133(100)
f_2	Mean(std.)	3.76e-03(3.77e-04)-	1.30e-15(1.35e-16)-	2.36e-152(8.26e-152)
	AVEN(SR)	NaN(0)	48287(100)	9415(100)
f_3	Mean(std.)	3.79e-03(5.81e-04)-	7.56e-01(4.61e-01)-	2.53e-97(1.36e-96)
	AVEN(SR)	35983(100)	123820(80)	652(100)
f_4	Mean(std.)	1.67e-01(5.82e-01)-	0.00e+00(0.00e+00)=	0.00e+00(0.00e+00)
	AVEN(SR)	NaN(0)	9582(100)	1009(100)
f_5	Mean(std.)	7.08e-02(1.74e-02)-	4.96e-02(9.05e-03)-	5.50e-04(2.55e-04)
	AVEN(SR)	16055(97)	45506(100)	223(100)
f_6	Mean(std.)	3.74e+01(2.84e+01)-	1.93e-01(4.71e-01)-	8.80e-30(2.52e-29)
	AVEN(SR)	NaN(0)	69257(77)	1984(100)
f_7	Mean(std.)	4.15e+01(9.22e+00)-	0.00e+00(0.00e+00)=	0.00e+00(0.00e+00)
	AVEN(SR)	NaN(0)	50031(100)	6760(100)
f_8	Mean(std.)	4.67e-03(5.05e-03)-	9.03e-04(3.73e-03)-	0.00e+00(0.00e+00)
	AVEN(SR)	NaN(0)	43382(93)	6291(100)
f_9	Mean(std.)	5.23e+03(7.00e+02)-	1.57e-12(2.14e-12)-	0.00e+00(0.00e+00)
	AVEN(SR)	NaN(0)	71002(100)	6918(100)
f_{10}	Mean(std.)	1.93e-03(1.16e-04)-	3.86e-14(4.76e-15)-	3.64e-15(1.07E-15)
	AVEN(SR)	NaN(0)	55905(100)	9192(100)
f_{11}	Mean(std.)	3.46e-03(1.86e-02)-	5.11e-16(7.01e-17)-	1.57e-32(5.47e-48)
	AVEN(SR)	NaN(0)	28153(100)	5568(100)
f_{12}	Mean(std.)	3.70e-03(6.60e-03)-	4.48e-16(9.65e-17)-	1.35e-31(6.57e-47)
	AVEN(SR)	NaN(0)	39590(100)	9029(100)
+/-/-		0/0/12	0/2/10	-

FA, firefly algorithm; ABC, artificial bee colony; FA-MABC, firefly and multi-strategy artificial bee colony; SR, success rate.

Table 3 | Comparison results of the FA, ABC and FA-MABC on 12 benchmark functions with 100D. The best results are indicated in bold.

Fun.	Metric	FA	ABC	FA-MABC
f_1	Mean(std.)	7.27e-04(7.56e-5)-	2.18e-15(1.71e-16)-	0.00e+00(0.00e+00)
	AVEN(SR)	NaN(0)	112177(100)	12911(100)
f_2	Mean(std.)	2.87e-02(5.24e-03)-	5.03e-15(2.29e-16)-	7.25e-226(0.00e+00)
	AVEN(SR)	NaN(0)	171409(100)	21435(100)
f_3	Mean(std.)	5.22e-02(1.69e-02)-	2.31e+01(2.93e+00)-	5.71e-07(1.62e-06)
	AVEN(SR)	262971(100)	NaN(0)	707(100)
f_4	Mean(std.)	1.40e+00(16.70e+00)-	0.00e+00(0.00e+00)=	0.00e+00(0.00e+00)
	AVEN(SR)	NaN(0)	35593(100)	995(100)
f_5	Mean(std.)	3.70e-01(5.24e-02)-	1.58e-01(2.06e-02)-	5.71e-04(1.85e-04)
	AVEN(SR)	NaN(0)	-(0)	454(100)
f_6	Mean(std.)	1.30e+02(9.42e+01)-	6.18e-01(1.20e-01)-	8.46e-30(2.48e-29)
	AVEN(SR)	NaN(0)	280551(47)	2133(100)
f_7	Mean(std.)	2.30e+02(2.86e+01)-	0.00e+00(0.00e+00)=	0.00e+00(0.00e+00)
	AVEN(SR)	NaN(0)	187870(100)	12718(100)
f_8	Mean(std.)	1.27e-03(2.19e-03)-	5.45e-15(2.70e-14)-	0.00e+00(0.00e+00)
	AVEN(SR)	NaN(0)	124554(100)	16061(100)
f_9	Mean(std.)	2.87e-02(5.24e-03)-	1.18e-10(2.96e-11)-	1.09e-10(0.00e+00)
	AVEN(SR)	NaN(0)	331965(100)	11844(100)
f_{10}	Mean(std.)	3.48e-03(1.58e-04)-	1.52e-13(1.10e-14)-	8.55e-15(3.26e-15)
	AVEN(SR)	NaN(0)	191609(100)	19556(100)
f_{11}	Mean(std.)	2.07e-03(7.76e-03)-	2.183e-15(6.04e-16)-	4.71e-33(1.37e-48)
	AVEN(SR)	NaN(0)	143832(100)	10058(100)
f_{12}	Mean(std.)	2.52e-02(3.55e-02)-	2.13e-15(1.56e-16)-	1.35e-31(6.57e-47)
	AVEN(SR)	NaN(0)	96054(100)	23399(100)
+/-/-		0/0/12	0/2/10	-

