A new multi-objective artificial bee colony algorithm based on reference point and opposition

Songyi Xiao
School of Information Engineering,
Nanchang Institute of Technology,
Nanchang 330099, China
Email: speaknow@126.com

Wenjun Wang
School of Business Administration,
Nanchang Institute of Technology,
Nanchang 330099, China
Email: wangwenjun881@126.com

Hui Wang* and Zhikai Huang
School of Information Engineering,
Nanchang Institute of Technology,
Nanchang 330099, China
Email: huiwang@whu.edu.cn
Email: huangzhik2001@163.com
*Corresponding authors

Abstract: A new multi-objective artificial bee colony (ABC) algorithm based on reference point and opposition (called ROMOABC) is proposed in this paper. Firstly, the original framework of ABC is modified to improve the efficiency of population renewal and accelerate the convergence rate. On the basis of this framework, two new strategies are proposed. In the scout bee search, opposition-based learning and elite solutions are used to reduce the waste of computing resources. Distribution of solutions is improved by using reference points’ associated external archive. Experiments are conducted on 16 multi-objective benchmark functions including ZDT, DTLZ and WFG multi-objective benchmark functions. The comparison of ROMOABC with five other multi-objective algorithms shows that it has competitive convergence and diversity.

Keywords: artificial bee colony; ABC; multi-objective optimisation; external archive; opposition; elite learning.


Biographical notes: Songyi Xiao received his BSc in Network Engineering from the Nanchang Institute of Engineering, China in 2018, where he is currently pursuing his Master’s degree with the School of Information Engineering. He has published three international journal/conference articles. His current research interests include swarm intelligence and global optimisation.

Wenjun Wang is currently an Associate Professor with the School of Business Administration, Nanchang Institute of Technology, Nanchang, China. She has published more than 20 international journal/conference articles. Her current research interests include swarm intelligence, evolutionary computation, global optimisation, and decision optimisation.

Hui Wang received his BSc and MSc in Computer Science from the China University of Geosciences, Wuhan, China in 2005 and 2008, respectively, and PhD in Computer Software and Theory from Wuhan University, Wuhan in 2011. He is currently a Professor with the School of Information Engineering, Nanchang Institute of Technology, China. His current research interests include evolutionary optimisation, swarm intelligence, and large-scale optimisation. He has published more than 50 research articles in the abovementioned fields.

Zhikai Huang is a Professor at the School of Information Engineering, Nanchang Institute of Technology, China. His research interest focuses on pattern recognition and artificial intelligence.
1 Introduction

Optimisation problems are widespread in engineering or scientific fields (Cai et al., 2020a; Cui et al., 2020c; Wang et al., 2018, 2020b). In general, optimisation problems can be divided into single-objective optimisation and multi-objective optimisation according to the number of objective functions. Single-objective optimisation only needs to find the global optimal value in the object space (Wang et al., 2017). However, global optimal solutions in multi-objective optimisation problem (MOP) appear in form of a solution set. Therefore, expansion of object space, convergence degree and distribution of the solution set lead to difficulty of solving MOP. Though traditional mathematical methods are not ideal to solve MOPs, evolutionary algorithm (EA) shows excellent global and local searching ability in solving single-objective optimisation problems. The main reason is EA can obtain approximate global optimal solution set with the evolution of population. So far, there have been many EAs for multi-objective optimisation (Cai et al., 2020b; Coello et al., 2004; Cui et al., 2019; Wang et al., 2020a).

There are three kinds of existing multi-objective optimisation algorithms. The first one is based on Pareto relationship and some representative methods includes NSGA-II (Deb et al., 2002) and SPEA2 (Zitzler et al., 2001). Pareto relationship is used to choose superior solutions. Then, density of solutions is calculated. By means of dominant relationship and density estimation, the quality of solutions in population is further evaluated. Then, superior solutions are reserved for the next generation. Crowding distance (Lei et al., 2014), k nearest neighbour (Yang et al., 2010), ε-dominance (Xu et al., 2012), and grading method are common solution methods of density estimation. The second one is based on evaluation index, such as IBEA (Zitzler and Kinzli, 2004) and HyPE (Bader and Zitzler, 2011). The MOP is indirectly optimised by evaluation index of the Pareto approximate set. Evaluation index of the Pareto approximate set transforms the MOP into a single-objective optimisation problem. The third one is built on decomposition technology, and one of famous methods is MOEA/D (Zhang and Li, 2007), which is a new framework about multi-objective evolutionary algorithm (MOEA). Similar to traditional decomposition-based algorithms, MOEA/D decomposes MOP into series of subproblems. However, traditional method needs to solve each subproblem separately, and it results in large amount of computation. MOEA/D solves all of these subproblems simultaneously, which improves the computational efficiency.

Artificial bee colony (ABC) algorithm, proposed in 2005, is one of the recently introduced population-based search methods (Karaboga, 2005). It has strong global search ability when solving single objective optimisation problems (Wang et al., 2020c, 2020d). Therefore, some scholars combined multi-objective optimisation technology with ABC to propose multi-objective ABC algorithm (Luo et al., 2017). However, compared with many classical multi-objective algorithms, MOABC is not efficient. In order to solve the MOP effectively, the population renewal of traditional MOABC methods should be improved. Therefore, a new multi-objective ABC algorithm is proposed in this paper, which based on reference point and opposition (called ROMOABC). By analysing the existing population updating mode of MOABC, the basic framework of MOABC is modified. Then, two new methods are proposed to enhance the performance. In the scout bee search, opposition-based learning and elite solutions is adopted to improve operational efficiency. External archive is updated by using reference point to maintain the distribution of non-dominated solution sets. To evaluate the performance of ROMOABC algorithm, we compared it with NSGA-II, MOEA/D, MOPSO, SPEA2, and MOABC algorithms in series of test functions. Result shows that our approach ROMOABC has good performance in terms of the IGD indicators on most problems. The effectiveness of the improved framework and strategies are analysed through experiments.

