European Journal of Operational Research 245 (2015) 168–193

Contents lists available at ScienceDirect

European Journal of Operational Research

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

Decision Support

An elitism based multi-objective artificial bee colony algorithm[☆]

Yi Xiang[a,b], Yuren Zhou[a,*], Hailin Liu[c]

[a] School of Advanced Computing, Sun Yat-sen University, Guangzhou 510275, P.R. China
[b] Department of Advanced Mathematics, Guangdong Baiyun University, Guangzhou 510450, P.R. China
[c] Department of Applied Mathematics, Guangdong University of Technology, Guangzhou 510090, P.R. China

ARTICLE INFO

Article history:
Received 27 April 2014
Accepted 2 March 2015
Available online 6 March 2015

Keywords:
Heuristics
Multi-objective optimization
Elitism strategy
Artificial bee colony algorithm

ABSTRACT

In this paper, we suggest a new multi-objective artificial bee colony (ABC) algorithm by introducing an elitism strategy. The algorithm uses a fixed-size archive that is maintained based on crowding-distance to store non-dominated solutions found during the search process. In the proposed algorithm, an improved artificial bee colony algorithm with an elitism strategy is adopted for the purpose of avoiding premature convergence. Specifically, the elites in the archive are selected and used to generate new food sources in both employed and onlooker bee phases in each cycle. To keep diversity, a member located at the most crowded region will be removed when the archive overflows. The algorithm is very easy to be implemented and it employs only a few control parameters. The proposed algorithm is tested on a wide range of multi-objective problems, and compared with other state-of-the-art algorithms in terms of often-used quality indicators with the help of a nonparametric test. It is revealed by the test procedure that the algorithm produces better or comparable results when compared with other well-known algorithms, and it can be used as a promising alternative tool to solve multi-objective problems with the advantage of being simple and effective.

1. Introduction

Multi-objective optimization is a common problem faced by scientists and engineers, which concerns optimizing problems with multiple and often conflicting objectives. In principle, there is no single solution for a multi-objective optimization problem (MOP), but a set of Pareto-optimal solutions (Deb, Pratap, Agarwal, & Meyarivan, 2002a). This paper considers the following continuous MOP:

$$\begin{aligned} &\mathrm{MinimizeF(x)} = (\mathrm{f_1(x), f_2(x), \ldots, f_m(x)}),\\ &\text{subject to } x \in \prod_{i=1}^{n} [lb_i, ub_i], \end{aligned} \tag{1}$$

where $\prod_{i=1}^{n} [lb_i, ub_i]$ is the decision space. $f : \prod_{i=1}^{n} [lb_i, ub_i] \rightarrow R^m$ consists of m real-valued continuous objective functions $f_1, f_2, \ldots, f_m$. R^m is called the objective space. Unlike single objective optimization, solutions of a MOP are in such a way that the performance of each objective cannot be improved without sacrificing the performance of at least another one. Hence, the solution to a MOP exists in the form of an alternate trade-off known as a Pareto-optimal set. The Pareto-optimal set is defined based on Pareto dominance (Akbari, Hedayatzadeh, Ziarati, & Hassanizadeh, 2012). Let $u = (u_1, u_2, \ldots, u_m)$ and $v = (v_1, v_2, \ldots, v_m)$ be two vectors in the objective space, u is said to dominate v if and only if $u_i \leq v_i$ for each i and there exists at least one index such that $u_i < v_i$. If there is no x in the decision space such that $F(x)$ dominates $F(x^*)$, x^* is called a (globally) Pareto-optimal solution, and $F(x^*)$ is a Pareto-optimal objective vector. The set of all Pareto-optimal solutions is called the *Pareto set* (*PS*) and the set of all Pareto-optimal objective vectors is the *Pareto front* (*PF*) (Liu, Gu, & Zhang, 2014).

Over the past decades, a number of multi-objective evolutionary algorithms (MOEAs) have been suggested. Of them, Knowles and Corne's PAES (Knowles & Corne, 1999), Zitzler *et al.*'s SPEA2 (Zitzler, Laumanns, & Thiele, 2001) and IBEA (Zitzler & Künzli, 2004), Deb's NSGAII (Deb et al., 2002a), Zhang *et al.*'s MOEA/D (Zhang, Liu, & Li, 2009a), Kukkonen and Lampinen's GDE3 (Kukkonen & Lampinen, 2009), Nebro *et al.*'s MOCell (Nebro, Durillo, Luna, Dorronsoro, & Alba, 2009b) enjoyed more attention. Some particle swarm-based multi-objective algorithms, such as Sierra and Coello Coello's OMPSO (Sierra & Coello, 2005), and Nebro *et al.*'s SMPSO (Nebro et al., 2009a) were also well studied. Besides, hybrid algorithms such as Nebro *et al.*'s AbYSS (Nebro et al., 2008) and Durillo *et al.*'s CellDE (Durillo, Nebro, Luna, & Alba, 2008) were also suggested in corresponding literature. All the above multi-objective optimization algorithms were developed for finding multiple Pareto-optimal solutions to approximate the true PF in one single simulation run.

[☆] This paper is jointly supported by the Natural Science Foundation of China (Grant nos. 61472143 and 61170081) and the Foundation for Distinguished Young Talents in Higher Education of Guangdong Province (Grant no. 2014KQNCX236).

[*] Corresponding author. Tel.: +86 20 39380288.
E-mail addresses: gzhuxiang_yi@163.com (Y. Xiang), yrzhou@scut.edu.cn (Y. Zhou).

http://dx.doi.org/10.1016/j.ejor.2015.03.005
0377-2217/© 2015 Elsevier B.V. All rights reserved.

```
Algorithm: eMOABC
◇ Input:
  ◇ $CS$, the size of the bee colony;
  ◇ $AS$, the size of the crowding archive;
  ◇ limit, the abandonment criteria;
  ◇ maxCycle, the termination criteria.
◇ Output:
  ◇ archive.
1. Initialization:
    1.1 Initialize the colony with the parameter $CS$;
    1.2 Initialize crowding-distance archive with the parameter $AS$.
    1.3 Add non-dominated solutions within the initial colony into the archive.
    // Compute crowding-distances for the members in the initial archive.
    1.4 crowdingDistanceAssignment (archive);
2. For (t=0; t< maxCycle-1; t++) {
     2.1 SendEmployedBees (colony, archive);
     2.2 crowdingDistanceAssignment (archive);
     2.3 SendOnlookerBees (colony, archive);
     2.4 crowdingDistanceAssignment (archive);
     2.5 SendScoutBees (colony, limit).
   }
3. Return archive
```

Fig. 1. The flowchart of the eMOABC algorithm.

```
Function 2: crowdingDistanceAssignment (archive)
1. $n$ = |archive| // number of solutions in the archive
  // Initialize distance as 0.0 for each solution $i$
2. For each $i$, archive[$i$].setCrowdingDistance (0.0)
3. For each objective $m$
     // Sort archive using each objective value
     3.1. archive = sort (archive,$m$)
     3.2. archive[1].setCrowdingDistance ($\infty$)
     3.3. archive[$n$].setCrowdingDistance ($\infty$)
     3.4. For $i$ = 2 to $n$-1
            distance = archive[$i$+1].$m$ − archive[$i$-1].$m$
            distance = distance/(objMax.$m$ − objMin.$m$)
            // (objMax.$m$-objMin.$m$), range of the $m$th objective
            archive[i].setCrowdingDistance (distance)
          End For $i$
   End For $m$
End of crowdingDistanceAssignment (archive)
```

Fig. 2. The pseudo code of crowdingDistanceAssignment (*archive*).

```
Function 1: SendEmployedBees (colony, archive)
Select the elite from the archive
For $i = 1$ to $FN$
    Determine one dimension $d$ to be modified randomly
    Select a neighbor $x_k$ from the colony stochastically
    For food source $x_i$, calculate its new position $v_i$
      $v_{id} = x_{id} + \phi_{id} \cdot (x_{id} - x_{kd}) + \varphi_{id} \cdot (x_{id} - elite_d)$
    Evaluate $v_i$
    If $v_i$ dominates $x_i$
        $x_i \leftarrow v_i$
        $trial_i = 0$
        Add $v_i$ into the archive.
    Else If $x_i$ and $v_i$ are non-dominated with each other
        If $v_i$ is successfully added into the archive
            $x_i \leftarrow v_i$
            $trial_i = 0$
        Else
            $trial_i = trial_i + 1$
        End If
    Else
        $trial_i = trial_i + 1$
    End If
End For
Use the function calculateFitness (colony, archive) to compute fitness
value of each food source
End of SendEmployedBees (colony, archive)
```

Fig. 3. The pseudo code of SendEmployedBees (*colony*, *archive*).

```
If $v_{id} < lb_d$
    If $rand(0, 1) \leq 0.5$
        $v_{id} = lb_d$
    Else
        $v_{id} = lb_d + 10^{-3} * rand(0, 1) * (ub_d - lb_d)$
    End If
End If
If $v_{id} > ub_d$
    If $rand(0, 1) \leq 0.5$
        $v_{id} = ub_d$
    Else
        $v_{id} = ub_d - 10^{-3} * rand(0, 1) * (ub_d - lb_d)$
    End If
End If
```

Fig. 4. The boundary treatment method used in the eMOABC algorithm.

The artificial bee colony (ABC) algorithm, proposed by Karaboga and Basturk (2007), is one of the most recently introduced swarm-based search methods (Akay & Karaboga, 2012; Szeto, Wu, & Ho, 2011). In ABC, there are three kinds of bees who do different tasks to make the algorithm useful. The employed bees will be sent to food sources and try to improve them by using neighbor information. Each onlooker bee will choose one of those food sources by a greedy strategy based on the quality information of food sources shared by employed bees, and then try to improve it. Finally, a scout bee will find a food source which has not been optimized in a limited number of cycles so as to re-initialize it to get rid of the poor solution. The ABC seems particularly suitable for multi-objective optimization mainly because it obtains high-quality solutions and shows fast speed of convergence for single-objective optimization (Akbari et al., 2012).

In recent years, extending basic ABC for handling multi-objective problems has received more attention from researchers. Some excellent works have been reported: Hedayatzadeh et al. proposed a multi-objective artificial bee colony (MOABC) which has adapted the basic ABC algorithm to multi-objective problems with a grid-based approach for maintaining and adaptively assessing the Pareto front. In this algorithm, the ε-dominance method was used to update the external archive (Hedayatzadeh, Hasanizadeh, Akbari, & Ziarati, 2010). The MOABC was compared with MOPSO (Coello Coello, Pulido, & Lechuga, 2004) and NSGAII (Deb et al., 2002a) on six test functions in terms of two performance metrics; Omkar et al. presented a method called the Vector-Evaluated ABC (or VEABC) for the multi-objective design of composite structures (Omkar, Senthilnath, Khandelwal, Naik, & Gopalakrishnan, 2011). In the VEABC, multiple populations were used to concurrently optimize the problem at hand. Zou et al. presented a novel artificial bee colony (ABC) algorithm for solving multi-objective optimization problems. In this algorithm, they

```
Function 3: SendOnlookerBees (colony, archive)
 Calculate the selection probability prob by Eq. (4) for each food source
 Select the elite from the archive
 $i$=1, $t = 1$;
 While $t \le FN$
     If $rand < prob_i$
         $t = t + 1$
         Determine one dimension $d$ to be modified randomly
         Select a neighbor $x_k$ from the colony stochastically
         For food source $x_i$, calculate its new position $v_i$
           $v_{id} = x_{id} + \phi_{id} \cdot (x_{id} - x_{kd}) + \varphi_{id} \cdot (x_{id} - elite_d)$
         Evaluate $v_i$
         If $v_i$ dominates $x_i$
             $x_i \leftarrow v_i$
             $trial_i = 0$
             Add $v_i$ into the archive.
         Else If $x_i$ and $v_i$ are non-dominated with each other
             If $v_i$ is successfully added into the archive
                 $x_i \leftarrow v_i$
                 $trial_i = 0$
             Else
                 $trial_i$= $trial_i + 1$
             End If
         Else
              $trial_i$= $trial_i + 1$
         End If
         $i = i + 1$
         If $i == FN + 1$, then $i = 1$
     End If
 End While
End of SendOnlookerBees (colony, archive)
```

Fig. 5. The pseudo code of SendOnlookerBees (*colony*, *archive*).

```
Function 4: SendScoutBees (colony, limit)
 Determine one food source $x_k$ with maximum trial value in the colony
 If $trial_k \ge limit$
     $x_{kd} = lb_d + rand(0, 1) \cdot (ub_d - lb_d), d = 1, 2, \ldots, n.$
     Evaluate the new food source $x_k$
     $trial_k = 0$
 End If
End of SendScoutBees (limit)
```

Fig. 6. The pseudo code of SendScoutBees (*colony*, *limit*).

used the Pareto concept and an external archive strategy to make the algorithm converge to the true Pareto optimal front. Moreover, they adopted a comprehensive learning strategy to ensure the diversity of the population (Zou, Zhu, Chen, & Zhang, 2011). Their proposed algorithm was compared with other three competitors on seven two-objective test functions and six three-objective functions by considering both convergence and diversity metrics. Akbari et al. proposed a new type of MOABC utilizing different types of bees (i.e. employed bees, onlookers, and scouts) and a fixed-sized archive which was maintained by an ε-dominance method. In this algorithm, the social information provided by the external archive was used by the employed bees to adjust their flying trajectories. The diversity over the external archive was controlled using a grid. The onlookers evaluate the solutions provided by the employed bees to adjust their next position. Finally, the scout bees replaced the solutions who had reached a trial limit by the new random solution in the search space (Akbari et al., 2012). Akay suggested three multi-objective ABC algorithms based on synchronous and asynchronous models using Pareto-dominance and non-dominated sorting: A-MOABC/PD, A-MOABC/NS and S-MOABC/NS (Akay, 2013). These algorithms were evaluated on 10 unconstrained test instances from the CEC2009 competition (Zhang et al., 2009b), and compared with three algorithms in terms of performance metrics: inverted generational distance (IGD) (Zhang et al., 2009a), hypervolume (HV) (Zitzler & Thiele, 1999) and SPREAD (Deb, 2001). In our previous work, we proposed a new multi-objective artificial bee colony algorithm called dMOABC (Zhong, Xiang, & Liu, 2014) by dividing the whole search space S into two independent parts S_1 and S_2. In the dMOABC, two "basic" colonies are assigned to search potential solutions in regions S_1 and $S_2,$ while the so-called "synthetic" colony explores in S. Benefiting from this multi-colony model, the dMOABC showed significant improvement over its competitors.

In this paper, we suggest a simple yet effective multi-objective artificial bee colony algorithm by using an improved ABC algorithm with an elitism strategy, abbreviated as eMOABC. In this algorithm, a crowding-distance archive used in NSGAII (Deb et al., 2002a) is adopted to keep a good spread of the obtained solutions. In each iteration, the eMOABC algorithm selects two elites defined as the archived intermediate solution with maximum value of crowding-distance, and then uses them to adjust the flying trajectories of employed and onlooker bees, respectively. Our eMOABC is very easy to be implemented and it does not introduce additional control parameters compared to the basic single-objective ABC algorithm.