FA, firefly algorithm; ABC, artificial bee colony; FA-MABC, firefly and multi-strategy artificial bee colony; SR, success rate.

from Ref. [12], while the results reached by the ICFA is taken from Ref. [24]. The results of ABCLGII are taken from Ref. [25], the results of the ABCG are taken from Ref. [26], while the results of the MFABC are taken from Ref. [28]. In the RaFA and NaFA, the value of $MaxNFEs$ was set to $5E+05$, while in the ICFA the value of $MaxNFEs$ was set to $3.8e+05$. In the ABCLGII, ABCG, MFABC and FA-MABC the value of $MaxNFEs$ was set to $1.5e+5$. Therefore the fair comparison between all these algorithms is ensured. All the other parameter settings of the FA-MABC are kept the same as mentioned in the Section 4.1. The specific parameter settings of each metaheuristic algorithm used in comparison with the FA-MABC can be found in their original papers.

To check for statistically significant differences between the FA-MABC and each compared algorithm for all benchmark functions, the multiproblem Wilcoxon signed-rank test at a 0.05 significance level is conducted [33]. The results of this test are represented as “+/-/-”, “ R^+ / R^- ” and p value. Sign “+” indicates that the FA-MABC is significantly better than the corresponding algorithm, sign “-” indicates that the FA-MABC is significantly worse than the corresponding algorithm and sign “=” that there is no significant difference between the two algorithms. The sum of ranks for the test problems in which the FA-MABC performs better than the corresponding algorithm is denoted by R^+ , while R^- denotes the sum of ranks for the opposite. Larger ranks point to larger performance discrepancy. The null hypothesis assumes that there is no significant difference between the mean results of the two samples, while the alternative hypothesis assumes that there is significance in the mean results of the two samples. The null hypothesis is rejected if the p -value is less than or equal to the significance level 0.05.

The mean best results and the statistical results of applying Wilcoxon test between the FA-MABC and each FA and ABC variant are listed in Table 4 for 12 problems with 30 variables. Best mean results are indicated in bold.

As shown in Table 4, the FA-MABC outperforms or performs similarly to its competitors in most cases. Precisely, the FA-MABC is better than the RaFA, NaFA, ICFA, ABCLGII, ABCG and MFABC on 9, 10, 9, 8, 6 and 7 cases respectively. In contrast, the FA-MABC is outperformed by the RaFA, NaFA, ICFA, ABCLGII, ABCG and

MFABC on 1, 0, 1, 1, 1, 1 benchmarks respectively. With respect to the best results reached by all compared metaheuristics, the FA-MABC performs the best, and achieves the best results on 10 instances for these benchmarks. The second best approach is the ABCG, which performs the best on 5 test problems. From the results of Wilcoxon test presented in Tables 4 and 5, it can be seen that the FA-MABC obtains higher R^+ values than R^- in comparison with each compared approach. The reason is that the FA-MABC outperforms each compared algorithm in most of the cases. From the obtained p values it can be concluded the FA-MABC significantly outperforms the RaFA, NaFA, ABCLGII and MFABC, while it is comparable with the ICFA and ABCG.

4.3. Comparison on CEC2013 Test Functions

In order to further examine the effectiveness of the FA-MABC, it is compared with two FA variants (i.e., FADMF [23], FADCG [19]) and two ABC variants (i.e., IDABC [27] and ABCG [26]) on benchmark functions from CEC2013 with 30 variables. Since differential evolution (DE) and particle swarm optimization (PSO) are among the most widely used metaheuristic algorithms, the results of their recent variants, the multi-population ensemble DE (MPEDE) [34] and heterogeneous comprehensive learning PSO (HCLPSO) [35], are also included in comparison with the same of the FA-MABC. The MPEDE algorithm uses an adaptive ensemble of three mutation strategies [34]. The HCLPSO divides the entire population into two heterogeneous subpopulation groups and each subpopulation group is assigned to carry out the exploration and exploitation search separately [35].