The rest of this article is organised as follows. Section 2 introduces MOP and the original ABC algorithm. Recent work on ABC is presented in Section 3. Our ROMOABC algorithm is described in Section 4. In Section 5, ROMOABC is compared with some famous algorithms and each strategy is validated. Finally, the conclusion is given in Section 6.

2 Background

2.1 Multi-objective optimisation problem

A general MOP can be defined as follows:

\[ \min F_m(x) = \{f_1(x), f_2(x), \ldots, f_m(x)\} \]

\[ \text{s.t. } h_u(x) \leq 0, u = 1, 2, \ldots, p \]

\[ g_v(x) = 0, v = 1, 2, \ldots, q \]

\[ x \in \Omega \]  

where \( m \) is the number of objective functions, \( f(X) \) is the \( i \)th objective function, \( h(x) \) and \( g(x) \) are inequality constraints and equality constraints respectively, and \( x = \{x_1, x_2, \ldots, x_n\} \) is the feasible field of decision variables, \( \Omega \subseteq \mathbb{R}^n \) is decision space, \( n \) is the number of variables. These objectives are often conflicting, and they are required to be optimised simultaneously. Because of relationship of constraints in decision variables among multiple objects, an optimal solution set is usually achieved.

2.2 Standard ABC algorithm

In ABC (Karaboga, 2005), the main search task is accomplished by employed bee part and onlooker bee part. Employed bee part is responsible for searching in a larger area, while onlooker bee part conducts further search in a smaller area. Each food source in ABC denotes a solution. Therefore, the number of food sources in the population is equal to the number of employed bees and onlooker bees. The search process of ABC is mainly composed of four
parts: initialisation part, employed bee part, onlooker bee part and scout bee part.

As a branch of random methods, a swarm intelligence optimisation algorithm generates initial population randomly in the search space. Therefore, the initialisation of ABC produces a set of initial solutions in following equation.

\[ x_{i,j} = x_{\text{min},j} + \text{rand}_{i,j} \cdot (x_{\text{max},j} - x_{\text{min},j}) \] (2)

where \( i \) stands for the index of solution, \( j \) represents the dimension, \( x_{i,j} \) is the \( j \)-th dimension of the \( i \)-th solution \( X_i \), \( x_{\text{min},j} \) is the lower search bound, \( x_{\text{max},j} \) is the upper search bound, and the range of \( \text{rand}_{i,j} \) is \([0, 1]\).

The main task of the employed bee part conducts the neighbourhood search in a large area. New solutions at this part are generated as follows.

\[ v_{i,j} = x_{i,j} + \phi_{i,j} \cdot (x_{i,j} - x_{\text{rand},j}) \] (3)

where \( X_i \neq X_{\text{rand}} \), the dimension index \( j \) randomly selects one from \([1, 2, ..., D]\), \( D \) is the maximum value of the dimension, and \( \phi_{i,j} \) is a weight factor in \([0, 1]\).

The onlooker bee part aims to further search some good solutions in a small area. This is done after the search of the employed bee part. Excellent solutions are selected by means of probabilistic selection:

\[ p_i = \frac{\text{fit}(X_i)}{\sum_{i=1}^{N} \text{fit}(X_i)} \] (4)

where \( p_i \) is the selection probability of solution \( X_i \) of the population calculated by fitness value \( \text{fit}(X_i) \) and it is defined by:

\[ \text{fit}_i = \begin{cases} \frac{1}{1 + f_i} & \text{if } f_i > 0 \\ 1 + \text{abs}(f_i) & \text{otherwise} \end{cases} \] (5)

where \( f_i \) is the objective function value of \( i \)-th solution. Onlooker bees select an excellent solution by equation (4) and uses equation (3) to produce offspring. When some solutions could not be renewed during a long search cycle, scout bees try to re-initialise them by equation (1). This may help some poor solutions jump to better positions.

3 Related work

ABC has achieved success in single objective optimisation problems. To effectively solve MOPs, various modified ABCs were proposed. In this section, recent progresses on multi-objective ABC algorithms are presented.

Based on fuzzy Pareto, Amarjeet and Chhabra (2018) proposed a new multi-objective ABC to solve the software module clustering problem. Xu et al. (2020) tried to build a multi-objective optimisation model of logistics task-resource allocation in the shared logistics network. Then, an improved multi-objective ABC based on adaptive neighbourhood was proposed to solve this model. To optimise the two-stage network programming model, Ma et al. (2019) designed a special multi-objective ABC on the basis of enhanced learning and orthogonal Latin square method. Gong et al. (2018) constructed a hybrid multi-objective discrete artificial bee colony (called HDABC) to solve the blocking batch flow shop scheduling problem.

Ning et al. (2018) designed a multi-objective ABC with external archives, in which external archives were used to preserve non-dominated solutions. In addition, different food source generation methods were used in different bee stages. There are various applications on wireless sensor networks (Cui et al., 2020a, 2020b), and Jena (2014) used a multi-objective ABC for the node placement of WSN. To approximate the real Pareto frontier (PF) as much as possible, Wang and Li (2015) employed a new onlooker bee selection method to lead the population close to the true PF. Moreover, an adaptive search model was designed by considering both convergence speed and diversity.

4 Proposed approach

4.1 Modified framework for multi-objective ABC

ABC is usually used to settle single objective optimisation problems. Facing MOPs, the original ABC should be extended. In some recent studies, several multi-objectives ABC (MOABC) were proposed by using Pareto dominance, decomposition, and indicator method.