Main features of our eMOABC are:

- The introduced elitism strategy. In the basic ABC algorithm, both employed and onlooker bees use only neighbor information to produce new food sources. Through this mechanism, the whole bee colony may be easily trapped into a poor region once the neighbors are located around a local optimum. However, the eMOABC uses both neighbors and the elites to guide the flying trajectories of the bees, which is an analogy to the idea in the PSO algorithm (Kennedy & Eberhart, 1995). In the employed bees phase, an intermediate solution (opposite to boundary solutions that are assigned an infinite distance value; Deb et al., 2002a) with maximum crowding-distance value is selected as the elite, and is used to produce new food sources. After the whole bee colony is updated, the crowding-distance values are computed again. Thereafter, another elite is determined and will be exploited in the following onlooker bees phase. The above procedures are repeated until the algorithm is terminated. The elitism strategy could improve the exploitation ability of the eMOABC algorithm, since the regions where elites are located will be exploited *FN* times by employed and onlooker bees. Hence, the whole bee colony may be pulled toward the sparsest zone. The advantages of this lie in : (1) exploiting more potential non-dominated solutions, and (2) preserving the diversity of solutions in the approximated set. The effect of the elitism strategy is experimentally proved in Section 4.
- A few control parameters. Actually, there are only two important control parameters *limit* (the abandonment criteria) and *CS* (the size of the bee colony) to be tuned manually in the eMOABC algorithm. A sensitivity analysis experiment is designed to investigate the influence of *limit* and *CS* on the performance of the algorithm. Based on the simulation results shown in Section 3.3, one is highly recommended to set the value of *limit* and *CS* on the ranges $50 \sim 200,$ and $20 \sim 60$ for general usage, respectively.
- The eMOABC achieves a good performance with less control parameters than other state-of-the-art algorithms. As previously stated, there are only two important parameters to be adjusted by the user, much less than those of MOEA/D (seven parameters) (Zhang et al., 2009a), MTS and DMOEADD (both have five

Table 1
Mean values of HV for *eMOABC* algorithm family.

	eMOABCa	eMOABCb	eMOABCc	eMOABCd	eMOABCe	eMOABCf
UF1	5.197e − 01	6.435e − 01	6.430e − 01	6.399e − 01	6.386e − 01	6.355e − 01
UF2	5.841e − 01	6.522e − 01	6.497e − 01	6.467e − 01	6.453e − 01	6.446e − 01
UF3	1.670e − 01	4.562e − 01	4.981e − 01	5.179e − 01	5.137e − 01	5.235e − 01
UF4	2.404e − 01	2.659e − 01	2.683e − 01	2.691e − 01	2.697e − 01	2.687e − 01
UF5	0.000e + 00	1.736e − 01	2.579e − 01	2.895e − 01	2.818e − 01	2.931e − 01
UF6	9.023e − 03	1.560e − 01	2.241e − 01	2.652e − 01	2.724e − 01	2.757e − 01
UF7	3.225e − 01	4.658e − 01	4.704e − 01	4.721e − 01	4.666e − 01	4.663e − 01
UF8	1.176e − 01	3.332e − 01	3.388e − 01	3.258e − 01	3.225e − 01	3.158e − 01
UF9	2.774e − 01	6.457e − 01	6.657e − 01	6.536e − 01	6.554e − 01	6.573e − 01
UF10	0.000e + 00	7.716e − 02	1.497e − 01	1.740e − 01	1.952e − 01	1.941e − 01
ZDT1	5.634e − 01	6.618e − 01	6.620e − 01	6.620e − 01	6.620e − 01	6.620e − 01
ZDT2	2.059e − 01	3.285e − 01	3.287e − 01	3.287e − 01	3.287e − 01	3.287e − 01
ZDT3	4.472e − 01	5.150e − 01	5.155e − 01	5.156e − 01	5.156e − 01	5.157e − 01
ZDT4	0.000e + 00	6.620e − 01	6.620e − 01	6.620e − 01	6.620e − 01	6.620e − 01
ZDT6	2.764e − 01	4.014e − 01	4.014e − 01	4.014e − 01	4.014e − 01	4.014e − 01
WFG1	1.389e − 01	6.000e − 01	5.997e − 01	6.022e − 01	6.011e − 01	6.004e − 01
WFG2	5.374e − 01	5.548e − 01	5.568e − 01	5.515e − 01	5.539e − 01	5.486e − 01
WFG3	4.361e − 01	4.394e − 01	4.381e − 01	4.373e − 01	4.358e − 01	4.349e − 01
WFG4	2.179e − 01	2.193e − 01	2.193e − 01	2.193e − 01	2.193e − 01	2.193e − 01
WFG5	1.965e − 01	1.967e − 01	1.972e − 01	1.971e − 01	1.968e − 01	1.968e − 01
WFG6	1.997e − 01	2.049e − 01	2.042e − 01	2.033e − 01	2.022e − 01	1.990e − 01
WFG7	2.086e − 01	2.108e − 01	2.108e − 01	2.108e − 01	2.108e − 01	2.107e − 01
WFG8	1.464e − 01	1.727e − 01	1.680e − 01	1.666e − 01	1.619e − 01	1.630e − 01
WFG9	2.335e − 01	2.374e − 01	2.372e − 01	2.371e − 01	2.370e − 01	2.370e − 01
DTLZ1	5.268e − 03	7.711e − 01	7.721e − 01	7.717e − 01	7.717e − 01	7.724e − 01
DTLZ2	3.881e − 01	3.953e − 01	3.941e − 01	3.942e − 01	3.935e − 01	3.926e − 01
DTLZ3	0.000e + 00	3.546e − 01	3.947e − 01	3.932e − 01	3.930e − 01	3.935e − 01
DTLZ4	2.954e − 01	3.503e − 01	3.298e − 01	3.034e − 01	2.933e − 01	2.568e − 01
DTLZ5	9.371e − 02	9.403e − 02	9.402e − 02	9.402e − 02	9.403e − 02	9.404e − 02
DTLZ6	4.717e − 02	9.498e − 02	9.498e − 02	9.497e − 02	9.498e − 02	9.497e − 02
DTLZ7	2.419e − 01	2.964e − 01	2.936e − 01	2.935e − 01	2.927e − 01	2.916e − 01

parameters) (Liu, Zou, Chen, & Wu, 2009; Tseng & Chen, 2009), and dMOABC (three parameters) (Zhong et al., 2014). In the practical application of one algorithm, much more efforts may be made by the user to find the best parameter settings. It is especially true when the algorithm employs more parameters or is sensitive to some of them. Thus, we are confident that our eMOABC is more suitable for applications in real-world problems from this perspective.

Based on the above two main features, we are confident that our proposed algorithm will be simple and effective. We conduct a comprehensive experiment to compare the eMOABC with 11 state-of-the-art multi-objective algorithms (MOEA/D; Zhang et al., 2009a, SMPSO; Nebro et al., 2009a, GDE3; Kukkonen & Lampinen, 2009, AbYSS; Nebro et al., 2008, CellDE; Durillo et al., 2008, IBEA; Zitzler & Künzli, 2004, MOCell; Nebro et al., 2009b, OMPSO; Sierra & Coello, 2005, NSGAII; Deb et al., 2002a, PAES; Knowles & Corne, 1999 and SPEA2; Zitzler et al., 2001) on 31 test functions, and they are the CEC2009 competition unconstrained function family (Zhang et al., 2009b) (UF1-UF10), ZDT benchmarks (Zitzler, Deb, & Thiele, 2000) (ZDT1-ZDT4, ZDT6), WFG series (Huband, Hingston, Barone, & While, 2006) (WFG1–WFG9) and DTLZ benchmarks (Deb, Thiele, Laumanns, & Zitzler, 2002b) (DTLZ1–DTLZ7). For the purpose of comparison, four performance indicators: HV (Zitzler & Thiele, 1999), SPREAD (Akay, 2013; Durillo & Nebro, 2011), EPSILON (Akay, 2013; Durillo & Nebro, 2011) and IGD (Zhang et al., 2009a) are selected and computed for each algorithm and each problem. It is shown by the simulation results that the eMOABC can provide better approximated fronts with high quality non-dominated solutions. Besides, the results of eMOABC are further compared with those obtained by the famous algorithms appearing in the CEC09 competition (MOEA/D; Zhang et al., 2009a, MTS; Tseng & Chen, 2009, DMOEADD; Liu et al., 2009 and LiuLiAlgorithm; Liu & Li, 2009), and the latest multi-objective artificial bee colony algorithms (S-MOABC/NS; Akay, 2013 and dMOABC; Zhong et al., 2014). The Wilcoxon signed ranks test (Derrac, Garcia, Molina, & Herrera, 2011) results show that the eMOABC is significantly better than or at least comparable to the state-of-the-art algorithms and can be used as a promising alternative tool to solve multi-objective problems with the advantage of being simple yet effective.

In the remainder of the paper, we describe our eMOABC algorithm in Section 2 with more details. Thereafter, in Section 3 we first highlight the sensitivity analysis on the parameters *limit* and *CS*, and then present and analyze the simulation results of the eMOABC in comparison with other 11 state-of-the-art multi-objective algorithms. In Section 4, the effect of the elitism strategy is well studied through a simulation experiment. And in Section 5, the eMOABC is further compared with the latest MOABC algorithms, as well as other competitive MOEAs on the benchmarks in the CEC09 competition. Finally, we outline the conclusions of the paper in Section 6.

Table 2
Mean values of SPREAD for *eMOABC* algorithm family.

	eMOABCa	eMOABCb	eMOABCc	eMOABCd	eMOABCe	eMOABCf
UF1	$6.501e-01$	$6.514e-01$	$7.576e-01$	$8.343e-01$	$8.965e-01$	$9.239e-01$
UF2	$5.698e-01$	$6.676e-01$	$7.886e-01$	$8.672e-01$	$8.969e-01$	$8.698e-01$
UF3	$7.638e-01$	$6.999e-01$	$7.714e-01$	$7.644e-01$	$8.148e-01$	$7.928e-01$
UF4	$5.356e-01$	$5.793e-01$	$5.923e-01$	$5.931e-01$	$6.078e-01$	$6.425e-01$
UF5	$7.761e-01$	$6.659e-01$	$6.204e-01$	$5.584e-01$	$5.667e-01$	$5.605e-01$
UF6	$7.471e-01$	$7.802e-01$	$7.535e-01$	$7.230e-01$	$7.423e-01$	$7.624e-01$
UF7	$6.714e-01$	$8.335e-01$	$8.840e-01$	$9.188e-01$	$9.151e-01$	$9.452e-01$
UF8	$8.166e-01$	$7.607e-01$	$8.608e-01$	$9.438e-01$	$9.521e-01$	$9.574e-01$
UF9	$7.818e-01$	$8.555e-01$	$9.066e-01$	$9.253e-01$	$9.589e-01$	$9.345e-01$
UF10	$7.547e-01$	$9.171e-01$	$1.094e+00$	$1.174e+00$	$1.206e+00$	$1.203e+00$
ZDT1	$5.304e-01$	$1.425e-01$	$1.164e-01$	$1.150e-01$	$1.146e-01$	$1.121e-01$
ZDT2	$5.962e-01$	$1.446e-01$	$1.124e-01$	$1.094e-01$	$1.128e-01$	$1.085e-01$
ZDT3	$7.145e-01$	$8.495e-01$	$7.997e-01$	$7.831e-01$	$7.811e-01$	$7.735e-01$
ZDT4	$9.647e-01$	$1.026e-01$	$9.142e-02$	$9.540e-02$	$9.054e-02$	$8.982e-02$
ZDT6	$1.164e+00$	$1.340e-01$	$1.309e-01$	$9.157e-02$	$1.358e-01$	$1.322e-01$
WFG1	$1.164e+00$	$8.680e-01$	$8.685e-01$	$8.519e-01$	$8.545e-01$	$8.650e-01$
WFG2	$9.918e-01$	$1.010e+00$	$8.984e-01$	$8.798e-01$	$9.160e-01$	$9.196e-01$
WFG3	$5.229e-01$	$3.954e-01$	$4.013e-01$	$4.046e-01$	$4.189e-01$	$4.190e-01$
WFG4	$1.899e-01$	$1.369e-01$	$1.355e-01$	$1.404e-01$	$1.349e-01$	$1.382e-01$
WFG5	$1.422e-01$	$1.444e-01$	$1.481e-01$	$1.431e-01$	$1.445e-01$	$1.461e-01$
WFG6	$4.173e-01$	$1.565e-01$	$1.483e-01$	$1.613e-01$	$1.544e-01$	$1.956e-01$
WFG7	$2.104e-01$	$1.111e-01$	$1.179e-01$	$1.119e-01$	$1.203e-01$	$1.120e-01$
WFG8	$8.325e-01$	$6.829e-01$	$6.672e-01$	$6.701e-01$	$6.473e-01$	$7.087e-01$
WFG9	$2.406e-01$	$1.755e-01$	$1.819e-01$	$1.809e-01$	$1.878e-01$	$1.863e-01$
DTLZ1	$1.005e+00$	$7.314e-01$	$7.164e-01$	$7.273e-01$	$7.120e-01$	$7.230e-01$
DTLZ2	$6.371e-01$	$6.254e-01$	$6.357e-01$	$6.262e-01$	$6.351e-01$	$6.210e-01$
DTLZ3	$8.407e-01$	$6.696e-01$	$6.223e-01$	$6.276e-01$	$6.290e-01$	$6.258e-01$
DTLZ4	$1.012e+00$	$9.695e-01$	$9.999e-01$	$1.068e+00$	$1.167e+00$	$1.213e+00$
DTLZ5	$1.960e-01$	$1.248e-01$	$1.261e-01$	$1.254e-01$	$1.268e-01$	$1.248e-01$
DTLZ6	$9.069e-01$	$1.186e-01$	$1.109e-01$	$1.070e-01$	$1.152e-01$	$1.066e-01$
DTLZ7	$7.458e-01$	$7.022e-01$	$7.205e-01$	$7.195e-01$	$7.424e-01$	$7.114e-01$

2. Elitism strategy based MOABC, *eMOABC*

The eMOABC is designed based on an improved ABC algorithm, and it uses an external archive to store non-dominated solutions found in the search process. Some user-defined control parameters that should be determined beforehand are as follows:

- *CS*, the size of the bee colony.
- *limit*, the abandonment criteria. A food source which could not be improved through "*limit*" trials will be abandoned.

In both the basic ABC and eMOABC, the above two parameters have influence on the algorithms' performance, and the sensitivity analysis on these parameters will be shown in Section 3.3. In the eMOABC, two additional parameters that are commonly used in the multi-objective evolutionary or swarm-based algorithms are:

- *AS*, the size of the external archive or the number of the returned non-dominated solutions used for computing performance metrics.
- *maxCycle*, the termination criteria. The value of this parameter could be tuned according to the maximal number of function evaluations ($maxFEs$) and CS, that is, $maxCycle = maxFEs/CS$.

Actually, values of the above additional parameters are usually fixed and kept the same for all algorithms in order to make a fair comparison. Hence, they are not the algorithms' control parameters that are adjusted by the users. The flow chart of the eMOABC is given in Fig. 1. The eMOABC algorithm consists of six main parts: *initialization*, *crowding-distance assignment*, *maintenance of the crowding-distance archive*, *send employed bees*, *send onlooker bees* and *send scout bees*. In the following subsections, we are going to describe them in details.

2.1. Initialization

In the eMOABC algorithm, each food source represents a potential solution to the considered problem, and the number of food sources (denoted as FN) is the half of the colony size (CS). Specifically, for an n-dimensional m-objective MOP, $FN = CS/2$ food sources, constituting the bee colony *colony*, will be randomly generated in decision space $\prod_{i=1}^{n}[lb_i, ub_i]$ in the initialization phase. In this way, a random n-dimensional vector $x_i = (x_{i1}, x_{i2}, \ldots, x_{in})$ will be assigned to food source i through the following equation:

$$x_{id} = lb_d + rand(0,1)\cdot(ub_d - lb_d), \tag{2}$$

where $i = 1, 2, \ldots, FN$; $d = 1, 2, \ldots, n$, and $rand(0,1)$ is a random number distributed uniformly over the interval [0,1]; lb_d and ub_d are lower and upper bounds of the dth dimension, respectively. Up to now, the initialization of the *colony* is accomplished.