The MPEDE, HCLPSO, FADMF, FADCG, ABCG and IDABC were recently tested to solve test problems from CEC2013. The results of FADMF and FADCG are taken from Ref. [19], while the results of the MPEDE, HCLPSO, ABCG and IDABC are taken from Ref. [27].

For fair comparison, value of $MaxNFEs$ was set to $3e+05$ for all the algorithms, according to the guidelines from Ref. [31]. The

Table 4 | Comparison results of the RaFA, NaFA, ICFA, ABCLGII, ABCG, MFABC and FA-MABC on 12 benchmark functions with 30D D . The best mean results are indicated in bold.

Fun.	RaFA Mean	NaFA Mean	ICFA Mean	ABCLGII Mean	ABCG Mean	MFABC Mean	FA-MABC Mean
f_1	5.36e-184	4.43e-29	1.24e-39	3.48e-89	3.67e-109	2.46e-123	3.97e-286
f_2	8.76e-05	2.98e-15	1.54e-20	2.13e-46	6.93e-062	7.17e-64	2.36e-152
f_3	2.43e+00	3.43e-15	1.67e-20	1.14e-04	2.03e-026	1.81e-02	2.53e-97
f_4	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0	0	0.00e+00
f_5	5.47e-02	2.91e-02	1.90e-04	1.44e-02	2.70e-03	1.41e-02	5.50e-04
f_6	2.92e+01	2.39e+01	2.53e-05	4.83e+00	5.39e+01	2.47e-01	8.80e-30
f_7	2.69e+01	2.09e+01	5.92e-17	0.00e+00	0	0	0.00e+00
f_8	0.00e+00	0.00e+00	3.70e-18	1.28e-03	0	0	0.00e+00
f_9	5.03e+02	6.86e+03	3.82e-04	3.57e-12	0	1.82e-13	0.00e+00
f_{10}	3.61e-14	3.02e-14	2.60e-14	8.95e-15	4.44e-15	1.68e-14	3.64e-15
f_{11}	4.50e-05	1.36e-31	1.57e-32	1.57e-32	1.57e-032	1.57e-032	1.57e-32
f_{12}	8.25e-32	2.13e-30	1.42e-31	1.50e-33	1.35e-032	1.35e-032	1.35e-31
R^+ / R^-	53/2	55/0	37/8	42/3	25/3	33/3	-
p value	0.009	0.005	0.086	0.021	0.063	0.036	-

RaFA, FA with random attraction; NaFA, FA with neighborhood attraction; ICFA, improved chaotic FA; ABCLGII, ABC algorithm with local and global information interaction; ABCG, ABC based on the gravity model; MFABC, ABC based on the multi-strategy fusion; FA-MABC, firefly and multi-strategy artificial bee colony.

Table 5 | Comparison results of the MPEDE, HCLPSO, FADMF, FADCG, IDABC, ABCG and FA-MABC on CEC2013 benchmark functions with 30D D . NA means not available. The best mean of the function error values are indicated in bold.