Population renewal and selection are important operations for multi-objective ABC. The greedy selection was replaced by the domination relationship between solutions (Akbari et al., 2012). When an offspring dominates its parent solution, the parent solution is replaced by its offspring. When parent is non-dominated with its offspring solution each other, some excellent offspring may be abandoned and the convergence rate is slowed. The solution selection of onlooker bees is determined by non-dominated solutions. Luo et al. (2017) proposed a multi-objective ABC in which solution index was used as fitness to judge solution renewal.

For the update of the trial value, most MOABC algorithms only considered that the trial value is set to 0 when the solution is dominated. But they ignored the update of the trial value under the non-domination relationship between two solutions. A new multi-objective ABC framework is proposed to solve this problem.

As mentioned before, when the current offspring and its parent solutions are not dominated each other, some excellent solutions cannot be selected into the next generation. To tackle this issue, a replication selection method based on non-domination hierarchical and crowded distance composite sorting is proposed in this paper. For the
update of the trial value, it is increased by 1 when the offspring does not dominate its parent solution. At the last evolutionary stage, there are many non-dominated solutions. Solution in population will quickly re-initialise because the trial value reached the maximum value (limit). It slows down the convergence and wastes a lot of computing resources. In our approach, the update rule for the trail value is modified. If the offspring dominates its parent solution, the trial value is set to 0. If the offspring dominated by its parent, the trial value is increased by 1. When parent and its offspring solution are non-dominated each other, and Pareto level of parent solution is 1, the trial value is set to 0; otherwise it is increased by 1.

When ABC is used to solve the single object problem, excellent solutions are selected by the fitness value. In non-dominated stratification, a lower level of solutions in the population, dominates more solutions. It is an excellent way to assess the quality of solutions in the early evolution. However, with increase of evolutionary iterations, the quantity of non-domination layers decreases, which also means that the quantity of non-dominated solutions increases. In this situation, excellent solutions cannot be selected according to the previous method. Therefore, a new probabilistic selection method on the basis of crowding distance in non-domination sorting and non-domination layers (Chen et al., 2019) is designed in this paper.

At early stage of evolution, there are few non-dominated solutions. Solutions with small levels are selected as excellent solutions. At later stage of evolution, there are many non-dominated solutions. It is impossible to judge the quality of solution according to their dominance level. It is better to introduce the size of the crowding distance. Then, a weighted strategy based on the size of the crowding distance is used to construct a new selection probability as follows.

\[
p_i = \frac{e^{level_i}}{\sum_{j=1}^{SN} e^{level_j}} + \frac{e^{dis_i}}{\sum_{j=1}^{SN} e^{dis_j}}
\]

(6)

where \(level_i\) is non-domination ranking of the \(i\)th solution. As seen, the probability is only determined by the Pareto level. For the same Pareto level, the same selection probability is obtained. \(dis_i\) is crowding distance of the \(i\)th solution. Solutions with large crowding distances are preferentially selected. A larger crowding distance means a higher selection probability.

At early stage of the evolution, there are few non-dominated solutions. Then, solutions with low Pareto level are selected. At later stage of evolution, there are many non-dominated solutions. It is impossible to judge the quality of solution according to their dominance level. Then, the size of the crowding distance is used to construct the selection probability.

### 4.2 Modified search strategy

The convergence speed of ABC has greatly influenced by search equation. This paper uses different search equations for two different search stages. In employed bees, the global search is emphasised. The neighbourhood search focuses on the global search, but its search mode cannot accelerate the convergence speed for multi-objective problems. Therefore, the simulated binary crossover is introduced in employed bees. Assume that \(X_1 = \{x_1^1, ..., x_1^k\} \) and \(X_2 = \{x_2^1, ..., x_2^k\}\) \((k\) represent dimension) are two parents. Then, two offspring solutions \(V_1 = \{v_1^1, ..., v_1^k\}\) and \(V_2 = \{v_2^1, ..., v_2^k\}\) are computed as follows (Deb, et al., 2002).

\[
\begin{align*}
    v_1^j &= 0.5 \times \left[ (1 + \beta) \cdot x_1^j + (1 - \beta) \cdot x_2^j \right] \\
    v_2^j &= 0.5 \times \left[ (1 - \beta) \cdot x_1^j + (1 + \beta) \cdot x_2^j \right]
\end{align*}
\]

(7)

where \(\beta\) follows the polynomial probability distribution and is calculated as follows.

\[
\beta = \begin{cases} 
(2 \times u_j)^{\frac{1}{\eta}} & \text{if } u_j \leq 0.5 \\
\left(\frac{1}{2 - 2 \times u_j}\right)^{\frac{1}{\eta}} & \text{otherwise}
\end{cases}
\]

(8)

where \(\eta\) is cross distribution index and \(u_j\) is random number in \([0, 1]\).

Then polynomial mutation is used in \(i\)th solution.

\[
x_{ij} = x_{ij} + \Delta_{ij}
\]

(9)

where \(\Delta_{ij}\) is calculated by:

\[
\Delta_{ij} = \begin{cases} 
\left(2u_j\right)^{\frac{1}{\lambda}} - 1 & u_j < 0.5 \\
1 - \left(2(1 - u_j)\right)^{\frac{1}{\lambda}} & \text{otherwise}
\end{cases}
\]

(10)

where \(u_j\) is random in \([0, 1]\), and \(\lambda\) is mutation distribution index.

For the onlooker bees, the local search is enhanced based on the idea of GABC, and the search equation is used as follows (Zhu and Kwong, 2010).

\[
x_i = x_i + \phi_i (x_j - x_i) + \phi_j(\text{Gbest}_j - x_j)
\]

(11)

where \(\phi_i\) is random in \([0, 1]\), and \(\phi_j\) is \([0, 1.5]\) between the random numbers, including Gbest is solution in the external archive.