Table 3
Mean values of EPSILON for *eMOABC* algorithm family.

	eMOABCa	eMOABCb	eMOABCc	eMOABCd	eMOABCe	eMOABCf
UF1	$1.485e-01$	$3.744e-02$	$4.387e-02$	$5.052e-02$	$5.136e-02$	$5.912e-02$
UF2	$8.495e-02$	$3.114e-02$	$4.746e-02$	$4.989e-02$	$5.057e-02$	$5.924e-02$
UF3	$4.320e-01$	$2.566e-01$	$2.366e-01$	$2.119e-01$	$2.255e-01$	$2.063e-01$
UF4	$7.325e-02$	$5.812e-02$	$5.669e-02$	$5.741e-02$	$5.812e-02$	$6.092e-02$
UF5	$1.160e+00$	$3.374e-01$	$2.302e-01$	$2.149e-01$	$2.153e-01$	$2.013e-01$
UF6	$6.578e-01$	$3.240e-01$	$2.794e-01$	$2.506e-01$	$2.664e-01$	$2.646e-01$
UF7	$2.086e-01$	$7.234e-02$	$6.577e-02$	$6.402e-02$	$7.987e-02$	$7.141e-02$
UF8	$6.282e-01$	$2.277e-01$	$2.222e-01$	$2.408e-01$	$2.535e-01$	$2.803e-01$
UF9	$5.473e-01$	$2.138e-01$	$1.945e-01$	$2.454e-01$	$2.182e-01$	$2.111e-01$
UF10	$1.960e+00$	$8.039e-01$	$6.110e-01$	$6.132e-01$	$5.759e-01$	$6.096e-01$
ZDT1	$9.977e-02$	$6.903e-03$	$6.177e-03$	$6.169e-03$	$6.108e-03$	$6.144e-03$
ZDT2	$1.839e-01$	$7.015e-03$	$6.033e-03$	$5.974e-03$	$5.973e-03$	$5.977e-03$
ZDT3	$1.583e-01$	$9.169e-03$	$7.807e-03$	$6.442e-03$	$6.019e-03$	$6.100e-03$
ZDT4	$5.261e+00$	$5.844e-03$	$5.734e-03$	$5.716e-03$	$5.688e-03$	$5.648e-03$
ZDT6	$2.263e-01$	$5.451e-03$	$5.343e-03$	$5.221e-03$	$5.339e-03$	$5.227e-03$
WFG1	$1.158e+00$	$2.131e-01$	$2.124e-01$	$2.016e-01$	$2.104e-01$	$2.097e-01$
WFG2	$1.079e-01$	$5.195e-02$	$4.448e-02$	$3.810e-02$	$4.143e-02$	$4.900e-02$
WFG3	$2.018e+00$	$2.006e+00$	$2.010e+00$	$2.011e+00$	$2.016e+00$	$2.018e+00$
WFG4	$2.044e-02$	$1.407e-02$	$1.403e-02$	$1.426e-02$	$1.412e-02$	$1.431e-02$
WFG5	$6.320e-02$	$6.299e-02$	$6.497e-02$	$6.230e-02$	$6.286e-02$	$6.626e-02$
WFG6	$4.394e-02$	$2.413e-02$	$2.545e-02$	$2.685e-02$	$2.831e-02$	$3.543e-02$
WFG7	$2.209e-02$	$1.429e-02$	$1.427e-02$	$1.450e-02$	$1.443e-02$	$1.460e-02$
WFG8	$3.870e-01$	$3.253e-01$	$3.054e-01$	$3.205e-01$	$3.499e-01$	$3.401e-01$
WFG9	$3.041e-02$	$2.352e-02$	$2.295e-02$	$2.373e-02$	$2.372e-02$	$2.416e-02$
DTLZ1	$1.639e+00$	$4.723e-02$	$4.751e-02$	$4.665e-02$	$4.715e-02$	$4.778e-02$
DTLZ2	$1.179e-01$	$1.138e-01$	$1.151e-01$	$1.130e-01$	$1.171e-01$	$1.194e-01$
DTLZ3	$4.289e+01$	$1.733e-01$	$1.129e-01$	$1.173e-01$	$1.123e-01$	$1.181e-01$
DTLZ4	$2.034e-01$	$1.584e-01$	$1.811e-01$	$2.313e-01$	$2.214e-01$	$2.925e-01$
DTLZ5	$5.548e-03$	$4.664e-03$	$4.752e-03$	$4.729e-03$	$4.755e-03$	$4.668e-03$
DTLZ6	$1.877e-01$	$4.333e-03$	$4.245e-03$	$4.302e-03$	$4.240e-03$	$4.277e-03$
DTLZ7	$3.701e-01$	$1.251e-01$	$1.863e-01$	$2.604e-01$	$2.380e-01$	$2.808e-01$

Thereafter, a variable $trial_i$ will be assigned to each food source i in order to find food sources to be abandoned in the next iterations. Variable $trial_i$ is a counter of unsuccessful trials for food source i, and each $trial_i, i = 1, 2, \ldots, FN$ is set to be 0 in the initialization phase. A food source could not get optimized in a number of trials (i.e. *limit*), its related employed bee will turn into a scout bee and after doing a random search, it will turn back to be an employed bee again.

Next, a crowding-distance based external archive *archive* with the size of *AS* is prepared for holding non-dominated solutions found by the algorithm. In the initialization stage, we add all non-dominated food sources in the *colony* into the *archive* whose maintenance method is shown in Section 2.3. Finally, the crowding-distance is computed for each member in the initial *archive* so as to determine the elite in the following employed bee phase. The details of the calculation of the crowding-distance will be shown in Section 2.2.

2.2. Crowding-distance assignment

The crowding-distance used in NSGAII (Deb et al., 2002a) is employed to get an estimate of the density of solutions surrounding a particular solution in the population. This quantity serves as an estimate of the perimeter of the cuboid formed by using the nearest neighbors as the vertices. The computation of the crowding-distance goes as follows: First, sort the population according to each objective function value in ascending order of magnitude. Second, for each objective function, the boundary solutions (solutions with smallest and largest function values) are assigned an infinite distance value. Third, all other intermediate solutions are assigned a distance value equal to the absolute normalized difference in the function values of two adjacent solutions. Finally, the above calculation is continued with other objective functions. The overall crowding-distance value is calculated as the sum of individual distance values corresponding to each objective.

The pseudo code of crowding-distance assignment algorithm is shown in Fig. 2.

2.3. Maintenance of the crowding-distance archive

As opposed to single-objective optimization, multi-objective algorithms usually maintain a non-dominated solutions set. In multi-objective optimization, for the absence of preference information, none of the non-dominated solutions can be said to be better than the others. Therefore, in our algorithm, we use an external archive to preserve diversity of the non-dominated vectors found along the search process (Zou et al., 2011). This technique is used in many algorithms, such as SMPSO (Nebro et al., 2009a), NSGAII (Deb et al., 2002a), PAES (Knowles & Corne, 1999), SPEA2 (Zitzler et al., 2001) and etc.

In our algorithm, the external *archive* is maintained based on crowding-distance method employed in NSGAII (Deb et al., 2002a).

Table 4
Mean values of IGD for *eMOABC* algorithm family.

	eMOABCa	eMOABCb	eMOABCc	eMOABCd	eMOABCe	eMOABCf
UF1	8.739e − 02	1.386e − 02	1.480e − 02	1.799e − 02	1.917e − 02	2.119e − 02
UF2	4.969e − 02	1.003e − 02	1.407e − 02	1.662e − 02	1.780e − 02	2.001e − 02
UF3	4.142e − 01	1.398e − 01	1.075e − 01	9.579e − 02	9.332e − 02	9.363e − 02
UF4	6.142e − 02	4.288e − 02	4.123e − 02	4.008e − 02	4.003e − 02	4.009e − 02
UF5	1.224e + 00	2.427e − 01	1.625e − 01	1.368e − 01	1.368e − 01	1.291e − 01
UF6	6.214e − 01	2.936e − 01	2.174e − 01	1.763e − 01	1.671e − 01	1.623e − 01
UF7	1.123e − 01	2.094e − 02	1.872e − 02	1.788e − 02	2.517e − 02	2.140e − 02
UF8	2.547e − 01	1.091e − 01	1.218e − 01	1.392e − 01	1.475e − 01	1.574e − 01
UF9	2.915e − 01	6.785e − 02	6.641e − 02	7.458e − 02	7.563e − 02	7.225e − 02
UF10	2.161e + 00	4.134e − 01	2.950e − 01	2.760e − 01	2.548e − 01	2.520e − 01
ZDT1	6.070e − 02	3.798e − 03	3.737e − 03	3.738e − 03	3.731e − 03	3.730e − 03
ZDT2	8.994e − 02	3.926e − 03	3.850e − 03	3.854e − 03	3.858e − 03	3.846e − 03
ZDT3	6.281e − 02	5.838e − 03	5.164e − 03	4.983e − 03	4.955e − 03	4.904e − 03
ZDT4	4.952e + 00	3.723e − 03	3.710e − 03	3.702e − 03	3.700e − 03	3.712e − 03
ZDT6	1.134e − 01	3.038e − 03	3.025e − 03	3.022e − 03	3.027e − 03	3.024e − 03
WFG1	1.074e + 00	9.941e − 02	1.018e − 01	9.492e − 02	9.718e − 02	1.004e − 01
WFG2	6.947e − 02	2.759e − 02	2.136e − 02	2.664e − 02	2.610e − 02	3.266e − 02
WFG3	1.511e − 01	1.432e − 01	1.462e − 01	1.485e − 01	1.526e − 01	1.551e − 01
WFG4	1.115e − 02	1.035e − 02	1.037e − 02	1.037e − 02	1.034e − 02	1.037e − 02
WFG5	6.576e − 02	6.546e − 02	6.461e − 02	6.478e − 02	6.529e − 02	6.533e − 02
WFG6	2.903e − 02	1.846e − 02	1.915e − 02	2.060e − 02	2.238e − 02	2.801e − 02
WFG7	1.343e − 02	1.182e − 02	1.182e − 02	1.183e − 02	1.179e − 02	1.188e − 02
WFG8	1.350e − 01	8.545e − 02	8.937e − 02	9.244e − 02	9.831e − 02	1.008e − 01
WFG9	2.000e − 02	1.485e − 02	1.504e − 02	1.526e − 02	1.535e − 02	1.536e − 02
DTLZ1	1.924e + 00	2.277e − 02	2.258e − 02	2.277e − 02	2.257e − 02	2.259e − 02
DTLZ2	6.085e − 02	6.024e − 02	6.050e − 02	6.090e − 02	6.047e − 02	6.118e − 02
DTLZ3	5.343e + 01	1.358e − 01	6.253e − 02	6.315e − 02	6.222e − 02	6.260e − 02
DTLZ4	5.396e − 02	5.148e − 02	6.262e − 02	8.337e − 02	8.229e − 02	1.211e − 01
DTLZ5	4.237e − 03	4.042e − 03	4.017e − 03	4.028e − 03	4.053e − 03	4.025e − 03
DTLZ6	1.295e − 01	3.903e − 03	3.934e − 03	3.930e − 03	3.895e − 03	3.940e − 03
DTLZ7	1.182e − 01	7.364e − 02	8.659e − 02	9.459e − 02	9.270e − 02	9.917e − 02

Table 5
Average rankings of the *eMOABC* algorithm family.

Algorithm	Ranking (HV)	Ranking (SPREAD)	Ranking (EPSILON)	Ranking (IGD)
eMOABCa	1.161	4.226	5.806	5.742
eMOABCb	3.742	**3.097**	3.258	3.129
eMOABCc	**4.387**	3.161	**2.677**	**2.677**
eMOABCd	**4.290**	**3.000**	**2.645**	3.065
eMOABCe	3.806	3.839	2.968	**2.806**
eMOABCf	3.613	3.677	3.645	3.581

Each solution i in the *archive* is assigned with a crowding-distance i_{distance}, the density estimation of solution i in the objective space (for details of crowding-distance assignment, please refer to Section 2.2). Solutions in the *archive* are non-dominated within this set, and we prefer the solution that is located in a less crowded region, i.e., the solution with larger finite crowding-distance value. Notice that the boundary solutions (solutions with smallest and largest function values) with an infinite distance value are not considered here. Contrarily, the solution with the minimum crowding-distance value is deemed as the worst one, since it is located in the most crowded region.

Now try to add a new solution i into the archive, if i is dominated by or equal to any member in the archive, then it will be discarded. If some members are dominated by i, then all dominated solutions will be removed from the archive and i will be added. If i is non-dominated with any member and the archive is not full, then directly insert i into the archive. If the archive is full and i is to be added, then we first mandatorily add i into the archive and then call the function *crowdingDistanceAssignment* (*archive*) to get the new crowding distances of each member in the archive, including the new solution i to be added. After that, one of the solutions with the minimum crowding-distance value will be removed from the archive. Thus, the number of the solutions in the archive is still *AS*. In this case, the crowding distance of the new solution i is also considered, and this maintenance method can guarantee a good spread of solutions in the archive.

Table 6
The summary of the Wilcoxon test on each indicator when the *CS* is set to different values. • means that the method in the row improves and the method of the column with a significance level $\alpha = 0.1$, while ○ indicates the opposite cases.

	Indicator HV							Indicator SPREAD						
	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
CS=20(1)	–							–						
CS=30(2)		–							–					
CS=40(3)			–			•	•			–				
CS=50(4)				–							–			
CS=60(5)					–							–		
CS=70(6)			○			–							–	
CS=80(7)			○				–							–
	Indicator EPLISON							Indicator IGD						
	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
CS=20(1)	–							–						
CS=30(2)		–	○						–					
CS=40(3)		•	–		•					–				
CS=50(4)				–							–			
CS=60(5)			○		–							–		
CS=70(6)						–							–	
CS=80(7)							–							–

2.4. Send employed bees

The pseudo code of *SendEmployedBees* (*colony*, *archive*) is given in Fig. 3. For each food source x_i, its employed bee will explore a temporary position denoted as v_i. The position v_i is a copy of the food source with one randomly selected dimension *d* to be changed (Akbari et al., 2012). The updating equation for each food source is as follows:

$$v_{id} = x_{id} + \phi_{id} \cdot (x_{id} - x_{kd}) + \varphi_{id} \cdot (x_{id} - elite_d), \tag{3}$$

where ϕ_{id} and φ_{id} are two different random numbers distributed uniformly over $[-1, 1]$. k is one random integer chosen from the set $\{1, 2, \ldots, FN\}$. Although k is determined stochastically, it has to be different from *i*. And the food source x_k is called the neighbor of x_i (Karaboga & Akay, 2009). In Eq. (3), *elite* is the best solution selected from the external *archive*, that is, an intermediate solution with the maximum crowding-distance value, or one of the intermediate solutions located at the least crowded region in the archive.

As can be seen from Eq. (3), the algorithm uses three parts to generate new food sources which is analogous to the idea introduced in Xiang et al. (2014). As the difference between x_{id} and x_{kd} decreases, the perturbation on the position x_{id} gets decreased, too. Thus, as the search approaches the optimum solution in the search space, the step length is adaptively reduced. By introducing an elitism mechanism, it will attract the whole colony into a less crowded search region and this operation will enable a good exploitation around the region where *elite* is located. Note that if the value produced by Eq. (3) overflows, then the following boundary treatment method is used (see Fig. 4).