Fun.	MPEDE Mean err.	HCLPSO Mean err.	FADMF Mean err.	FADCG Mean err.	IDABC Mean err.	ABCG Mean err.	FA-MABC Mean err.
F1	0.000e+0	3.695e-13	5.61e-4	1.06e-3	0.000e+0	6.537e-13	2.274e-13
F2	1.271e+4	1.036e+6	3.96e+6	6.59e+6	1.069e+4	1.092e+7	9.965e-01
F3	4.370e+4	4.748e+7	9.14e+6	1.19e+7	2.686e+3	6.250e+8	8.771e+03
F4	1.383e-4	1.969e+3	1.38e+5	2.57e+5	9.680e-6	7.686e+4	5.402+00
F5	1.137e-13	4.832e-13	1.92e-2	1.78e-1	1.137e-13	5.400e-13	3.411e-13
F6	6.679e-13	1.796e+1	2.65e+1	2.73e+1	2.274e-13	4.160e+1	3.542e-08
F7	1.528e+1	1.704e+1	4.98e+0	6.35e+0	1.703e+1	9.944e+1	1.056e-01
F8	2.095e+1	2.096e+1	2.14e+1	2.14e+1	2.093e+1	2.093e+1	1.124e-01
F9	1.990e+1	1.716e+1	1.01e+1	8.74e+0	1.822e+1	2.913e+1	7.448e-01
F10	3.911e-2	1.935e-1	1.95e-1	5.50e-1	4.678e-2	2.635e-1	9.865e-03
F11	0.000e+0	1.812e-10	3.88e+1	2.42e+1	7.105e-15	1.492e-13	5.684e-14
F12	2.823e+1	5.261e+1	3.77e+1	2.96e+1	1.487e+1	1.307e+2	9.865e-03
F13	6.903e+1	1.282e+2	9.43e+1	7.85e+2	2.925e+1	2.065e+2	2.842e-13
F14	5.079e-1	1.392e+1	2.20e+2	1.51e+3	2.030e-1	2.504e-1	1.748e-12
F15	3.462e+3	3.598e+3	2.23e+3	2.52e+3	3.393e+3	3.816e+3	8.463e-11
F16	2.396e+0	1.618e+0	1.29e-1	3.30e-1	5.884e-3	1.653e+0	2.461e-01
F17	3.044e+1	3.504e+1	8.71e+1	7.59e+1	3.044e+1	3.044e+1	1.430e-03
F18	6.095e+1	8.244e+1	9.16e+1	8.61e+1	5.234e+1	2.101e+2	0.000e+00
F19	1.299e+0	1.434e+0	4.01e+0	3.47e+0	1.127e+0	6.914e-1	3.411e-13
F20	1.054e+1	1.003e+1	NA	NA	1.047e+1	1.437e+1	1.137e-13
F21	3.054e+2	2.607e+2	3.39e+2	3.12e+2	3.593e+2	2.375e+2	1.364e-12
F22	7.861e+1	1.359e+2	2.61e+3	1.71e+3	8.839e+1	2.327e+1	1.023e-12
F23	3.515e+3	3.787e+3	3.31e+3	2.99e+3	3.561e+3	5.095e+3	9.095e-13
F24	2.426e+2	2.247e+2	2.22e+2	2.23e+2	2.402e+2	2.807e+2	9.095e-13
F25	2.530e+2	2.584e+2	2.32e+2	2.25e+2	2.616e+2	2.959e+2	6.150e-13
F26	2.000e+2	2.001e+2	2.85e+2	3.00e+2	2.000e+2	2.009e+2	2.501e-12
F27	7.491e+2	5.371e+2	5.17e+2	4.85e+2	6.783e+2	7.925e+2	2.274e-13
F28	4.356e+2	3.000e+2	3.09e+2	2.97e+2	2.750e+2	3.000e+2	2.274e-12
R^+ / R^-	387/19	406/0	375/3	378/0	353/53	406/0	-
p value	0.000	0.000	0.000	0.000	0.001	0.000	-

MPEDE, multi-population ensemble DE; HCLPSO, heterogeneous comprehensive learning PSO; IDABC, individual dependent multi-colony ABC algorithm; ABCG, ABC based on the gravity model; FA-MABC, firefly and multi-strategy artificial bee colony.

FA-MABC algorithm is run 51 times. The other parameter settings of the FA-MABC are kept the same as mentioned in the Section 4.1. The mean of the function error values obtained by the MPEDE, HCLPSO, FADMF, FADCG, IDABC, ABCG and FA-MABC are listed in Table 5. The best results are indicated in bold. In the Table 5 the statistical results of applying Wilcoxon test between the FA-MABC and each competing metaheuristic algorithm are also presented.

According to the comparison results in Table 5, the FA-MABC outperforms competing metaheuristic approaches in the majority of CEC2013 test functions. Concretely, the FA-MABC performs better than MPEDE, HCLPSO, FADMF, FADCG, IDABC and ABCG on 23, 28, 26, 27, 21 and 28 benchmark problems respectively. On the other hand, the FA-MABC is outperformed by the MPEDE, HCLPSO, FADMF, FADCG, IDABC and ABCG on 5, 0, 1, 0, 7 and 0 problems respectively. Overall, based on the obtained p values it is clear that the FA-MABC significantly outperforms the MPEDE, HCLPSO, FADMF, FADCG, IDABC and ABCG on the CEC2013 functions.

5. DISCUSSION

In this section some experiments are performed with the aim to investigate impact of proposed algorithmic components on performance of the FA-MABC. Four different versions of the

FA-MABC have been tested and compared against the proposed one on CEC2013 benchmark functions with 30 D . These versions are described as follows:

1. *Version 1*: To explore the effectiveness of using the FA algorithm with random attraction model as the global optimizer alone, a variant of the FA-MABC which excludes this algorithmic component is implemented. This version is denoted as the MABC.
2. *Version 2*: To investigate the impact of using the proposed multi-strategy ABC as the local optimizer alone, a variant of the FA-MABC which excludes this algorithmic component is implemented. This version is denoted as the FA-ra.
3. *Version 3*: To explore the impact of using the modified ABC search strategy given by Eq. (8) alone, a variant of the FA-MABC which excludes this search equation is implemented. This version is denoted as the MABC-FA1.
4. *Version 4*: To examine the impact of using the original ABC search strategy given by Eq. (5) alone, a variant of the FA-MABC which excludes this search strategy is implemented. This version is denoted as the MABC-FA2.