### 4.3 Elite opposition-based learning

The scout bee phase aims to help stagnant solution jump out. In ABC, the scout bee only re-initialises the solutions in the stagnant state. Although the resetting operation can effectively stop the solutions from stagnating, it may reduce the search efficiency. The previous search on the stagnant solutions wastes many commuting resources. In this paper, an elite opposition-based learning mechanism is used to avoid resetting and help find new non-dominated solutions with a high probability (Mahdavi et al., 2018).

\[
\bar{X}_{i,j} = r \ast (\text{global}_1 + \text{global}_2) - X_{i,j}
\]

(12)
where \( \text{global}_1 \) and \( \text{global}_2 \) are the solutions with the maximum and minimum crowding distances in the external archive, respectively.

### 4.4 External archive maintenance strategy

The preservation of the external archive is directly associated with the quality of final PF. Therefore, the maintenance of the external archive is very important. There are many ways to maintain the external archive. These strategies directly or indirectly evaluate the density (crowding degree) or dominant strength of each solution in the archive. Usually, old solution with poor fitness is replaced by new solution with good fitness. This can make Pareto front have a better distribution performance. In MOABC (Akbari et al., 2012) and MOPSO (Coello et al., 2004), they used the grid archive method for external archive to filter solutions in the constructed grid according to the density information of solution. When filtering external documents with grids, the size of the grid needs to be recalculated for each update and the computational burden is increased. The crowding distance method used in NSABC selected solutions by judging the size of the crowding distance in the external archive (Kishor et al., 2016). When two solutions are close and their neighbours are farther apart from each other, the crowding distance between the two solutions is equal. So, it is possible for both solutions to be eliminated or to be retained at the same time. The elimination of one of these two solutions makes the population more evenly distributed. The maintenance of the external archive has similarities to the environmental selection in population evolution. Both of them need to balance the distribution and convergence.

To effectively maintain the external archive, the reference point associated operation used in NSGA-III is employed (Deb and Jain, 2014). On the basis of NSGA-II, NSGA-III introduced a reference point associated operation to preserve those solutions who are not dominant but close to the reference point. It aims to improve the distribution of the population. In other words, the associated operation represents the convergence of the solution by vertical distance from solution to reference vector, the distribution of the solution by projection distance of solution on reference vector. And then the solutions are screened by these two measures. Therefore, it is feasible to maintain the external archive by using the reference point.

At the beginning, \( N \) uniform weight points are generated by systematic approach (Das and Dennis, 1998), where \( N \) is the predefined archive size. Then, the ideal point of the problem \( z_{\text{min}} = (z_{\text{min}}^1, z_{\text{min}}^2, ..., z_{\text{min}}^M) \) is consist of minimum value \( (z_{\text{min}}^i) \) for each objective function \( i = 1, 2, ..., M \). Setting ideal point as new zero vector, the translational position of each objective value calculated, which subtracting objective \( f_i \) by \( z_{\text{min}}^i \). We denote this translated objective as temp \( f_k(x) \).

Thereafter, extreme points on each objective axis are calculated. These \( M \) extreme points constitute hyper-plane of \( M-1 \) dimension. At this time, the intercept of each objective direction can be calculated. Then true normalised objective value is calculated by the intercept and temporary normalised objective value.

\[
f_{i}^{\rho}(x) = \frac{f_{i}(x) - z_{i}^{\rho}}{a_{i} - z_{i}^{\rho}}
\]

where \( a_i \) is intercept of \( i^{\rho} \) object, \( z_{i}^{\rho} \) is minimum value of \( i^{\rho} \) object.

A point that is closest to the objective axis is called extreme point. The way to tell extreme point is as below.

\[
ASF(x, w) = \max_{i=1}^{M} \left( \frac{f_{i}(x) - z_{i}^{\rho}}{\omega_i} \right)
\]

where \( \omega_i \) is a weight vector, \( z_{i}^{\rho} \) is minimum value of \( i^{\rho} \) object. \( \omega \) is weight vector which is set artificially to find out extreme point of each objective axis direction. For example, \((1, 0.2, 0.1), (0.8, 2, 0.2), (0.4, 0.6, 1)\) are three solutions of three dimension in three object problem. In order to find out extreme point of \( i^{\rho} \) object direction, the direction vector \( \omega_i \) is set to 1, and the weight of other directions is set to \( 10^{-6} \) infinitely close to 0, then the weight vector \( \omega_i = (1, 10^{-6}, 10^{-6}) \). According to equation (16), three solutions are calculated as \((1, 2 \times 10^6, 1 \times 10^6), (0.8, 2 \times 10^6, 2 \times 10^5), (0.4, 6 \times 10^5, 1 \times 10^6)\). Among of them, the biggest value is \( 2 \times 10^6, 2 \times 10^5 \) and \( 2 \times 10^6 \), respectively. And the smallest value of three biggest values, \( 2 \times 10^5 \) \((1, 0.2, 0.1)\), which represent extreme point of \( i^{\rho} \) object axis.

When the solutions in the external archive are normalised, they are associated with reference points. Zero vector and reference point to form a reference line in the hyper-plane. The vertical distance between each solution in the external archive and each reference line is calculated. The reference point closest to a population member is considered to be associated with that solutions in the external archive.

After completed association of all solution in archive, ideal situation is one solution associate one reference point. Two problems arise when the number of non-dominated solutions is more than predefined number of reference point. The first one is two or more solutions associate a reference point. The second one is no one solution associate a reference point. NSGA-III solves it by a new technology. To be specific, the number of solutions associated with the reference point \( k \) is denoted as \( q_k \). Then the minimum reference point of \( q_k \) is selected. If there are multiple reference points with a minimum count, a random one would be selected. Of all the solutions associated with the reference point, the solution closest to reference point is added to the current archive. However, a reference point is not considered if it is not associated with a solution. Repeat the above steps until the current external archive size is \( N \).
The pseudocode for external archive updates and opposite of elite learning is given in Algorithm 1. In Algorithm 2, the main framework of ROMOABC is described.