In each iteration, all employed bees share the same *elite*. After all food sources have been updated, their fitness values are computed by the function *calculateFitness* (*colony*, *archive*). In the eMOABC, the method for calculating a fitness value is the same as in SPEA2 (Zitzler et al., 2001). Given an individual *i*, the fitness value $F(i)$ contains two parts : raw fitness value $R(i)$ and density information $D(i)$, namely, $F(i) = R(i) + D(i)$. The raw fitness is determined by the strengths of its dominators in both the archive and the colony, and it could avoid the situation that individuals dominated by the same archive members have identical fitness values. Density information is incorporated to discriminate between individuals having identical raw fitness values. Details of calculating this kind of fitness value are available in Zitzler et al. (2001). It is important to note here that fitness in the eMOABC is to be minimized.

2.5. Send onlooker bees

The pseudo code of function *SendOnlookerBees* (*colony*, *archive*) is shown in Fig. 5. After all employed bees optimized their food sources, they will come to the hive to share information with the onlooker bees about the quality of corresponding food sources, and onlooker bees will then choose one food source to be exploited. For this purpose, the probability for each food source *k* advertised by the corresponding employed bee will be calculated as follows:

$$\text{prob}_k = 1 - \frac{F(x_k)}{\sum_{m=1}^{FN} F(x_m)}, \tag{4}$$

where $F(x_m)$ returns the fitness value of x_m. From Eq. (4) we can find that a food source with lower fitness value will be assigned with a higher selection probability, and this is because the fitness value used in the eMOABC is to be minimized.

Next, each onlooker bee will choose a food source *k* based on the probability provided by its employed bee by using roulette wheel method. After that, it will randomly select one dimension of that food source and try to improve it as the employed bee does. The updating equation is also given by Eq. (3). Note again that if the parameter value produced by those operations exceeds its predetermined limit, the same boundary treatment method is applied as in the employed bees phase.

2.6. Send scout bees

The pseudo code of function *SendScoutBees* (*colony*, *limit*) is given in Fig. 6. In this step, the algorithm will find abandoned food sources and replace them with new ones. A food source that cannot be improved by its employed or onlooker bee for *limit* cycles will be discarded and replaced with a vector which is generated similarly as in the initialization phase (see Eq. (2)). The scout bees phase can help the algorithm get rid of food sources that have been trapped in a local optimum. Just as the basic ABC, only one scout is allowed in each cycle in our method.

3. Simulation results

3.1. Benchmarks

In this paper, 31 benchmarks are used for assessing each algorithm. These functions come from four problem families, and they

Table 7
Mean values of indicator HV for all the algorithms.

	eMOABC	MOEA/D	SMPSO	GDE3	AbYSS	CellDE	IBEA	MOCell	OMOPSO	NSGAII	PAES	SPEA2
UF1	6.430e − 01	6.177e − 01	4.608e − 01	5.379e − 01	4.952e − 01	5.601e − 01	4.942e − 01	4.721e − 01	4.769e − 01	5.261e − 01	3.366e − 01	5.084e − 01
UF2	6.497e − 01	6.352e − 01	6.092e − 01	6.090e − 01	6.066e − 01	6.107e − 01	6.085e − 01	6.020e − 01	6.125e − 01	6.196e − 01	5.135e − 01	6.121e − 01
UF3	4.981e − 01	5.451e − 01	3.326e − 01	2.217e − 01	3.876e − 01	2.829e − 01	3.663e − 01	3.344e − 01	3.955e − 01	4.419e − 01	2.421e − 01	4.187e − 01
UF4	2.683e − 01	2.220e − 01	2.498e − 01	2.640e − 01	2.386e − 01	2.513e − 01	2.362e − 01	2.514e − 01	2.389e − 01	2.583e − 01	1.320e − 01	2.582e − 01
UF5	2.579e − 01	2.384e − 03	0.000e + 00	7.465e − 03	4.053e − 02	5.528e − 03	1.206e − 01	9.999e − 02	3.193e − 03	7.454e − 02	7.364e − 02	9.113e − 02
UF6	2.241e − 01	1.648e − 01	1.186e − 02	1.793e − 01	2.071e − 01	1.182e − 01	2.276e − 01	1.604e − 01	3.176e − 02	2.243e − 01	5.807e − 02	2.196e − 01
UF7	4.704e − 01	4.725e − 01	4.010e − 01	4.339e − 01	2.490e − 01	4.092e − 01	2.742e − 01	2.348e − 01	3.940e − 01	3.360e − 01	1.389e − 01	3.440e − 01
UF8	3.388e − 01	2.551e − 01	1.477e − 01	3.259e − 02	1.204e − 01	1.853e − 02	2.027e − 01	1.119e − 01	3.459e − 02	1.413e − 01	1.570e − 01	1.751e − 01
UF9	6.657e − 01	5.426e − 01	1.694e − 01	4.327e − 01	2.281e − 01	4.346e − 02	4.133e − 01	3.172e − 01	5.642e − 02	2.770e − 01	4.707e − 01	4.307e − 01
UF10	1.497e − 01	1.605e − 03	8.931e − 02	0.000e + 00	2.739e − 02	0.000e + 00	5.300e − 02	3.724e − 02	0.000e + 00	1.497e − 02	4.912e − 02	5.772e − 02
ZDT1	6.620e − 01	6.452e − 01	6.620e − 01	6.620e − 01	6.620e − 01	6.591e − 01	6.616e − 01	6.620e − 01	6.617e − 01	6.603e − 01	6.518e − 01	6.616e − 01
ZDT2	3.287e − 01	3.169e − 01	3.287e − 01	3.287e − 01	3.287e − 01	3.273e − 01	3.270e − 01	3.288e − 01	3.284e − 01	3.272e − 01	3.216e − 01	3.284e − 01
ZDT3	5.155e − 01	4.979e − 01	5.158e − 01	5.159e − 01	5.132e − 01	5.053e − 01	5.104e − 01	5.156e − 01	5.151e − 01	5.153e − 01	4.917e − 01	5.156e − 01
ZDT4	6.620e − 01	4.687e − 01	6.617e − 01	1.672e − 01	6.597e − 01	3.337e − 03	2.219e − 01	6.612e − 01	0.000e + 00	6.595e − 01	6.549e − 01	6.606e − 01
ZDT6	4.014e − 01	3.997e − 01	4.013e − 01	4.013e − 01	4.006e − 01	4.008e − 01	3.973e − 01	4.011e − 01	4.013e − 01	3.982e − 01	3.954e − 01	4.004e − 01
WFG1	5.997e − 01	5.124e − 01	1.307e − 01	6.129e − 01	2.917e − 01	6.069e − 01	4.835e − 01	5.104e − 01	2.024e − 01	5.408e − 01	8.501e − 02	4.998e − 01
WFG2	5.568e − 01	5.611e − 01	5.632e − 01	5.646e − 01	5.561e − 01	5.627e − 01	5.582e − 01	5.620e − 01	5.638e − 01	5.624e − 01	4.461e − 01	5.625e − 01
WFG3	4.381e − 01	4.406e − 01	4.417e − 01	4.419e − 01	4.411e − 01	4.398e − 01	4.416e − 01	4.419e − 01	4.419e − 01	4.409e − 01	4.314e − 01	4.416e − 01
WFG4	2.193e − 01	2.140e − 01	2.066e − 01	2.191e − 01	2.193e − 01	2.154e − 01	2.169e − 01	2.191e − 01	2.091e − 01	2.174e − 01	2.164e − 01	2.185e − 01
WFG5	1.972e − 01	1.934e − 01	1.963e − 01	1.961e − 01	1.964e − 01	1.955e − 01	1.952e − 01	1.963e − 01	1.963e − 01	1.948e − 01	1.945e − 01	1.956e − 01
WFG6	2.042e − 01	2.077e − 01	2.100e − 01	2.103e − 01	1.680e − 01	2.085e − 01	1.923e − 01	1.822e − 01	2.103e − 01	1.965e − 01	1.707e − 01	2.014e − 01
WFG7	2.108e − 01	2.077e − 01	2.101e − 01	2.104e − 01	2.107e − 01	2.087e − 01	2.084e − 01	2.107e − 01	2.104e − 01	2.090e − 01	2.061e − 01	2.099e − 01
WFG8	1.680e − 01	1.698e − 01	1.523e − 01	1.455e − 01	1.481e − 01	1.492e − 01	1.507e − 01	1.502e − 01	1.479e − 01	1.590e − 01	8.056e − 02	1.542e − 01
WFG9	2.372e − 01	2.361e − 01	2.368e − 01	2.384e − 01	2.379e − 01	2.339e − 01	2.376e − 01	2.389e − 01	2.377e − 01	2.375e − 01	1.428e − 01	2.390e − 01
DTLZ1	7.721e − 01	7.508e − 01	7.381e − 01	7.634e − 01	7.578e − 01	4.762e − 01	1.672e − 01	7.439e − 01	0.000e + 00	7.580e − 01	6.472e − 01	7.849e − 01
DTLZ2	3.941e − 01	3.691e − 01	3.556e − 01	3.792e − 01	3.829e − 01	3.772e − 01	4.119e − 01	3.766e − 01	3.566e − 01	3.739e − 01	2.129e − 01	4.054e − 01
DTLZ3	3.947e − 01	2.328e − 01	3.212e − 01	9.647e − 02	2.555e − 01	7.741e − 03	1.243e − 03	1.001e − 01	0.000e + 00	2.865e − 01	2.399e − 01	2.434e − 01
DTLZ4	3.298e − 01	3.493e − 01	3.640e − 01	3.754e − 01	3.864e − 01	3.386e − 01	2.375e − 01	3.041e − 01	3.631e − 01	3.741e − 01	2.056e − 02	3.088e − 01
DTLZ5	9.402e − 02	8.971e − 02	9.390e − 02	9.398e − 02	9.400e − 02	8.903e − 02	9.179e − 02	9.407e − 02	9.380e − 02	9.294e − 02	9.204e − 02	9.335e − 02
DTLZ6	9.498e − 02	9.130e − 02	9.488e − 02	9.487e − 02	5.108e − 02	9.431e − 02	6.727e − 02	0.000e + 00	9.490e − 02	8.677e − 02	9.352e − 02	8.753e − 02
DTLZ7	2.936e − 01	2.096e − 01	2.763e − 01	2.958e − 01	2.567e − 01	2.661e − 01	2.438e − 01	2.665e − 01	2.654e − 01	2.839e − 01	1.781e − 01	2.957e − 01

Table 8
Mean values of indicator SPREAD for all the algorithms.

	eMOABC	MOEA/D	SMPSO	GDE3	AbYSS	CellDE
UF1	$7.576e-01$	$9.075e-01$	$8.262e-01$	$1.443e+00$	$1.420e+00$	$1.105e+00$
UF2	$7.886e-01$	$7.428e-01$	$5.579e-01$	$6.584e-01$	$4.694e-01$	$5.910e-01$
UF3	$7.714e-01$	$9.660e-01$	$7.363e-01$	$1.216e+00$	$1.052e+00$	$9.862e-01$
UF4	$5.923e-01$	$6.939e-01$	$3.872e-01$	$4.321e-01$	$4.910e-01$	$5.354e-01$
UF5	$6.204e-01$	$1.271e+00$	$9.261e-01$	$1.862e+00$	$1.146e+00$	$1.113e+00$
UF6	$7.535e-01$	$1.244e+00$	$9.923e-01$	$1.685e+00$	$1.057e+00$	$1.106e+00$
UF7	$8.840e-01$	$7.560e-01$	$8.713e-01$	$9.485e-01$	$1.162e+00$	$7.955e-01$
UF8	$8.608e-01$	$9.252e-01$	$7.597e-01$	$6.669e-01$	$8.387e-01$	$6.286e-01$
UF9	$9.066e-01$	$9.140e-01$	$8.383e-01$	$7.326e-01$	$8.609e-01$	$6.590e-01$
UF10	$1.094e+00$	$9.421e-01$	$7.423e-01$	$5.857e-01$	$8.931e-01$	$6.279e-01$
ZDT1	$1.164e-01$	$5.639e-01$	$6.851e-02$	$1.503e-01$	$7.822e-02$	$3.051e-01$
ZDT2	$1.124e-01$	$5.280e-01$	$6.565e-02$	$1.492e-01$	$7.389e-02$	$3.068e-01$
ZDT3	$7.997e-01$	$1.087e+00$	$7.028e-01$	$7.018e-01$	$7.068e-01$	$7.482e-01$
ZDT4	$9.142e-02$	$1.095e+00$	$7.758e-02$	$6.806e-01$	$8.630e-02$	$8.296e-01$
ZDT6	$1.309e-01$	$4.982e-01$	$2.817e-01$	$1.561e-01$	$5.958e-02$	$4.438e-01$
WFG1	$8.685e-01$	$1.195e+00$	$1.032e+00$	$3.824e-01$	$5.852e-01$	$7.471e-01$
WFG2	$8.984e-01$	$1.165e+00$	$7.658e-01$	$7.567e-01$	$7.487e-01$	$7.971e-01$
WFG3	$4.013e-01$	$7.052e-01$	$3.647e-01$	$4.392e-01$	$3.685e-01$	$5.326e-01$
WFG4	$1.355e-01$	$7.016e-01$	$3.745e-01$	$1.985e-01$	$1.311e-01$	$3.591e-01$
WFG5	$1.481e-01$	$6.595e-01$	$1.362e-01$	$2.198e-01$	$1.359e-01$	$3.728e-01$
WFG6	$1.483e-01$	$6.242e-01$	$1.201e-01$	$1.749e-01$	$1.443e-01$	$3.604e-01$
WFG7	$1.179e-01$	$6.197e-01$	$1.255e-01$	$1.783e-01$	$1.045e-01$	$3.502e-01$
WFG8	$6.672e-01$	$7.767e-01$	$7.918e-01$	$6.072e-01$	$5.665e-01$	$6.385e-01$
WFG9	$1.819e-01$	$6.703e-01$	$1.993e-01$	$2.116e-01$	$1.415e-01$	$3.720e-01$
DTLZ1	$7.164e-01$	$7.995e-01$	$6.693e-01$	$7.063e-01$	$9.000e-01$	$5.996e-01$
DTLZ2	$6.357e-01$	$8.072e-01$	$6.366e-01$	$6.063e-01$	$7.684e-01$	$5.710e-01$
DTLZ3	$6.223e-01$	$8.839e-01$	$7.405e-01$	$6.273e-01$	$8.720e-01$	$6.259e-01$
DTLZ4	$9.999e-01$	$8.695e-01$	$6.346e-01$	$6.341e-01$	$6.755e-01$	$5.742e-01$
DTLZ5	$1.261e-01$	$9.132e-01$	$1.509e-01$	$1.941e-01$	$1.245e-01$	$4.941e-01$
DTLZ6	$1.109e-01$	$9.190e-01$	$1.132e-01$	$1.900e-01$	$8.682e-01$	$3.296e-01$
DTLZ7	$7.205e-01$	$9.946e-01$	$7.230e-01$	$6.908e-01$	$7.115e-01$	$6.609e-01$