Each FA-MABC variant is performed over 51 independent runs for each benchmark function. Value of $MaxNFEs$ was set to $3e+05$ for

Table 6 | Comparison results of the MABC, FA-ra, FA-MABC1, FA-MABC2 and FA-MABC on CEC2013 benchmark functions with 30D D . The best mean of the function error values are indicated in bold.

Fun.	MABC Mean error	FA-ra Mean error	FA-MABC1 Mean error	FA-MABC2 Mean error	FA-MABC Mean error
F1	2.274e-13	9.616e-01	2.274e-13	4.972e-01	2.274e-13
F2	1.250e+00	2.11e+00	9.999e-01	9.999e-01	9.965e-01
F3	8.880e+03	3.500e+05	3.027e+04	3.038e+04	8.771e+03
F4	8.392e+00	3.246e+01	9.029e+01	5.284e+00	5.402e+00
F5	3.411e-13	4.344e-01	3.411e-13	2.565e-01	3.411e-13
F6	4.250e-08	1.494e-01	8.286e-04	8.112e-02	3.542e-08
F7	1.092e-01	4.935e-01	3.572e-01	1.340e-01	1.056e-01
F8	2.095e+01	2.129e+00	2.035e+00	3.607e-01	1.124e-01
F9	1.235e+00	1.980e+00	4.399e-01	8.752e-01	7.448e-01
F10	9.865e-03	4.779e-02	2.929e-02	2.920e-02	9.865e-03
F11	5.684e-14	1.918e+00	5.684e-14	8.371e-01	5.684e-14
F12	9.865e-03	4.770e-02	2.841e-02	2.875e-02	9.865e-03
F13	3.829e+01	1.839e+00	9.071e-02	8.519e-01	2.842e-13
F14	1.28e+00	5.072e+01	6.677e-02	2.135e+01	1.748e-12
F15	2.632e+03	1.241e+01	9.876e+00	9.939e+00	8.463e-11
F16	1.066e+01	9.303e-02	9.136e-01	2.515e+00	2.461e-01
F17	3.916e-02	9.665e+01	6.798e-03	4.392e+01	1.430e-03
F18	6.459e-08	7.580e+00	0.000e+00	5.169e+00	0.000e+00
F19	4.854e-01	3.197e-02	3.308e-05	7.789e-03	3.411e-13
F20	4.153e+00	1.396e+00	1.171e+01	6.779e+00	1.137e-13
F21	1.137e-12	1.947e+02	1.137e-12	1.685e+02	1.364e-12
F22	9.95e+01	1.538e+02	5.850e+01	1.348e+02	1.023e-12
F23	4.787e+02	1.394e+02	9.397e+01	1.285e+02	9.095e-13
F24	9.095e-13	1.028e+02	3.383e+01	8.334e+01	9.095e-13
F25	3.270e+02	1.030e+02	3.677e+01	4.064e+02	6.150e-13
F26	2.273e-12	2.013e+02	4.088e+01	1.757e+02	2.501e-12
F27	2.728e-12	2.009e+02	3.254e+01	1.766e+02	2.274e-13
F28	2.274e-12	2.017e+02	5.790e+01	1.760e+02	2.274e-12
R^+ / R^-	228/3	401/5	289/11	399/7	-
p value	0.000	0.000	0.000	0.000	-

FA-MABC, firefly and multi-strategy artificial bee colony; ABC, artificial bee colony.

each algorithm. Other parameter settings for each tested algorithm are the same as mentioned in the Section 4.1. The error mean results and the statistical results of applying Wilcoxon test between the FA-MABC and each described version of the FA-MABC are presented in Table 6. Best mean error results are indicated in bold. The convergence graphs obtained by different versions of FA-MABC on the four selected benchmarks are given in Figure 1 to provide an inside into evolutionary trajectories of mean error values.

From Table 6 it can be seen that the FA-MABC outperforms each tested version on the majority of benchmarks. Concretely, the FA-MABC performs better than MABC, FA-ra, MABC-FA1 and MABC-FA2 on 19, 27, 22 and 27 benchmark problems respectively. On the other hand, the FA-MABC is outperformed by the MABC, FA-ra, MABC-FA1 and MABC-FA2 on 2, 1, 2 and 1 problems respectively. According to the obtained p values it is clear that the FA-MABC significantly outperforms each tested variant on the CEC2013 functions. As shown in Figure 1, FA-MABC clearly outperforms its competitors on the selected problems at the end of evolution.