**Algorithm 1**  
External archive updating

Create SN reference points;

Begin

archive = population + archive;

fast non-domination sorting (archive);

new archive = Pareto solutions;

If the current archive size > predefined archive size:

Normalised archive;

Associate archive with reference points to filter out excess solutions;

End If

Output new-archive;

End

**Algorithm 2**  
Proposed approach (ROMOABC)

Begin

initialise all trial = 0;

while FEs < MaxFEs do

for i in population of employed bee do

Generate $V_i$ based on equation (7) and equation (9);

Calculate the function value of $V_i$ and $FEs++$;

if $f(V_i)$ dominates $f(X_i)$ then

trial, is set to zero;

else if $f(V_i)$ and $f(X_i)$ are not dominated each other and levels($X_i$)==1

trial, is set to zero;

else

trial++;  
end if

end for

Save solutions according to fast non-domination sort and crowding distance;

Execute Algorithm 1;

Calculate selection probability according to equation (6);

$i = 0$ and $t = 0$

while (r < SN)

if prob, > r

Generate $V_i$ according to equation (11);

Calculate the function value and fitness value of $V_i$;

FEs increase by 1;

if $f(V_i)$ dominates $f(X_i)$ then

trial, is set to zero;

else if $f(V_i)$ and $f(X_i)$ are not dominated each other and levels($X_i$)==1

trial, is set to zero;

else

trial++;  
end if

end if

end while

Save solutions according to fast non-domination sort and crowding distance;

Execute Algorithm 1;

if trial, > limit then

Replace $X_i$ with a new solution generated by equation (12);

end if

Output the external archive;

End

### 5 Experimental study

#### 5.1 Parameter settings

Several classical multi-objective algorithms are used to verify the effectiveness of our ROMOABC algorithm. In the experiments, there are three kinds of benchmark problems including ZDT (Zitzler et al., 2000), DTLZ (Deb et al., 2002), and WFG (Huband et al., 2006). A simple description of each problem is shown in Table 1, where $D$ is dimension, $M$ is objectives number, and range is search boundaries of decision variables. In Table 1, $[0]^*$ means all dimension are 0, $[2]^*(D - 1) + 1$ represent dimension value change with dimension number, for example, the dimension value is $[2]^*(D - 1) + 1 = [3, 5, 7, ..., 2*(D - 1) + 1]$. $[0, -5]^*(D - 1)$ represent the first dimension value is 0, other dimension value is $[-5]^*(D - 1) = [-5, -10, -15, ..., -5*(D - 1)]$.

<table>
<thead>
<tr>
<th>Function</th>
<th>$M$</th>
<th>$D$</th>
<th>Range([low],[up])</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZDT1</td>
<td>2</td>
<td>30</td>
<td>$[0]^*D$, $[1]^*D$</td>
</tr>
<tr>
<td>ZDT2</td>
<td>2</td>
<td>30</td>
<td>$[0]^*D$, $[1]^*D$</td>
</tr>
<tr>
<td>ZDT3</td>
<td>3</td>
<td>10</td>
<td>$[0]^*D$, $[1]^*D$</td>
</tr>
<tr>
<td>ZDT4</td>
<td>2</td>
<td>10</td>
<td>$[0, -5]^<em>(D - 1)$, $[1, 5]^</em>(D - 1)$</td>
</tr>
<tr>
<td>ZDT6</td>
<td>2</td>
<td>10</td>
<td>$[0]^*D$, $[1]^*D$</td>
</tr>
<tr>
<td>DTLZ1</td>
<td>3</td>
<td>7</td>
<td>$[0]^*D$, $[1]^*D$</td>
</tr>
<tr>
<td>DTLZ2</td>
<td>3</td>
<td>12</td>
<td>$[0]^*D$, $[1]^*D$</td>
</tr>
<tr>
<td>DTLZ3</td>
<td>3</td>
<td>12</td>
<td>$[0]^*D$, $[1]^*D$</td>
</tr>
<tr>
<td>DTLZ4</td>
<td>3</td>
<td>12</td>
<td>$[0]^*D$, $[1]^*D$</td>
</tr>
<tr>
<td>DTLZ5</td>
<td>3</td>
<td>12</td>
<td>$[0]^*D$, $[1]^*D$</td>
</tr>
<tr>
<td>DTLZ7</td>
<td>3</td>
<td>22</td>
<td>$[0]^*D$, $[1]^*D$</td>
</tr>
<tr>
<td>WFG1</td>
<td>3</td>
<td>12</td>
<td>$[0]^*D$, $[1]^*D$</td>
</tr>
<tr>
<td>WFG2</td>
<td>3</td>
<td>12</td>
<td>$[0]^<em>D$, $[2]^</em>(D - 1) + 1$</td>
</tr>
<tr>
<td>WFG3</td>
<td>3</td>
<td>12</td>
<td>$[0]^<em>D$, $[2]^</em>(D - 1) + 1$</td>
</tr>
<tr>
<td>WFG4</td>
<td>3</td>
<td>12</td>
<td>$[0]^<em>D$, $[2]^</em>(D - 1) + 1$</td>
</tr>
<tr>
<td>WFG5</td>
<td>3</td>
<td>12</td>
<td>$[0]^<em>D$, $[2]^</em>(D - 1) + 1$</td>
</tr>
</tbody>
</table>
Inverse generation distance (IGD) is used to evaluate distribution and convergence of all algorithms (Zhang et al., 2008). The IGD indicator evaluates performance of the algorithm by evaluating degree of coincidence between Pareto solution set generated by algorithm and true PF. The IGD indicator is computed by (Zhang et al., 2008):

\[
IGD(P^*, P) = \frac{\sum_{v \in P^*} d(v, P)}{|P^*|}
\]

where \(P\) is a set of obtained solutions, \(d(v, P)\) is the shortest Euclidean distance between points in \(P\) and \(v\), and \(|P^*|\) is sample points of true PF. When the number of \(P^*\) is large enough, it can better represent true PF. The mean and standard deviation of the IGD results are calculated on each problem. Mean value of IGD represents the performance of algorithm and standard deviation shows stability. All problems are run 25 times independently on each algorithm. The best average result for each function is shown in italics.