	IBEA	MOCell	OMOPSO	NSGAII	PAES	SPEA2
UF1	$1.476e+00$	$1.409e+00$	$1.028e+00$	$1.495e+00$	$1.080e+00$	$1.524e+00$
UF2	$1.025e+00$	$5.353e-01$	$4.813e-01$	$6.315e-01$	$9.350e-01$	$5.950e-01$
UF3	$1.050e+00$	$1.123e+00$	$6.772e-01$	$1.180e+00$	$1.062e+00$	$1.146e+00$
UF4	$1.472e+00$	$4.229e-01$	$3.812e-01$	$4.847e-01$	$1.037e+00$	$4.089e-01$
UF5	$1.239e+00$	$1.121e+00$	$8.949e-01$	$1.301e+00$	$9.860e-01$	$1.290e+00$
UF6	$1.216e+00$	$1.208e+00$	$9.851e-01$	$1.337e+00$	$9.728e-01$	$1.356e+00$
UF7	$1.219e+00$	$1.102e+00$	$8.366e-01$	$1.076e+00$	$1.011e+00$	$1.120e+00$
UF8	$7.672e-01$	$7.475e-01$	$8.007e-01$	$7.789e-01$	$7.664e-01$	$5.800e-01$
UF9	$1.160e+00$	$7.920e-01$	$8.561e-01$	$8.638e-01$	$6.966e-01$	$6.083e-01$
UF10	$1.319e+00$	$8.320e-01$	$6.860e-01$	$7.668e-01$	$9.509e-01$	$7.936e-01$
ZDT1	$2.947e-01$	$5.900e-02$	$6.864e-02$	$3.737e-01$	$6.109e-01$	$1.495e-01$
ZDT2	$3.226e-01$	$7.606e-02$	$6.961e-02$	$3.837e-01$	$6.814e-01$	$1.419e-01$
ZDT3	$1.225e+00$	$7.040e-01$	$7.066e-01$	$7.397e-01$	$9.485e-01$	$7.047e-01$
ZDT4	$1.114e+00$	$7.528e-02$	$8.816e-01$	$4.085e-01$	$9.407e-01$	$1.159e-01$
ZDT6	$7.804e-01$	$4.715e-02$	$6.629e-01$	$6.465e-01$	$7.829e-01$	$1.415e-01$
WFG1	$8.846e-01$	$5.605e-01$	$1.165e+00$	$6.168e-01$	$1.032e+00$	$6.261e-01$
WFG2	$1.257e+00$	$7.501e-01$	$7.634e-01$	$7.882e-01$	$9.794e-01$	$7.560e-01$
WFG3	$5.103e-01$	$3.643e-01$	$3.631e-01$	$5.865e-01$	$7.764e-01$	$4.335e-01$
WFG4	$6.283e-01$	$1.428e-01$	$3.410e-01$	$3.927e-01$	$6.204e-01$	$2.754e-01$
WFG5	$6.746e-01$	$1.376e-01$	$1.313e-01$	$4.145e-01$	$6.510e-01$	$2.785e-01$
WFG6	$6.452e-01$	$1.419e-01$	$1.160e-01$	$3.842e-01$	$6.508e-01$	$2.469e-01$
WFG7	$6.373e-01$	$1.179e-01$	$1.202e-01$	$3.821e-01$	$6.229e-01$	$2.460e-01$
WFG8	$1.376e+00$	$5.687e-01$	$5.407e-01$	$6.442e-01$	$1.032e+00$	$6.010e-01$
WFG9	$5.375e-01$	$1.435e-01$	$1.883e-01$	$4.086e-01$	$6.428e-01$	$2.950e-01$
DTLZ1	$1.615e+00$	$8.901e-01$	$6.785e-01$	$8.633e-01$	$8.495e-01$	$7.293e-01$
DTLZ2	$5.785e-01$	$6.920e-01$	$6.228e-01$	$6.993e-01$	$6.697e-01$	$5.325e-01$
DTLZ3	$1.677e+00$	$9.485e-01$	$7.419e-01$	$7.819e-01$	$9.604e-01$	$8.076e-01$
DTLZ4	$7.168e-01$	$7.287e-01$	$6.126e-01$	$7.016e-01$	$9.611e-01$	$5.337e-01$
DTLZ5	$6.760e-01$	$1.398e-01$	$1.891e-01$	$4.312e-01$	$6.376e-01$	$2.044e-01$
DTLZ6	$1.158e+00$	$8.241e-01$	$1.073e-01$	$6.160e-01$	$8.332e-01$	$6.338e-01$
DTLZ7	$8.135e-01$	$7.044e-01$	$7.007e-01$	$7.468e-01$	$7.729e-01$	$5.818e-01$

Table 9
Mean values of indicator EPSILON for all the algorithms.

	eMOABC	MOEA/D	SMPSO	GDE3	AbYSS	CellDE
UF1	4.387e − 02	8.631e − 02	1.842e − 01	1.879e − 01	2.378e − 01	1.477e − 01
UF2	4.746e − 02	9.911e − 02	1.204e − 01	1.185e − 01	1.681e − 01	9.755e − 02
UF3	2.366e − 01	1.874e − 01	2.477e − 01	3.931e − 01	4.203e − 01	2.987e − 01
UF4	5.669e − 02	9.253e − 02	6.984e − 02	6.173e − 02	8.283e − 02	6.540e − 02
UF5	2.302e − 01	9.571e − 01	1.853e + 00	4.925e − 01	7.373e − 01	9.501e − 01
UF6	2.794e − 01	5.227e − 01	7.502e − 01	3.407e − 01	5.809e − 01	4.840e − 01
UF7	6.577e − 02	7.310e − 02	1.574e − 01	1.609e − 01	6.245e − 01	1.742e − 01
UF8	2.222e − 01	2.640e − 01	7.101e − 01	7.556e − 01	6.929e − 01	7.861e − 01
UF9	1.945e − 01	3.711e − 01	6.780e − 01	2.720e − 01	6.039e − 01	7.723e − 01
UF10	6.110e − 01	1.063e + 00	7.303e − 01	1.510e + 00	9.855e − 01	2.072e + 00
ZDT1	6.177e − 03	2.103e − 02	5.305e − 03	6.860e − 03	5.980e − 03	1.229e − 02
ZDT2	6.033e − 03	2.290e − 02	5.251e − 03	6.870e − 03	5.690e − 03	1.182e − 02
ZDT3	7.807e − 03	3.684e − 02	5.053e − 03	5.474e − 03	1.096e − 01	2.300e − 02
ZDT4	5.734e − 03	2.198e − 01	5.648e − 03	6.012e − 01	7.742e − 03	4.510e + 00
ZDT6	5.343e − 03	9.664e − 03	4.690e − 03	5.710e − 03	4.780e − 03	9.439e − 03
WFG1	2.124e − 01	3.791e − 01	1.090e + 00	1.154e − 01	7.943e − 01	1.754e − 01
WFG2	4.448e − 02	5.618e − 02	1.081e − 02	9.991e − 03	6.258e − 01	1.837e − 02
WFG3	2.010e + 00	2.000e + 00	2.001e + 00	2.000e + 00	2.003e + 00	2.002e + 00
WFG4	1.403e − 02	4.421e − 02	4.251e − 02	1.877e − 02	1.423e − 02	3.648e − 02
WFG5	6.497e − 02	8.506e − 02	6.334e − 02	6.807e − 02	6.339e − 02	7.959e − 02
WFG6	2.545e − 02	3.272e − 02	1.476e − 02	1.741e − 02	8.611e − 02	3.094e − 02
WFG7	1.427e − 02	3.625e − 02	1.556e − 02	1.951e − 02	1.472e − 02	3.413e − 02
WFG8	3.054e − 01	3.829e − 01	3.292e − 01	4.960e − 01	4.584e − 01	4.319e − 01
WFG9	2.295e − 02	4.395e − 02	2.567e − 02	2.725e − 02	2.021e − 02	4.334e − 02
DTLZ1	4.751e − 02	4.992e − 02	5.746e − 02	4.811e − 02	5.680e − 02	1.455e − 01
DTLZ2	1.151e − 01	1.399e − 01	1.366e − 01	1.153e − 01	1.263e − 01	1.024e − 01
DTLZ3	1.129e − 01	1.117e + 00	2.098e − 01	2.178e + 00	4.144e − 01	5.769e + 00
DTLZ4	1.811e − 01	1.493e − 01	1.152e − 01	1.091e − 01	1.011e − 01	1.125e − 01
DTLZ5	4.752e − 03	2.589e − 02	4.333e − 03	5.381e − 03	4.831e − 03	1.526e − 02
DTLZ6	4.245e − 03	2.638e − 02	4.185e − 03	5.264e − 03	7.940e − 02	8.120e − 03
DTLZ7	1.863e − 01	3.496e − 01	1.655e − 01	1.235e − 01	1.277e + 00	3.749e − 01
	IBEA	MOCell	OMOPSO	NSGAII	PAES	SPEA2
UF1	2.467e − 01	2.827e − 01	2.120e − 01	1.953e − 01	5.606e − 01	2.238e − 01
UF2	1.692e − 01	1.803e − 01	9.307e − 02	1.146e − 01	3.492e − 01	1.348e − 01
UF3	4.756e − 01	5.360e − 01	2.246e − 01	3.364e − 01	6.441e − 01	4.074e − 01
UF4	1.089e − 01	6.898e − 02	7.759e − 02	5.724e − 02	5.011e − 01	6.424e − 02
UF5	6.465e − 01	7.835e − 01	1.243e + 00	5.687e − 01	8.735e − 01	5.686e − 01
UF6	5.363e − 01	6.504e − 01	6.304e − 01	4.272e − 01	1.009e + 00	4.336e − 01
UF7	5.502e − 01	6.194e − 01	1.799e − 01	4.178e − 01	8.016e − 01	3.814e − 01
UF8	6.988e − 01	6.941e − 01	7.493e − 01	5.728e − 01	7.587e − 01	6.940e − 01
UF9	4.919e − 01	5.026e − 01	8.122e − 01	5.276e − 01	4.833e − 01	4.547e − 01
UF10	9.010e − 01	8.926e − 01	1.839e + 00	9.021e − 01	8.403e − 01	8.635e − 01
ZDT1	8.987e − 03	5.388e − 03	5.552e − 03	1.320e − 02	3.676e − 02	7.936e − 03
ZDT2	1.618e − 02	5.320e − 03	5.463e − 03	1.265e − 02	9.273e − 02	7.926e − 03
ZDT3	2.359e − 02	2.588e − 02	5.826e − 03	8.593e − 03	2.211e − 01	8.775e − 03
ZDT4	7.663e − 01	6.077e − 03	5.213e + 00	1.283e − 02	2.362e − 02	7.865e − 03
ZDT6	1.098e − 02	4.586e − 03	4.933e − 03	1.256e − 02	2.171e − 02	6.195e − 03
WFG1	2.874e − 01	4.286e − 01	1.020e + 00	2.425e − 01	1.386e + 00	4.650e − 01
WFG2	5.765e − 01	5.469e − 01	9.928e − 03	3.632e − 01	1.656e + 00	3.635e − 01
WFG3	2.001e + 00	2.001e + 00	2.000e + 00	2.001e + 00	2.130e + 00	2.001e + 00
WFG4	4.543e − 02	1.429e − 02	3.748e − 02	3.391e − 02	1.006e − 01	2.341e − 02
WFG5	8.623e − 02	6.335e − 02	6.351e − 02	8.474e − 02	1.414e − 01	7.353e − 02
WFG6	6.885e − 02	6.071e − 02	1.423e − 02	5.109e − 02	2.494e − 01	3.454e − 02
WFG7	4.118e − 02	1.411e − 02	1.496e − 02	3.667e − 02	1.478e − 01	2.574e − 02
WFG8	4.633e − 01	4.707e − 01	4.959e − 01	3.850e − 01	2.238e + 00	4.117e − 01
WFG9	3.534e − 02	1.769e − 02	2.367e − 02	3.927e − 02	3.862e − 01	2.849e − 02
DTLZ1	2.945e − 01	6.173e − 02	9.131e + 00	5.712e − 02	1.369e − 01	3.424e − 02
DTLZ2	8.977e − 02	1.270e − 01	1.338e − 01	1.324e − 01	6.338e − 01	8.066e − 02
DTLZ3	4.259e − 01	7.954e − 01	9.304e + 01	3.049e − 01	5.036e − 01	3.855e − 01
DTLZ4	4.983e − 01	2.801e − 01	1.096e − 01	1.063e − 01	9.429e − 01	3.203e − 01
DTLZ5	3.145e − 02	4.777e − 03	4.697e − 03	1.039e − 02	2.070e − 02	7.170e − 03
DTLZ6	6.882e − 02	6.235e − 01	4.326e − 03	1.766e − 02	2.831e − 02	1.632e − 02
DTLZ7	1.066e + 00	8.037e − 01	1.839e − 01	1.370e − 01	2.650e + 00	1.324e − 01

Table 10
Mean values of indicator IGD for all the algorithms.