These observations imply that each of the four algorithmic components significantly contributes to the good performance of the FA-MABC. Use of the FA with random attraction model helps the FA-MABC to discover the promising regions of the search space, while using the proposed multi-strategy ABC assists in reaching more accurate optimization results by performing intensive local

search. Experimental results also confirm that the use of each suggested ABC search operators enables the FA-MABC to significantly improve the quality of the obtained solutions.

6. CONCLUSION

In this paper, a novel multi-stage collaborative hybrid algorithm based on the FA and multi-strategy ABC (FA-MABC) is proposed to solve numerical optimization problems. The FA with random attraction model performs global search whereas a novel multi-strategy ABC is developed to act as local optimizer.

The proposed FA-MABC efficiently utilizes the benefits of the FA and MABC through the two stages. In the first stage, the FA extensively explores the search space to find the promising areas. It is noticed that the FA can provide strong exploration ability, but it may show slow convergence rate due to its oscillatory behavior as the search process proceeded toward the optimum. To avoid stagnation of the search, the diversity measure is employed to determine when the output of the FA will be supplied as the input to the MABC. The MABC employs two search strategies with different advantages so that they complement each other during the search. The use of the efficient integration of different search operators and the selection mechanism can conduct a more refined search in the promising regions to improve both, the accuracy of the solutions and convergence speed.

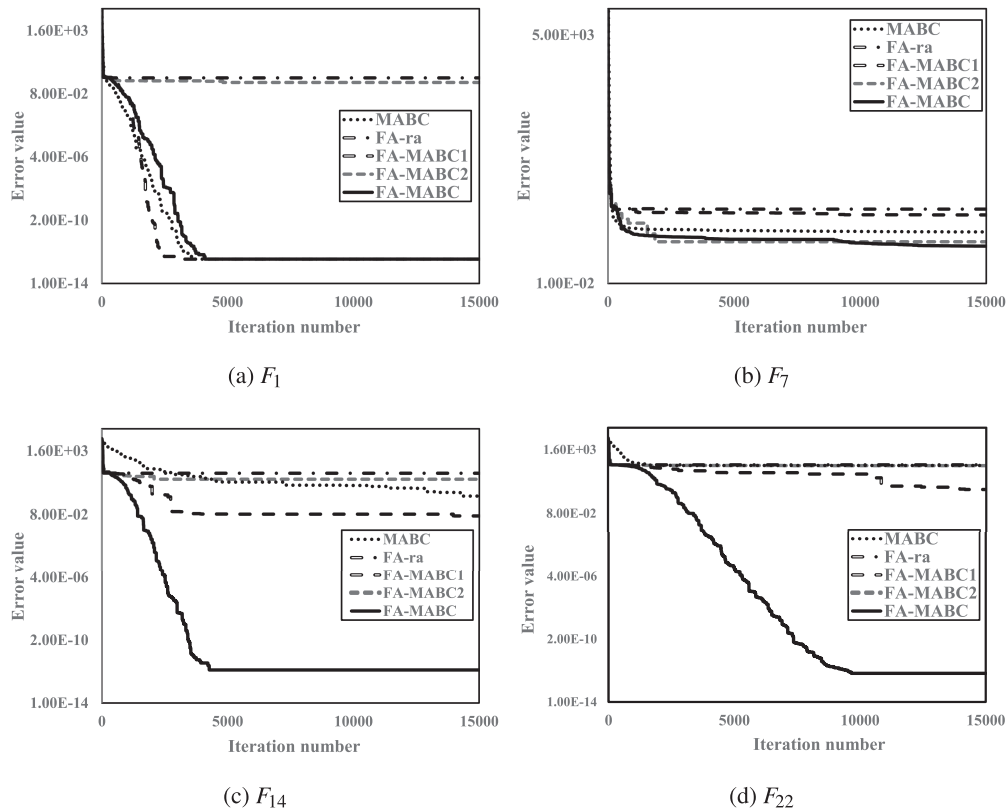


Figure 1 | Convergence graphs of the mean function error values for the selected problems.