The best performance of ROMOABC algorithm was verified by two experiments. The first experiment aims to compare some classical multi-objective algorithms with ROMOABC. The second experiment investigates the effectiveness of each strategy in ROMOABC algorithm. To ensure fairness of the experiments, some parameters are unified. Population size SN is set to 200 in two-objective test function ZDT, 250 in the three-objective test function DTLZ, and 300 in the WFG. The unique limit in the ABC variant is 100. Considering the difference between ABC and other algorithms, the function evaluations times (FEs) is taken as the termination condition. ZDT, DTLZ, and WFG, their corresponding MaxFEs are set to \(5 \times 10^4\), \(1.0 \times 10^5\), and \(1.5 \times 10^5\), respectively.

NSGA-II, MOABC and ROMOABC are implemented by Python in the framework of geatpy (Jazbin, 2020). MOPSO, SPEA2 and MOEA/D are implemented by MATLAB in the framework of platEMO (Tian et al., 2017). All experiments were performed on a computer configured with Intel Core i5-5200U CPU 2.6 GHz processor and 4.0 GB memory.

5.2 Comparison of ROMOABC with other multi-objective algorithms

To verify the performance of ROMOABC, several multi-objective algorithms including MOEA/D (Zhang and Li, 2007), NSGA-II (Deb et al., 2002), MOPSO (Coello et al., 2004), SPEA2 (Zitzler et al., 2001), and MOABC (Akbari et al., 2012) are involved. The detailed parameters of each algorithm are shown in Table 2.

The IGD results of ROMOABC and five other algorithms are shown in Table 3. As seen, ROMOABC has an excellent performance in two-objective problems. It shows that ROMOABC has better and distribution performance convergence on simple PFs. For ZDT1, ZDT2, ZDT4 and ZDT6, ROMOABC achieves the best performance in comparison algorithms. For those three-objective optimisation problems, they generally have complex PFs and the performance of ROMOABC is influenced. ROMOABC performs worse than MOEA/D on DTLZ1 and DTLZ3, but it takes the second place. For DTLZ2, DTLZ4 and DTLZ7, ROMOABC is better than other algorithms. ROMOABC obtain the best results on all WFG problems except for WFG1.

Table 2 Parameter settings of each algorithm

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Parameter settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>NSGA-II</td>
<td>(p_c = 1, p_m = 1/n, \eta_c = 20, \eta_m = 20)</td>
</tr>
<tr>
<td>MOEA/D</td>
<td>(p_c = 1, p_m = 1/n, \eta_c = 20, \eta_m = 20, p_s = 0.9)</td>
</tr>
<tr>
<td>MOPSO</td>
<td>Grid number = 10, archive size = SN</td>
</tr>
<tr>
<td>MOABC</td>
<td>(w_3 = 0.7, w_2 = 0.8, \text{Grid number} = 10, \text{archive size} = \text{SN})</td>
</tr>
<tr>
<td>SPEA2</td>
<td>(p_c = 1, p_m = 1/n, \eta_c = 20, \eta_m = 20, \text{Grid number} = 10)</td>
</tr>
</tbody>
</table>

Table 4 gives the average ranking values obtained by the Friedman test for all algorithms. The best ranking value is shown in italics. As shown, ROMOABC ranks the best among all six multi-objective algorithms.

From the results of Tables 3 and 4, ROMOABC achieves excellent IGD results when compared with MOEA/D, NSGA-II, MOPSO, SPEA2 and MOABC. To observe the quality of final solutions, Figure 1 presents the obtained Pareto fronts of ROMOABC on some test problems. As seen, ROMOABC has good distribution and convergence on selected problems.

5.3 Study on different strategies

ROMOABC consists of three main strategies: modified framework for multi-objective ABC (MF), reference point associated archive method (RP) and elite opposition-based learning (EO). To study the role of different strategies in ROMOABC, five kinds of strategy combinations are constructed as follows:

- MOABC: ABC with the original multi-objective framework
- ABC-MF: ABC with modified multi-objective framework
- ABC-MF-EO: ABC with modified multi-objective framework and elite opposition-based learning
- ABC-MF-RP: ABC with modified multi-objective framework and reference point associated archive method
- ROMOABC: ABC with three improved strategies.
Table 3  IGD results of each algorithm