	eMOABC	MOEA/D	SMPSO	GDE3	AbYSS	CellDE
UF1	1.480e − 02	3.114e − 02	1.264e − 01	8.602e − 02	1.340e − 01	6.246e − 02
UF2	1.407e − 02	2.478e − 02	4.668e − 02	4.306e − 02	6.518e − 02	3.854e − 02
UF3	1.075e − 01	7.855e − 02	2.260e − 01	3.526e − 01	2.800e − 01	2.657e − 01
UF4	4.123e − 02	7.966e − 02	5.498e − 02	4.709e − 02	6.269e − 02	5.516e − 02
UF5	1.625e − 01	7.612e − 01	1.861e + 00	3.319e − 01	5.071e − 01	7.720e − 01
UF6	2.174e − 01	2.560e − 01	6.853e − 01	1.567e − 01	4.168e − 01	3.165e − 01
UF7	1.872e − 02	1.528e − 02	6.198e − 02	4.118e − 02	3.258e − 01	5.674e − 02
UF8	1.218e − 01	1.150e − 01	2.264e − 01	3.323e − 01	2.807e − 01	4.054e − 01
UF9	6.641e − 02	1.576e − 01	3.976e − 01	2.016e − 01	3.812e − 01	6.578e − 01
UF10	2.950e − 01	8.738e − 01	2.929e − 01	1.552e + 00	6.696e − 01	2.589e + 00
ZDT1	3.737e − 03	1.258e − 02	3.672e − 03	3.777e − 03	3.725e − 03	4.831e − 03
ZDT2	3.850e − 03	9.135e − 03	3.796e − 03	3.913e − 03	3.826e − 03	4.367e − 03
ZDT3	5.164e − 03	1.722e − 02	4.285e − 03	4.369e − 03	1.500e − 02	1.020e − 02
ZDT4	3.710e − 03	1.433e − 01	3.717e − 03	4.722e − 01	4.412e − 03	4.240e + 00
ZDT6	3.025e − 03	4.167e − 03	3.003e − 03	3.124e − 03	3.058e − 03	3.436e − 03
WFG1	1.018e − 01	3.218e − 01	1.109e + 00	5.070e − 02	7.324e − 01	8.736e − 02
WFG2	2.136e − 02	4.976e − 02	1.071e − 02	1.008e − 02	6.210e − 02	1.143e − 02
WFG3	1.462e − 01	1.435e − 01	1.385e − 01	1.386e − 01	1.394e − 01	1.427e − 01
WFG4	1.037e − 02	2.224e − 02	2.699e − 02	1.087e − 02	1.046e − 02	1.611e − 02
WFG5	6.461e − 02	6.821e − 02	6.631e − 02	6.643e − 02	6.595e − 02	6.649e − 02
WFG6	1.915e − 02	1.903e − 02	1.287e − 02	1.304e − 02	9.322e − 02	1.450e − 02
WFG7	1.182e − 02	2.002e − 02	1.195e − 02	1.242e − 02	1.191e − 02	1.438e − 02
WFG8	8.937e − 02	8.300e − 02	1.031e − 01	1.360e − 01	1.304e − 01	1.215e − 01
WFG9	1.504e − 02	2.017e − 02	1.557e − 02	1.451e − 02	1.452e − 02	2.031e − 02
DTLZ1	2.258e − 02	2.543e − 02	2.828e − 02	2.339e − 02	2.733e − 02	1.605e − 01
DTLZ2	6.050e − 02	6.714e − 02	7.178e − 02	6.288e − 02	6.888e − 02	6.611e − 02
DTLZ3	6.253e − 02	1.179e + 00	1.155e − 01	2.259e + 00	3.940e − 01	8.518e + 00
DTLZ4	6.262e − 02	5.490e − 02	6.805e − 02	6.571e − 02	6.053e − 02	7.711e − 02
DTLZ5	4.017e − 03	1.040e − 02	4.009e − 03	4.199e − 03	4.088e − 03	8.561e − 03
DTLZ6	3.934e − 03	9.369e − 03	3.937e − 03	4.153e − 03	7.892e − 02	4.542e − 03
DTLZ7	8.659e − 02	1.902e − 01	8.520e − 02	7.477e − 02	3.946e − 01	1.234e − 01
	IBEA	MOCell	OMOPSO	NSGAII	PAES	SPEA2
UF1	1.460e − 01	1.649e − 01	1.086e − 01	1.038e − 01	3.647e − 01	1.248e − 01
UF2	7.514e − 02	6.823e − 02	4.057e − 02	3.858e − 02	1.792e − 01	4.522e − 02
UF3	2.901e − 01	2.832e − 01	1.784e − 01	1.581e − 01	3.450e − 01	1.942e − 01
UF4	6.839e − 02	5.691e − 02	6.437e − 02	5.123e − 02	2.181e − 01	5.205e − 02
UF5	3.995e − 01	4.596e − 01	1.195e + 00	3.801e − 01	5.296e − 01	3.802e − 01
UF6	3.770e − 01	4.300e − 01	5.341e − 01	2.580e − 01	7.262e − 01	2.749e − 01
UF7	2.848e − 01	3.455e − 01	6.487e − 02	1.854e − 01	5.057e − 01	1.699e − 01
UF8	4.436e − 01	2.578e − 01	3.667e − 01	2.213e − 01	4.133e − 01	1.995e − 01
UF9	2.836e − 01	2.872e − 01	5.596e − 01	3.159e − 01	2.717e − 01	2.040e − 01
UF10	6.013e − 01	4.437e − 01	2.201e + 00	3.855e − 01	5.101e − 01	3.248e − 01
ZDT1	4.109e − 03	3.688e − 03	3.711e − 03	4.838e − 03	1.175e − 02	3.921e − 03
ZDT2	9.413e − 03	3.796e − 03	3.832e − 03	4.898e − 03	1.463e − 02	3.895e − 03
ZDT3	2.974e − 02	6.176e − 03	4.358e − 03	5.389e − 03	5.611e − 02	4.848e − 03
ZDT4	6.263e − 01	3.843e − 03	4.926e + 00	4.933e − 03	7.346e − 03	4.075e − 03
ZDT6	5.167e − 03	3.000e − 03	3.019e − 03	4.765e − 03	7.074e − 03	3.172e − 03
WFG1	2.892e − 01	3.462e − 01	8.366e − 01	1.962e − 01	1.252e + 00	3.771e − 01
WFG2	9.849e − 02	4.932e − 02	1.030e − 02	3.755e − 02	3.066e − 01	3.586e − 02
WFG3	1.398e − 01	1.384e − 01	1.384e − 01	1.418e − 01	1.674e − 01	1.390e − 01
WFG4	2.026e − 02	1.043e − 02	2.305e − 02	1.368e − 02	1.550e − 02	1.272e − 02
WFG5	7.281e − 02	6.628e − 02	6.628e − 02	6.812e − 02	6.975e − 02	6.672e − 02
WFG6	5.399e − 02	6.321e − 02	1.269e − 02	3.494e − 02	9.740e − 02	2.316e − 02
WFG7	1.555e − 02	1.172e − 02	1.178e − 02	1.620e − 02	1.959e − 02	1.295e − 02
WFG8	1.322e − 01	1.175e − 01	1.269e − 01	9.909e − 02	6.957e − 01	1.059e − 01
WFG9	2.062e − 02	1.350e − 02	1.454e − 02	1.744e − 02	3.070e − 01	1.358e − 02
DTLZ1	1.818e − 01	2.869e − 02	1.184e + 01	2.612e − 02	5.860e − 02	2.021e − 02
DTLZ2	1.222e − 01	6.681e − 02	6.886e − 02	6.887e − 02	3.151e − 01	5.420e − 02
DTLZ3	5.116e − 01	7.556e − 01	1.152e + 02	2.937e − 01	1.919e − 01	3.387e − 01
DTLZ4	2.107e − 01	1.356e − 01	6.487e − 02	6.390e − 02	3.991e − 01	1.379e − 01
DTLZ5	1.939e − 02	4.053e − 03	4.134e − 03	5.420e − 03	6.835e − 03	4.332e − 03
DTLZ6	5.754e − 02	7.551e − 01	3.899e − 03	1.357e − 02	7.135e − 03	1.254e − 02
DTLZ7	3.998e − 01	2.451e − 01	8.687e − 02	7.647e − 02	8.878e − 01	6.961e − 02

Table 11
Average rankings of the algorithms on each indicator.

Algorithm	Ranking(HV)	Ranking(SPREAD)	Ranking(EPSILON)	Ranking(IGD)
eMOABC	**10.129**	5.484	**3.032**	**2.903**
MOEA/D	5.645	9.677	7.065	6.710
SMPSO	6.871	**4.323**	**4.774**	5.290
GDE3	**7.968**	5.548	4.903	**5.000**
AbYSS	6.581	5.226	6.774	6.935
CellDE	4.968	5.613	7.194	7.452
IBEA	5.323	10.194	8.613	9.484
MOCell	7.000	4.774	6.613	6.129
OMOPSO	5.805	**4.000**	6.226	6.387
NSGAII	6.839	8.258	6.258	6.161
PAES	3.032	9.226	10.742	10.290
SPEA2	7.839	5.677	5.806	5.258

Table 12
Results obtained by the Wilcoxon test for the algorithm eMOABC. • indicates that eMOABC significantly outperforms other algorithms, and ⋆ means no obvious differences are detected in the pairwise comparison.

eMOABC VS	Indicator HV				Indicator SPREAD			
	R^+	R^-	p-value	Sig. sign	R^+	R^-	p-value	Sig. sign
MOEA/D	428.0	68.0	0.000404	•($\alpha = 0.01$)	469.0	27.0	0.000014	•($\alpha = 0.01$)
SMPSO	433.0	63.0	0.000259	•($\alpha = 0.01$)	216.0	280.0	1	⋆
GDE3	384.5	111.5	0.007259	•($\alpha = 0.01$)	271.0	225.0	0.645144	⋆
AbYSS	425.5	39.5	0.000069	•($\alpha = 0.01$)	223.0	273.0	1	⋆
CellDE	442.0	54.0	0.000138	•($\alpha = 0.01$)	326.0	170.0	0.123968	⋆
IBEA	458.0	38.0	0.000034	•($\alpha = 0.01$)	478.0	18.0	0.000006	•($\alpha = 0.01$)
MOCell	428.0	37.0	0.000046	•($\alpha = 0.01$)	220.0	245.0	1	⋆
OMOPSO	438.0	58.0	0.000189	•($\alpha = 0.01$)	217.0	279.0	1	⋆
NSGAII	445.0	51.0	0.000109	•($\alpha = 0.01$)	394.0	102.0	0.004093	•($\alpha = 0.01$)
PAES	496.0	0.0	0.000001	•($\alpha = 0.01$)	469.0	27.0	0.000014	•($\alpha = 0.01$)
SPEA2	409.0	87.0	0.001552	•($\alpha = 0.01$)	278.0	218.0	0.550044	⋆
eMOABC VS	Indicator EPSILON				Indicator IGD			
	R^+	R^-	p-value	Sig. sign	R^+	R^-	p-value	Sig. sign
MOEA/D	453.0	43.0	0.000056	•($\alpha = 0.01$)	428.0	68.0	0.000404	•($\alpha = 0.01$)
SMPSO	376.0	120.0	0.011797	•($\alpha = 0.05$)	399.0	97.0	0.002989	•($\alpha = 0.01$)
GDE3	382.0	114.0	0.008396	•($\alpha = 0.01$)	369.0	127.0	0.017267	•($\alpha = 0.05$)
AbYSS	442.0	54.0	0.000133	•($\alpha = 0.01$)	464.0	32.0	0.000021	•($\alpha = 0.01$)
CellDE	431.0	65.0	0.000323	•($\alpha = 0.01$)	450.0	46.0	0.000072	•($\alpha = 0.01$)
IBEA	485.0	11.0	0.000003	•($\alpha = 0.01$)	490.0	6.0	0.000002	•($\alpha = 0.01$)
MOCell	449.0	47.0	0.000079	•($\alpha = 0.01$)	460.0	36.0	0.000031	•($\alpha = 0.01$)
OMOPSO	384.0	112.0	0.007475	•($\alpha = 0.01$)	424.0	72.0	0.000543	•($\alpha = 0.01$)
NSGAII	450.0	46.0	0.000072	•($\alpha = 0.01$)	465.0	31.0	0.00002	•($\alpha = 0.01$)
PAES	496.0	0.0	0.000001	•($\alpha = 0.01$)	496.0	0.0	0.000001	•($\alpha = 0.01$)
SPEA2	438.0	58.0	0.000189	•($\alpha = 0.01$)	426.0	70.0	0.000469	•($\alpha = 0.01$)

are (1) the CEC2009 competition benchmarks (Zhang et al., 2009b) (UF1–UF10), (2) ZDT benchmarks (Zitzler et al., 2000) (ZDT1–ZDT4, ZDT6), (3) WFG benchmarks (Huband et al., 2006) (WFG1–WFG9) and (4) DTLZ benchmarks (Deb et al., 2002b) (DTLZ1–DTLZ7). Of them, 10 functions (UF8–UF10, and DTLZ1–DTLZ7) are three-objective problems, and the remaining are two-objective ones.

3.2. Performance indicators

Let P^* be a set of uniformly distributed points along the true PF (in the objective space), and A be an approximate set to the PF. The following indicators are defined based on the above notations.

3.2.1. HV (Durillo & Nebro, 2011; Zitzler & Thiele, 1999)

HV calculates the volume, in the objective space, covered by members of the approximate set A. Mathematically, for each solution $i \in A$, a hypercube v_i is constructed with a reference point W and the solution i as the diagonal corners of the hypercube. The reference point can simply be found by constructing a vector of worst objective function values. Thereafter, a union of all hypercubes is found and its HV is calculated:

$$\mathrm{HV} = \mathrm{volume}\left(\bigcup_{i=1}^{|A|} v_i\right). \tag{5}$$

Algorithms with larger values of HV are desirable.

3.2.2. SPREAD or Δ (Akay, 2013; Durillo & Nebro, 2011)

This indicator measures the extent of spread by the set of computed solutions. It is defined as:

$$\Delta = \frac{\sum_{i=1}^{m} d(e_i, A) + \sum_{x \in A} |d(x, A) - \bar{d}|}{\sum_{i=1}^{m} d(e_i, A) + |A| \cdot \bar{d}}, \tag{6}$$

where $e_1, \ldots, e_m$ are m extreme points in P^* ; $d(e_i, A)$ is the minimum Euclidean distance between e_i and the points in A; m is the number of objectives and

$$d(x, A) = \min_{y \in A, y \neq x} ||x - y||^2,$$

$$\bar{d} = \frac{1}{|A|} \sum_{x \in A} d(x, A).$$

Table 13
Mean values of indicator RUNTIME for all the algorithms.