In order to verify the performance of the FA-MABC, it is tested on a large set of numerical benchmark functions. The computational results showed that the FA-MABC significantly outperformed the basic FA and its five prominent variants, the basic ABC and its four outstanding variants and two other popular metaheuristic algorithms on majority of the benchmark functions. Also, experimental results validate that each algorithmic component, the FA with random attraction model, the multi-strategy ABC and each used ABC search strategy, significantly contributes to superior performance of the FA-MABC. In the future work, the performance of the FA-MABC applied to multi-objective optimization problems will be investigated. The FA-MABC can also be used to solve some practical optimization problems.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

Ivona Brajević proposed the main idea of the paper, implemented the algorithm, performed the experiments and wrote the major draft of the paper. Predrag S. Stanimirović contributed to the writing of the manuscript and in analyzing of the results. Shuai Li and Xinwei Cao reviewed the paper and provided helpful comments to further improve the quality of the paper.

FUNDING

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

ACKNOWLEDGMENTS

Predrag Stanimirović gratefully acknowledges support from the Ministry of Education, Science and Technological Development, Republic of Serbia, Grant No. 174013.

REFERENCES

- [1] X. Cui, Z. Chen, F. Yin, Differential evolution and local search based Monarch butterfly optimization algorithm with applications, *Int. J. Comp. Intell. Syst.* 12 (2018), 149–163.
- [2] S. Lu, X. Wang, Discrete firefly algorithm for clustered multi-temperature joint distribution with fuzzy travel times, *Int. J. Comp. Intell. Syst.* 11 (2018), 195–205.
- [3] I. Stojanović, I. Brajević, P.S. Stanimirović, L.A. Kazakovtsev, Z. Zdravev, Application of Heuristic and Metaheuristic algorithms in solving constrained weber problem with feasible region bounded by arcs, *Math. Prob. Eng.* 13 (2017), 1–13.
- [4] Q. Wu, Z. Ma, G. Xu, S. Li, D. Chen, A novel neural network classifier using beetle antennae search algorithm for pattern classification, *IEEE Access.* 7 (2019), 64686–64696.