<table>
<thead>
<tr>
<th>Problem</th>
<th>MOEA/D</th>
<th>NSGA-II</th>
<th>MOPSO</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean</td>
<td>Std</td>
<td>Mean</td>
</tr>
<tr>
<td>ZDT1</td>
<td>3.73E-03</td>
<td>2.69E-03</td>
<td>2.35E-03</td>
</tr>
<tr>
<td>ZDT2</td>
<td>3.80E-03</td>
<td>2.20E-03</td>
<td>2.37E-03</td>
</tr>
<tr>
<td>ZDT3</td>
<td>1.64E-02</td>
<td>2.19E-02</td>
<td>2.65E-03</td>
</tr>
<tr>
<td>ZDT4</td>
<td>5.25E-01</td>
<td>1.96E-01</td>
<td>8.62E-01</td>
</tr>
<tr>
<td>ZDT6</td>
<td>3.09E-03</td>
<td>5.34E-04</td>
<td>1.85E-03</td>
</tr>
<tr>
<td>DTLZ1</td>
<td>1.24E-02</td>
<td>6.60E-05</td>
<td>1.70E-02</td>
</tr>
<tr>
<td>DTLZ2</td>
<td>3.28E-02</td>
<td>4.65E-07</td>
<td>4.36E-02</td>
</tr>
<tr>
<td>DTLZ3</td>
<td>3.50E-02</td>
<td>2.10E-03</td>
<td>4.49E-02</td>
</tr>
<tr>
<td>DTLZ4</td>
<td>4.60E-01</td>
<td>3.86E-01</td>
<td>4.59E-02</td>
</tr>
<tr>
<td>DTLZ5</td>
<td>1.79E-02</td>
<td>7.09E-04</td>
<td>4.31E-03</td>
</tr>
<tr>
<td>DTLZ7</td>
<td>8.82E-02</td>
<td>7.21E-04</td>
<td>4.61E-02</td>
</tr>
<tr>
<td>WFG1</td>
<td>1.77E-01</td>
<td>1.14E-02</td>
<td>1.30E-01</td>
</tr>
<tr>
<td>WFG2</td>
<td>1.54E-01</td>
<td>2.00E-02</td>
<td>1.22E-01</td>
</tr>
<tr>
<td>WFG3</td>
<td>3.95E-01</td>
<td>8.98E-02</td>
<td>1.35E-01</td>
</tr>
<tr>
<td>WFG4</td>
<td>1.50E-01</td>
<td>3.42E-03</td>
<td>1.62E-01</td>
</tr>
<tr>
<td>WFG5</td>
<td>1.52E-01</td>
<td>1.68E-03</td>
<td>1.80E-01</td>
</tr>
<tr>
<td>+/-=/=</td>
<td>13/0/3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Problem</td>
<td>SPEA2</td>
<td>MOABC</td>
<td>ROMOABC</td>
</tr>
<tr>
<td>---------</td>
<td>--------</td>
<td>--------</td>
<td>---------</td>
</tr>
<tr>
<td></td>
<td>Mean</td>
<td>Std</td>
<td>Mean</td>
</tr>
<tr>
<td>ZDT1</td>
<td>2.35E-03</td>
<td>6.96E-05</td>
<td>3.24E-01</td>
</tr>
<tr>
<td>ZDT2</td>
<td>2.38E-03</td>
<td>6.52E-05</td>
<td>9.85E+00</td>
</tr>
<tr>
<td>ZDT3</td>
<td>2.63E-03</td>
<td>7.05E-05</td>
<td>2.44E-01</td>
</tr>
<tr>
<td>ZDT4</td>
<td>8.50E-01</td>
<td>2.88E-01</td>
<td>2.01E-01</td>
</tr>
<tr>
<td>ZDT6</td>
<td>1.84E-03</td>
<td>5.61E-05</td>
<td>2.75E-02</td>
</tr>
<tr>
<td>DTLZ1</td>
<td>1.68E-02</td>
<td>3.71E-04</td>
<td>4.22E-01</td>
</tr>
<tr>
<td>DTLZ2</td>
<td>4.39E-02</td>
<td>1.48E-03</td>
<td>4.66E-02</td>
</tr>
<tr>
<td>DTLZ3</td>
<td>4.45E-02</td>
<td>1.61E-03</td>
<td>1.05E+01</td>
</tr>
<tr>
<td>DTLZ4</td>
<td>6.30E-02</td>
<td>9.03E-02</td>
<td>3.53E-01</td>
</tr>
<tr>
<td>DTLZ5</td>
<td>5.74E-03</td>
<td>9.08E-04</td>
<td>3.96E-03</td>
</tr>
<tr>
<td>DTLZ7</td>
<td>4.54E-02</td>
<td>1.87E-03</td>
<td>4.14E-01</td>
</tr>
<tr>
<td>WFG1</td>
<td>1.50E-01</td>
<td>2.42E-02</td>
<td>2.29E+00</td>
</tr>
<tr>
<td>WFG2</td>
<td>1.21E-01</td>
<td>3.83E-03</td>
<td>5.60E-01</td>
</tr>
<tr>
<td>WFG3</td>
<td>2.01E-01</td>
<td>2.19E-02</td>
<td>4.93E-01</td>
</tr>
<tr>
<td>WFG4</td>
<td>1.59E-01</td>
<td>3.59E-03</td>
<td>4.33E-01</td>
</tr>
<tr>
<td>WFG5</td>
<td>1.78E-01</td>
<td>4.15E-03</td>
<td>2.27E-01</td>
</tr>
<tr>
<td>+/-=/=</td>
<td>14/0/2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4  Average ranking under Friedman test for each algorithm

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Mean ranking</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOEA/D</td>
<td>3.41</td>
</tr>
<tr>
<td>NSGA-II</td>
<td>2.91</td>
</tr>
<tr>
<td>MOPSO</td>
<td>5.83</td>
</tr>
<tr>
<td>SPEA2</td>
<td>2.75</td>
</tr>
<tr>
<td>MOABC</td>
<td>4.94</td>
</tr>
<tr>
<td>ROMOABC</td>
<td>1.63</td>
</tr>
</tbody>
</table>
Table 5  IGD results obtained by multi-objective ABC with different strategies