	eMOABC	MOEAD	SMPSO	GDE3	AbYSS	CellDE
UF1	1.563e + 03	2.374e + 03	6.073e + 02	1.177e + 03	2.192e + 03	1.818e + 03
UF2	3.864e + 03	2.451e + 03	9.771e + 02	1.532e + 03	2.479e + 03	4.497e + 03
UF3	1.246e + 03	2.905e + 03	1.158e + 03	1.824e + 03	2.631e + 03	1.588e + 03
UF4	2.758e + 03	2.454e + 03	9.355e + 02	1.506e + 03	2.488e + 03	2.559e + 03
UF5	6.281e + 02	2.447e + 03	7.927e + 02	1.424e + 03	1.869e + 03	9.687e + 02
UF6	6.365e + 02	2.529e + 03	7.813e + 02	1.467e + 03	1.908e + 03	1.019e + 03
UF7	2.247e + 03	2.297e + 03	6.759e + 02	1.676e + 03	2.191e + 03	6.196e + 03
UF8	9.152e + 03	2.984e + 03	1.709e + 03	3.454e + 03	2.912e + 03	1.335e + 04
UF9	8.245e + 03	2.988e + 03	1.128e + 03	1.582e + 03	2.615e + 03	1.136e + 04
UF10	2.909e + 03	3.136e + 03	1.085e + 03	1.868e + 03	2.531e + 03	2.423e + 03
ZDT1	9.194e + 03	2.223e + 03	2.117e + 03	2.196e + 03	2.633e + 03	2.911e + 04
ZDT2	9.080e + 03	2.175e + 03	2.236e + 03	2.246e + 03	2.614e + 03	3.786e + 04
ZDT3	7.009e + 03	2.243e + 03	9.021e + 02	1.446e + 03	2.382e + 03	1.061e + 04
ZDT4	7.602e + 03	1.742e + 03	9.989e + 02	8.821e + 02	1.681e + 03	2.972e + 04
ZDT6	1.066e + 04	1.705e + 03	1.637e + 03	5.967e + 03	1.728e + 03	8.096e + 04
WFG1	1.049e + 04	1.827e + 03	4.365e + 02	4.092e + 03	2.142e + 03	1.563e + 04
WFG2	7.548e + 03	1.710e + 03	3.959e + 02	1.662e + 03	1.512e + 03	1.792e + 04
WFG3	8.657e + 03	1.695e + 03	1.266e + 03	3.704e + 03	2.231e + 03	5.911e + 04
WFG4	1.071e + 04	1.739e + 03	3.801e + 02	2.349e + 03	2.299e + 03	6.488e + 03
WFG5	1.142e + 04	1.678e + 03	1.267e + 03	5.672e + 03	2.474e + 03	1.050e + 05
WFG6	8.599e + 03	1.679e + 03	1.234e + 03	3.582e + 03	2.144e + 03	5.964e + 04
WFG7	1.038e + 04	1.757e + 03	1.318e + 03	4.161e + 03	2.405e + 03	6.762e + 04
WFG8	8.533e + 03	1.803e + 03	4.251e + 02	1.290e + 03	9.573e + 02	2.140e + 03
WFG9	9.409e + 03	1.892e + 03	8.447e + 02	2.446e + 03	2.201e + 03	3.279e + 04
DTLZ1	1.194e + 04	2.385e + 03	1.207e + 03	4.961e + 03	2.580e + 03	5.951e + 04
DTLZ2	1.415e + 04	2.494e + 03	2.465e + 03	5.796e + 03	3.532e + 03	7.007e + 04
DTLZ3	8.075e + 03	2.519e + 03	1.053e + 03	9.782e + 02	2.029e + 03	1.702e + 04
DTLZ4	1.203e + 04	2.556e + 03	1.996e + 03	5.647e + 03	3.327e + 03	6.527e + 04
DTLZ5	1.267e + 04	2.458e + 03	1.829e + 03	3.850e + 03	2.804e + 03	3.943e + 04
DTLZ6	1.235e + 04	2.451e + 03	2.176e + 03	7.158e + 03	2.351e + 03	6.107e + 04
DTLZ7	1.345e + 04	2.715e + 03	1.568e + 03	3.098e + 03	3.187e + 03	3.293e + 04
	IBEA	MOCell	OMOPSO	NSGAII	PAES	SPEA2
UF1	1.643e + 04	2.779e + 03	5.291e + 02	2.666e + 03	4.271e + 02	1.000e + 04
UF2	1.678e + 04	3.471e + 03	9.219e + 02	2.749e + 03	7.573e + 02	1.279e + 04
UF3	1.705e + 04	3.011e + 03	1.100e + 03	3.247e + 03	1.036e + 03	8.639e + 03
UF4	1.676e + 04	3.274e + 03	9.156e + 02	2.692e + 03	7.220e + 02	1.212e + 04
UF5	1.610e + 04	2.498e + 03	7.001e + 02	2.868e + 03	5.155e + 02	8.039e + 03
UF6	1.626e + 04	2.460e + 03	7.020e + 02	2.866e + 03	5.405e + 02	8.072e + 03
UF7	1.629e + 04	2.860e + 03	5.896e + 02	2.651e + 03	4.218e + 02	1.248e + 04
UF8	1.824e + 04	5.492e + 03	1.671e + 03	2.686e + 03	1.691e + 03	2.433e + 04
UF9	1.842e + 04	3.991e + 03	1.470e + 03	2.731e + 03	2.601e + 03	1.410e + 04
UF10	1.828e + 04	3.319e + 03	8.228e + 02	2.898e + 03	8.009e + 02	1.133e + 04
ZDT1	1.609e + 04	3.754e + 03	1.753e + 03	2.342e + 03	4.927e + 02	1.612e + 04
ZDT2	1.616e + 04	3.676e + 03	1.612e + 03	2.366e + 03	4.792e + 02	1.512e + 04
ZDT3	1.623e + 04	3.646e + 03	7.773e + 02	2.371e + 03	4.283e + 02	1.546e + 04
ZDT4	1.485e + 04	1.838e + 03	7.073e + 02	1.981e + 03	3.427e + 02	1.098e + 04
ZDT6	1.495e + 04	2.090e + 03	9.064e + 02	1.864e + 03	4.396e + 02	1.233e + 04
WFG1	1.509e + 04	2.273e + 03	4.022e + 02	1.865e + 03	5.791e + 02	1.764e + 04
WFG2	1.502e + 04	1.994e + 03	4.687e + 02	1.754e + 03	4.521e + 02	1.390e + 04
WFG3	1.490e + 04	2.531e + 03	1.617e + 03	1.699e + 03	6.447e + 02	1.876e + 04
WFG4	1.501e + 04	2.517e + 03	3.811e + 02	1.753e + 03	7.334e + 02	1.832e + 04
WFG5	1.491e + 04	3.106e + 03	1.384e + 03	1.709e + 03	9.322e + 02	2.670e + 04
WFG6	1.498e + 04	2.361e + 03	1.384e + 03	1.718e + 03	6.823e + 02	1.767e + 04
WFG7	1.495e + 04	2.750e + 03	1.552e + 03	1.751e + 03	8.147e + 02	2.023e + 04
WFG8	1.515e + 04	1.571e + 03	4.615e + 02	1.863e + 03	3.865e + 02	1.018e + 04
WFG9	1.511e + 04	2.717e + 03	1.115e + 03	1.905e + 03	1.266e + 03	1.945e + 04
DTLZ1	1.626e + 04	2.157e + 03	2.479e + 02	1.996e + 03	2.046e + 03	1.518e + 04
DTLZ2	1.701e + 04	5.122e + 03	6.485e + 03	1.994e + 03	3.723e + 03	2.825e + 04
DTLZ3	1.673e + 04	1.872e + 03	3.313e + 02	2.120e + 03	1.261e + 03	9.477e + 03
DTLZ4	1.750e + 04	4.284e + 03	3.709e + 03	2.099e + 03	3.530e + 02	2.421e + 04
DTLZ5	1.666e + 04	3.457e + 03	1.826e + 03	1.905e + 03	1.455e + 03	1.982e + 04
DTLZ6	1.669e + 04	3.179e + 03	1.384e + 03	2.127e + 03	1.080e + 03	1.604e + 04
DTLZ7	1.750e + 04	4.901e + 03	2.058e + 03	2.265e + 03	2.411e + 03	2.353e + 04

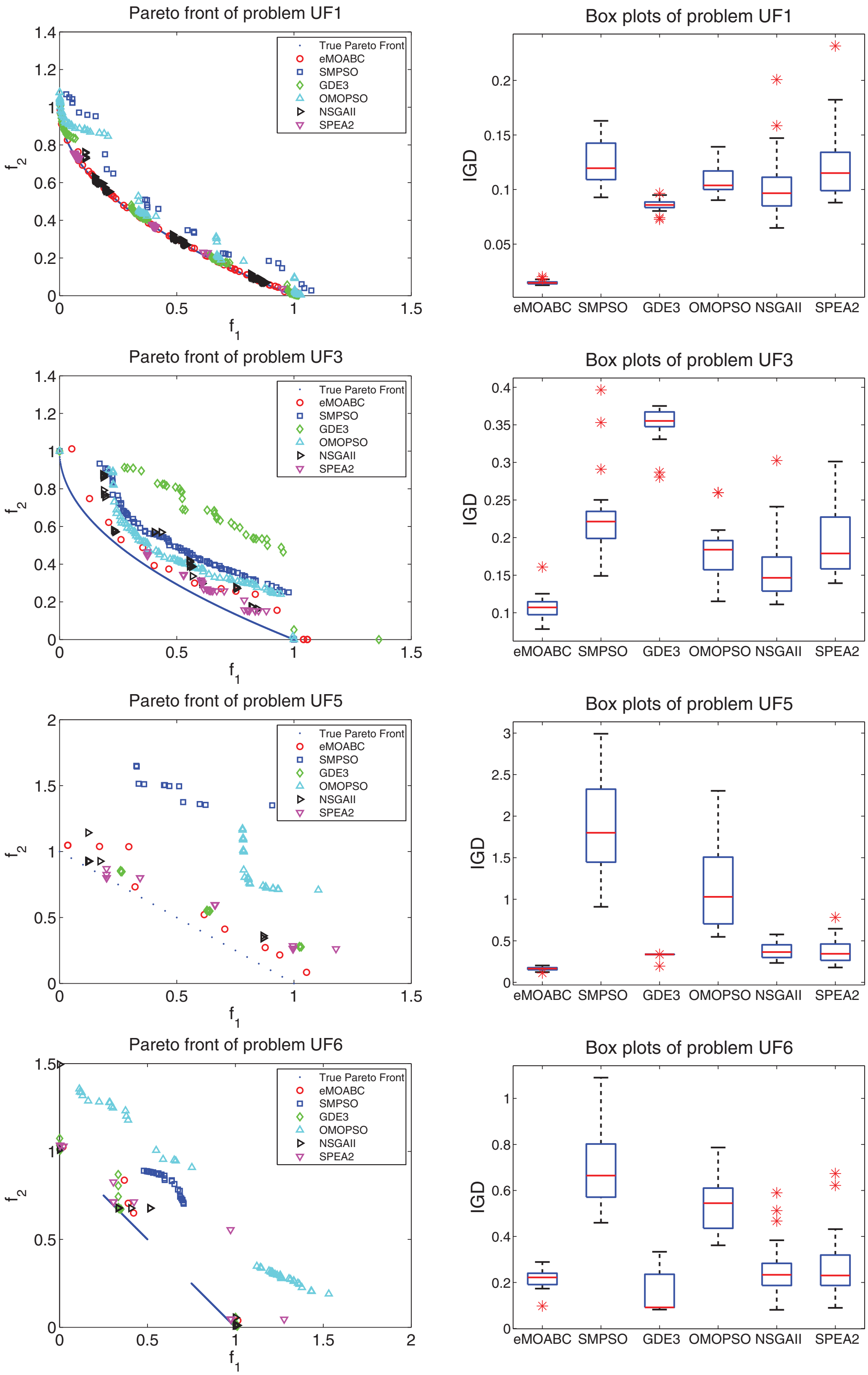

Fig. 7. Pareto fronts computed by the algorithms on UF1, UF3,UF5 and UF6.

3.2.3. EPSILON (Durillo & Nebro, 2011)

Given a computed front for a problem, A, this indicator is a measure of the smallest distance one would need to translate every solution in A so that it dominates the optimal Pareto front of this problem. More formally, given $\vec{a} = (a_1, \ldots, a_m)$ and $\vec{p} = (p_1, \ldots, p_m)$, where m is the number of objectives. Suppose ϵ is a small positive real number, then the EPSILON value of A is defined as below.

$$E(A, P^*) = \inf_{\epsilon \in \mathbb{R}^+}\{\forall \vec{p} \in P^*, \exists \vec{a} \in A : \vec{a} \prec_\epsilon \vec{p}\}, \tag{7}$$

where $\vec{a} \prec_\epsilon \vec{p}$ if and only if $\forall 1 \le i \le m : a_i < \epsilon + p_i$.

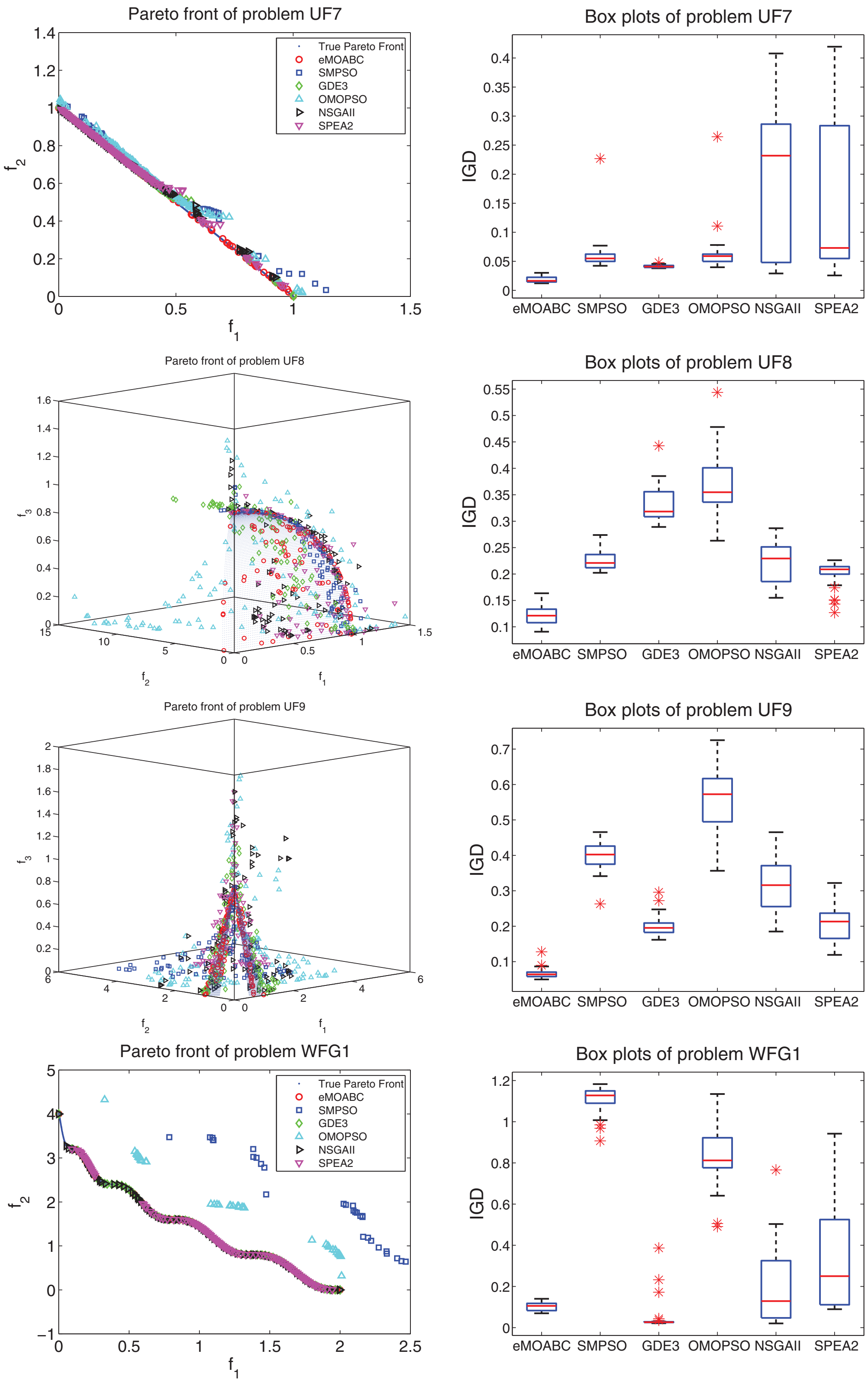

Fig. 8. Pareto fronts obtained by the algorithms on UF7, UF8, UF9 and WFG1.

3.2.4. IGD (Zhang et al., 2009b)

The indicator IGD is used to measure the average distance from P^* to A and it is defined using the following equation:

$$\mathrm{IGD}(A, P^*) = \frac{\sum_{v \in P^*} d(v, A)}{|P^*|}, \tag{8}$$

where $d(v, A)$ is the minimum Euclidean distance between v and the points in A. If $|P^*|$ is large enough to represent the Pareto front very well, both the diversity and convergence of the approximated set A could be measured using $\mathrm{IGD}(A, P^*)$. To have a small value of $\mathrm{IGD}(A, P^*)$, the set A must be very close to the true PF and cannot miss any part of the whole PF.