- [5] Y. Sun, Y. Gao, An efficient modified particle swarm optimization algorithm for solving mixed-integer nonlinear programming problems, *Int. J. Comp. Intell. Syst.* 12 (2019), 530–543.
- [6] M. Črepinšek, S.-H. Liu, M. Mernik, Exploration and exploitation in evolutionary algorithms: a survey, *ACM Comput. Surv.* 45 (2013), 1–33.
- [7] X.-S. Yang, X. He, Firefly algorithm: recent advances and applications, *Int. J. Swarm Intell.* 1 (2013), 36–50.
- [8] X.-S. Yang, *Nature-Inspired Metaheuristic Algorithms*, Luniver Press, BA11 6TT United Kingdom, 2010. ISBN 978-1-905986-28-6
- [9] X.-S. Yang, Firefly algorithms for multimodal optimizations, in: O. Watanabe, T. Zeugmann (Eds.), *Stochastic Algorithms: Foundations and Applications*, Springer Lecture Notes in Computer Science, vol. 5792, Springer, Berlin, Heidelberg, Germany, 2009, pp. 169–178.
- [10] D. Karaboga, An Idea Based on Honey Bee Swarm for Numerical Optimization, Technical Report-tr06, Erciyes University, Engineering Faculty, Computer Engineering Department, Turkey. 2005 <https://pdfs.semanticscholar.org/015d/f4d97ed1f541752842c49d12e429a785460b.pdf>.
- [11] H. Wang, W.J. Wang, H. Sun, S. Rahnamayan, Firefly algorithm with random attraction, *Int. J. Bio Inspired Comput.* 8 (2016), 33–41.
- [12] H. Wang, W. Wang, X. Zhou, H. Sun, J. Zhao, X. Yu, Z. Cui, Firefly algorithm with neighborhood attraction, *Inf. Sci.* 382–383 (2017), 374–387.
- [13] H. Wang, Z. Wu, S. Rahnamayan, H. Sun, Y. Liu, J.-S. Pan, Multi-strategy ensemble artificial bee colony algorithm, *Inf. Sci.* 279 (2014), 587–603.
- [14] B. Akay, D. Karaboga, A modified artificial bee colony algorithm for real parameter optimization, *Inf. Sci.* 192 (2012), 120–142.
- [15] D.H. Wolpert, W.G. Macready, No free lunch theorems for optimization, *IEEE Trans. Evol. Comput.* 1 (1997), 67–82.
- [16] I. Brajević, J. Ignjatović, An upgraded firefly algorithm with feasibility-based rules for constrained engineering optimization problems, *J. Intell. Manuf.* 30 (2019), 2545–2574.
- [17] I. Brajevic, Crossover-based artificial bee colony algorithm for constrained optimization problems, *Neural Comput. Appl.* 26 (2015), 1587–1601.
- [18] T.O. Ting, X.-S. Yang, S. Cheng, K. Huang, Hybrid metaheuristic algorithms: past, present, and future, in: X.S. Yang (Eds.), *Recent Advances in Swarm Intelligence and Evolutionary Computation*, Studies in Computational Intelligence, vol. 585, Springer, Cham, Switzerland, 2015, pp. 71–83.
- [19] M. Takeuchi, H. Matsushia, Y. Uwate, Y. Nishio, Hybrid method of genetic algorithm and firefly algorithm distinguishing between males and females, in *Proceedings of International Symposium on Nonlinear Theory and its Applications (NOLTA'16)*, Yagawara, Japan, 2016, pp. 542–545.
- [20] X. Xia, L. Gui, G. He, C. Xie, B. Wei, Y. Xing, R. Wu, Y. Tang, A hybrid optimizer based on firefly algorithm and particle swarm optimization algorithm, *J. Comput. Sci.* 26 (2018), 488–500.
- [21] I. Fister, I. Fister Jr, X.S. Yang, J. Brest, A comprehensive review of firefly algorithms, *Swarm Evol. Comput.* 13 (2011), 36–46.
- [22] A.H. Gandomi, X.-S. Yang, S. Talatahari, A.H. Alavi, Firefly algorithm with chaos, *Commun. Nonlin. Sci. Numer. Simul.* 18 (2013), 89–98.
- [23] M. Takeuchi, H. Matsushia, Y. Uwate, Y. Nishio, Investigation of firefly algorithm distinguishing between males and females for minimum optimization problems, in *Proceedings of RISP International Workshop on Nonlinear Circuits, Communications and Signal Processing (NCSP'16)*, Honolulu, Hawaii, USA. 2016, pp. 534–537, <http://nlab.ee.tokushima-u.ac.jp/nishio/PublicData/CONF/C569.pdf>.
- [24] I. Brajević, P. Stanimirović, An improved chaotic firefly algorithm for global numerical optimization, *Int. J. Comp. Intell. Syst.* 12 (2018), 131–148.
- [25] Q. Lin, M. Zhu, G. Li, W. Wang, L. Cui, J. Chen, J. Lu, A novel artificial bee colony algorithm with local and global information interaction, *Appl. Soft Comput.* 62 (2018), 702–735.
- [26] W. Xiang, X. Meng, Y. Li, R. He, M. An, An improved artificial bee colony algorithm based on the gravity model, *Inf. Sci.* 429 (2018), 49–71.
- [27] J. Zhou, X. Yao, F.T.S. Chan, Y. Lin, H. Jin, L. Gao, X. Wang, An individual dependent multi-colony artificial bee colony algorithm, *Inf. Sci.* 485 (2019), 114–140.
- [28] X. Song, M. Zhao, S. Xing, A multi-strategy fusion artificial bee colony algorithm with small population, *Expert Syst. Appl.* 142 (2020), 112921.
- [29] D. Karaboga, B. Gorkemli, C. Ozturk, N. Karaboga, A comprehensive survey: Artificial Bee Colony (ABC) algorithm and applications, *Artif. Intell. Rev.* 42 (2014), 21–57.
- [30] X.-S. Yang, X.-S. He, Why the firefly algorithm works?, in X.-S. Yang (Ed.), *Nature-Inspired Algorithms and Applied Optimization*, Springer International Publishing, Cham, Switzerland, 2018, pp. 245–259.
- [31] J. Liang, P. Suganthan, A. Hernandez-Daz, Problem Definitions And Evaluation Criteria for the CEC 2013 Special Session on Real-Parameter Optimization, Technical Report 201212, Zhengzhou University, Zhengzhou, China, and Nanyang Technological University, Singapore, 2013. https://www.al-roomi.org/multimedia/CEC_Database/CEC2013/RealParameterOptimization/CEC2013_RealParameterOptimization_TechnicalReport.pdf
- [32] A.H. Gandomi, X.-S. Yang, A.H. Alavi, Mixed variable structural optimization using firefly algorithm, *Comput. Struct.* 89 (2011), 2325–2336.
- [33] J. Derrac, S. García, D. Molina, F. Herrera, A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, *Swarm Evol. Comput.* 1 (2011), 3–18.
- [34] G. Wu, R. Mallipeddi, P.N. Suganthan, R. Wang, H. Chen, Differential evolution with multi-population based ensemble of mutation strategies, *Inf. Sci.* 329 (2016), 329–345.
- [35] N. Lynn, P.N. Suganthan, Heterogeneous comprehensive learning particle swarm optimization with enhanced exploration and exploitation, *Swarm Evol. Comput.* 24 (2015), 11–24.