<table>
<thead>
<tr>
<th>Problem</th>
<th>MOABC</th>
<th>MOABC-MF</th>
<th>ABC-MF-EO</th>
<th>ABC-MF-RP</th>
<th>ROMOABC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean</td>
<td>Std</td>
<td>Mean</td>
<td>Std</td>
<td>Mean</td>
</tr>
<tr>
<td>ZDT1</td>
<td>3.24E-01</td>
<td>5.54E-02</td>
<td>3.18E-03</td>
<td>1.28E-04</td>
<td>1.97E-03</td>
</tr>
<tr>
<td>ZDT2</td>
<td>9.85E+00</td>
<td>2.47E+00</td>
<td>3.80E-03</td>
<td>1.20E-04</td>
<td>3.84E-03</td>
</tr>
<tr>
<td>ZDT3</td>
<td>2.44E-01</td>
<td>2.89E-02</td>
<td>3.44E-03</td>
<td>5.81E-04</td>
<td>3.38E-03</td>
</tr>
<tr>
<td>ZDT4</td>
<td>2.01E-01</td>
<td>8.20E-02</td>
<td>3.30E-03</td>
<td>6.03E-04</td>
<td>3.20E-03</td>
</tr>
<tr>
<td>ZDT6</td>
<td>2.75E-02</td>
<td>8.22E-03</td>
<td>3.04E-03</td>
<td>8.31E-05</td>
<td>3.06E-02</td>
</tr>
<tr>
<td>DTLZ1</td>
<td>4.22E-01</td>
<td>1.18E-01</td>
<td>1.77E-02</td>
<td>3.86E-04</td>
<td>1.74E-02</td>
</tr>
<tr>
<td>DTLZ2</td>
<td>4.66E-02</td>
<td>1.62E-03</td>
<td>4.45E-02</td>
<td>1.02E-03</td>
<td>4.46E-02</td>
</tr>
<tr>
<td>DTLZ3</td>
<td>1.05E+01</td>
<td>2.84E+00</td>
<td>5.22E-02</td>
<td>6.36E-03</td>
<td>5.20E-02</td>
</tr>
<tr>
<td>DTLZ4</td>
<td>3.53E-01</td>
<td>5.42E-02</td>
<td>4.49E-02</td>
<td>3.51E-03</td>
<td>4.45E-02</td>
</tr>
<tr>
<td>DTLZ5</td>
<td>3.96E-03</td>
<td>2.94E-04</td>
<td>2.60E-03</td>
<td>8.37E-05</td>
<td>2.60E-03</td>
</tr>
<tr>
<td>DTLZ7</td>
<td>4.14E-01</td>
<td>7.33E-02</td>
<td>4.67E-02</td>
<td>4.94E-03</td>
<td>4.66E-02</td>
</tr>
<tr>
<td>WFG1</td>
<td>2.29E+00</td>
<td>4.69E-02</td>
<td>2.72E-01</td>
<td>3.68E-02</td>
<td>2.59E-01</td>
</tr>
<tr>
<td>WFG2</td>
<td>5.60E-01</td>
<td>1.94E-02</td>
<td>1.27E-01</td>
<td>3.69E-03</td>
<td>1.26E-01</td>
</tr>
<tr>
<td>WFG3</td>
<td>4.93E-01</td>
<td>1.45E-02</td>
<td>5.19E-02</td>
<td>7.48E-03</td>
<td>5.04E-02</td>
</tr>
<tr>
<td>WFG4</td>
<td>4.33E-01</td>
<td>2.07E-02</td>
<td>1.76E-01</td>
<td>4.13E-03</td>
<td>1.77E-01</td>
</tr>
<tr>
<td>WFG5</td>
<td>2.27E-01</td>
<td>5.35E-03</td>
<td>1.84E-01</td>
<td>3.46E-03</td>
<td>1.86E-01</td>
</tr>
</tbody>
</table>

Figure 1  The obtained Pareto fronts of ROMOABC on some test problems, (a) ZDT1 (b) ZDT4 (c) DTLZ3 (d) DTLZ4 (see online version for colours)
Table 6 Average ranking obtained by Friedman test for multi-objective ABC with different strategies

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Mean ranking</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOABC</td>
<td>4.81</td>
</tr>
<tr>
<td>ABC-MF</td>
<td>3.41</td>
</tr>
<tr>
<td>ABC-MF-EO</td>
<td>3.28</td>
</tr>
<tr>
<td>ABC-MF-RP</td>
<td>1.94</td>
</tr>
<tr>
<td>ROMOABC</td>
<td>1.56</td>
</tr>
</tbody>
</table>

For the above five multi-objective ABC algorithms, the same parameter settings are consistent with the Section 5.2. Table 5 lists IGD results of five ABCs. Result shows that ROMOABC obtains the best IGD values among 16 problems. It indicates that proposed algorithm achieves good stability while improving the optimisation performance. ROMOABC is superior to ABC-MF-EO on 15 test functions. It means the reference point association method plays a positive role in improving the population diversity. Compared with ABC-MF-RP, ROMOABC achieves better or similar results on 13 test functions. It demonstrates the elite opposition is also important for improving the convergence. From the comparison between MOABC and ABC-MF, ABC-MF surpasses MOABC on all 16 problems. It confirms the effectiveness of our proposed method. The above experiments show ROMOABC can improve the population diversity and speed up convergence rate.

In Table 6, Friedman test is used to calculate the average ranking of five multi-objective ABC algorithms. Table 6 shows their average ranking. As seen, ROMOABC gains the best ranking value. It means ROMOABC is the most competitive algorithm among five different multi-objective ABCs. The ranking values of ABC-MF-RP and ABC-MF-EO are better than that of ABC-MF. It demonstrates those two strategies reference point associated archive method and elite opposition-based learning are effective.

6 Conclusions

In MOP, a new ABC algorithm is proposed to improve the convergence and distribution performance. In ROMOABC, there are three main improved strategies. Firstly, the original framework of ABC is modified to improve the efficiency of population renewal and speed up the convergence rate. Then, to reduce waste of computing resources, elite opposition-based learning is used in the scout bee search. Thirdly, a method based on reference point association is used to improve the distribution performance of external archive.

A set of 16 benchmark functions including ZDT, DTLZ and WFG are tested in two experiments. The experiment one between MOEA/D, NSGA-II, MOPSO, SPEA2, MOABC and ROMOABC show that our approach ROMOABC outperforms other algorithms on most test functions. Moreover, another experiment investigates the effects of the above three strategies. Results confirm that the proposed strategies are effective.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (No. 61663028), and the Science and Technology Plan Project of Jiangxi Provincial Education Department (Nos. GJJ170994 and GJJ190958).

References