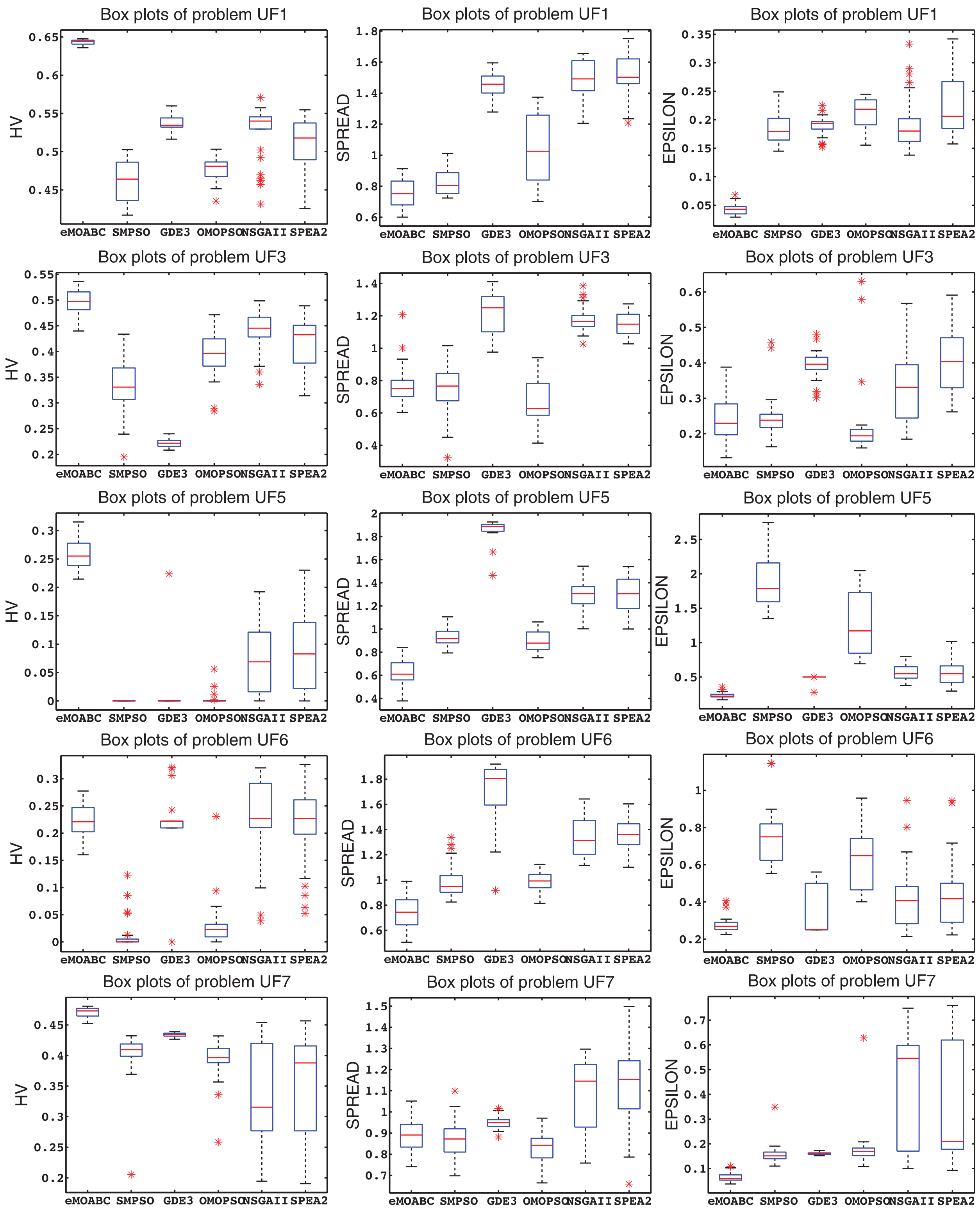

Fig. 9. Box plot graphs of indicators HV (the left panel), SPREAD (the middle panel) and EPSILON (the right panel) on problems UF1, UF3, UF5–UF7.

It is should be noted that all indicators but HV are to be minimized. For HV, the lager the better.

3.3. Sensitivity analysis on the control parameters

As mentioned earlier, the eMOABC employs the control parameters *limit* and *CS*. In this section, we design a sensitivity analysis experiment to investigate the influence of the two parameters on the eMOABC algorithm. Let values of the *limit* be 10, 50, 100, 150, 200 and 250, then we get six different algorithms, denoted as eMOABCa, eMOABCb, eMOABCc, eMOABCd, eMOABCe and eMOABCf, respectively. The eMOABC algorithm family is independently tested on all benchmarks shown in Section 3.1 with the size of the bee colony and the archive being 20 and 100, respectively. These algorithms are

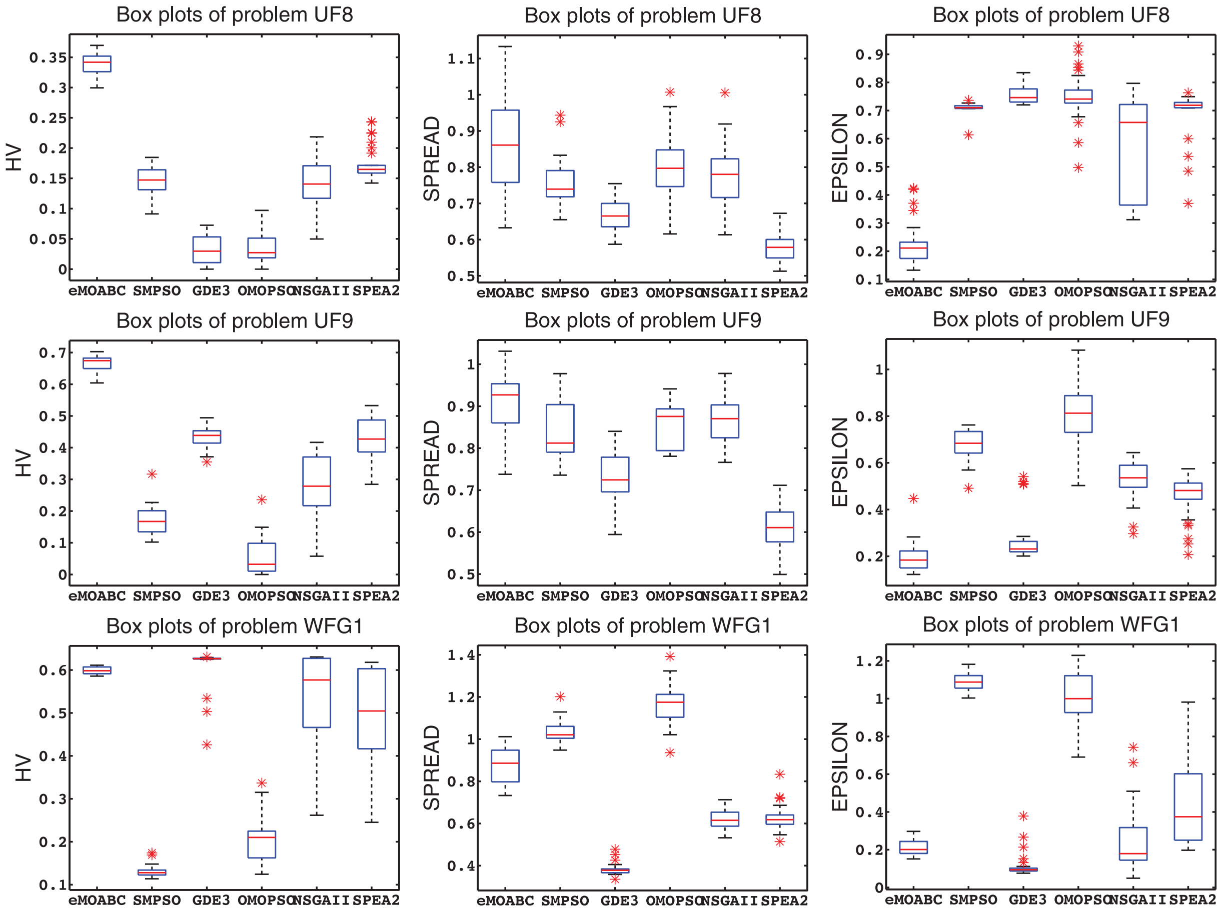

Fig. 10. Box plot graphs of indicators HV (the left panel), SPREAD (the middle panel) and EPSILON (the right panel) on problems UF8, UF9 and WFG1.

terminated when function evaluations reach the maximum value 50,000. The mean values for each algorithm on each indicator over 30 runs are recorded in Tables 1–4, respectively.

The average rankings of the eMOABC algorithm family are calculated by applying Friedman test (Derrac et al., 2011; Friedman, 1937) using jMetal (Durillo & Nebro, 2011; Durillo, Nebro, & Alba, 2010) software package. The test results are given in Table 5, where the best values are bolded and underlined, and the second best values are bolded only. The Friedman test assumes that the lower the values of the indicators the better. This is true in all the indicators but the HV. Therefore, from Table 5, we can find that eMOABCc and eMOABCd are two best algorithms in terms of HV and EPSILON. As for SPREAD, eMOABCd and eMOABCb perform better, and for IGD indicator, eMOABCc and eMOABCe get better ranks. Hence, it is recommended to set the value of *limit* at the range varying from 50 to 200 for general usage.

In a similar way, the effect of the parameter *CS* is analyzed. Let *CS* be a small (≤10), medium (*between* 20 *and* 80) and large (≥90) size, respectively. Then, through the analysis of the statistical results obtained by using Wilcoxon signed ranks test (Alcalá-Fdez et al., 2011; Derrac et al., 2011)[1] on the mean values of the above four indicators, we find that the eMOABC significantly prefers a medium size of *CS*. The Wilcoxon test results of different values of *CS* are summarized in Table 6, which shows that no significant differences are detected among all the pairwise comparisons when *CS* is varied from 20 to 60 in terms of the indicators HV, SPREAD and IGD. As for EPLISON, only $CS = 40$ significantly improves the algorithm's performance when compared against $CS = 30$ and 60. Hence, it is suggested to assign an integer value between 20 and 60 to *CS* for common usage.

3.4. Simulation results for benchmark functions

In this part, the eMOABC algorithm is compared with MOEA/D, SMPSO, GDE3, AbYSS, CellDE, IBEA, MOCell, OMOPSO, NSGAII, PAES and SPEA2 in terms of indicators HV, SPREAD, EPSILON and IGD. For all algorithms, the solutions are coded by real numbers. The maximal number of the solutions in the approximate set produced by each algorithm for computing indicators (e.g. *AS* used in eMOABC) is 100 for both two and three objectives problems. And the maximum number of function evaluations is set to be 50,000 for all test functions. According to Section 3.3, the parameters *limit* and *CS* are set to be 100 and 20, respectively. Thus, the *maxCycle* will be $50,000/20 = 2500$. The eMOABC is implemented by our JAVA code, and other algorithms are realized by Durillo & Nebro's jMetal software package (Durillo & Nebro, 2011; Durillo et al., 2010). For details of default parameter settings for the algorithms, please refer to jMetal source code available at http://jmetal.sourceforge.net/. For each problem, all algorithms are repeated 30 times and four indicators are calculated for each run.

The experimental results of indicators HV, SPREAD, EPSILON and IGD are presented in Tables 7–10, respectively. To ease the analysis of these tables, some cells have a gray colored background in each row; particularly, there are two different gray levels: a darker one, pointing out the algorithm obtaining the best value of the indicator, and a lighter one, highlighting the algorithm obtaining the second best value of the indicator (Durillo & Nebro, 2011).

Firstly, we describe the values obtained in the HV indicator. For this indicator, the higher the value the better the computed fronts.

[1] Throughout this work, the Wilcoxon test results are computed by the module for nonparametric statistical analysis in the KEEL Software Tool available at http://www.keel.es/datasets.php.

Table 14
Mean and standard deviation for *eMOABC* and its variants on HV (standard deviations are in brackets).

	eMOABC	*MOABCrandElite*	*MOABCcrowdedElite*	*MOABCnoElite*
UF1	6.430*e* – 01 (3.3*e* – 03)	6.422*e* – 01 (3.0*e* – 03)	6.409*e* – 01 (3.7*e* – 03)	6.360*e* – 01 (5.7*e* – 03)
UF2	6.497*e* – 01 (2.3*e* – 03)	6.491*e* – 01 (3.7*e* – 03)	6.490*e* – 01 (2.5*e* – 03)	6.468*e* – 01 (3.6*e* – 03)
UF3	4.981*e* – 01 (2.3*e* – 02)	4.965*e* – 01 (1.9*e* – 02)	4.891*e* – 01 (2.2*e* – 02)	5.026*e* – 01 (2.4*e* – 02)
UF4	2.683*e* – 01 (1.4*e* – 03)	2.697*e* – 01 (1.3*e* – 03)	2.673*e* – 01 (2.1*e* – 03)	2.658*e* – 01 (2.2*e* – 03)
UF5	2.579*e* – 01 (2.6*e* – 02)	2.537*e* – 01 (2.7*e* – 02)	2.469*e* – 01 (2.9*e* – 02)	2.894*e* – 01 (2.9*e* – 02)
UF6	2.241*e* – 01 (2.9*e* – 02)	2.072*e* – 01 (2.7*e* – 02)	2.204*e* – 01 (3.6*e* – 02)	2.863*e* – 01 (3.2*e* – 02)
UF7	4.704*e* – 01 (8.1*e* – 03)	4.698*e* – 01 (8.3*e* – 03)	4.570*e* – 01 (2.0*e* – 02)	4.577*e* – 01 (2.3*e* – 02)
UF8	3.388*e* – 01 (1.7*e* – 02)	3.275*e* – 01 (2.5*e* – 02)	3.233*e* – 01 (2.9*e* – 02)	3.184*e* – 01 (3.5*e* – 02)
UF9	6.657*e* – 01 (2.3*e* – 02)	6.708*e* – 01 (2.3*e* – 02)	6.740*e* – 01 (2.2*e* – 02)	6.667*e* – 01 (3.3*e* – 02)
UF10	1.497*e* – 01 (6.7*e* – 02)	1.860*e* – 01 (4.4*e* – 02)	1.791*e* – 01 (4.8*e* – 02)	1.843*e* – 01 (4.3*e* – 02)
ZDT1	6.620*e* – 01 (3.6*e* – 05)	6.620*e* – 01 (4.7*e* – 05)	6.619*e* – 01 (8.1*e* – 05)	6.618*e* – 01 (4.8*e* – 04)
ZDT2	3.287*e* – 01 (3.2*e* – 05)	3.287*e* – 01 (2.7*e* – 05)	3.287*e* – 01 (4.0*e* – 05)	3.286*e* – 01 (1.0*e* – 04)
ZDT3	5.155*e* – 01 (1.4*e* – 04)	5.153*e* – 01 (3.2*e* – 04)	5.149*e* – 01 (5.9*e* – 04)	5.128*e* – 01 (7.3*e* – 03)
ZDT4	6.620*e* – 01 (1.7*e* – 05)	6.620*e* – 01 (2.0*e* – 05)	6.620*e* – 01 (1.9*e* – 05)	6.620*e* – 01 (4.2*e* – 05)
ZDT6	4.014*e* – 01 (1.7*e* – 05)	4.014*e* – 01 (2.5*e* – 05)	4.013*e* – 01 (8.0*e* – 05)	4.012*e* – 01 (2.3*e* – 04)
WFG1	5.997*e* – 01 (8.3*e* – 03)	5.935*e* – 01 (4.6*e* – 03)	5.901*e* – 01 (6.1*e* – 03)	5.817*e* – 01 (5.3*e* – 03)
WFG2	5.568*e* – 01 (4.8*e* – 03)	5.539*e* – 01 (4.4*e* – 03)	5.529*e* – 01 (5.9*e* – 03)	5.482*e* – 01 (7.7*e* – 03)
WFG3	4.381*e* – 01 (2.4*e* – 03)	4.373*e* – 01 (3.8*e* – 03)	4.380*e* – 01 (2.4*e* – 03)	4.339*e* – 01 (8.0*e* – 03)
WFG4	2.193*e* – 01 (1.9*e* – 05)	2.193*e* – 01 (2.9*e* – 05)	2.193*e* – 01 (2.3*e* – 05)	2.193*e* – 01 (4.2*e* – 05)
WFG5	1.972*e* – 01 (2.3*e* – 03)	1.980*e* – 01 (3.1*e* – 03)	1.967*e* – 01 (1.3*e* – 03)	1.985*e* – 01 (2.8*e* – 03)
WFG6	2.042*e* – 01 (3.7*e* – 03)	2.034*e* – 01 (4.0*e* – 03)	2.032*e* – 01 (3.5*e* – 03)	1.989*e* – 01 (6.4*e* – 03)
WFG7	2.108*e* – 01 (3.6*e* – 05)	2.108*e* – 01 (2.9*e* – 05)	2.108*e* – 01 (7.2*e* – 05)	2.108*e* – 01 (5.4*e* – 05)
WFG8	1.680*e* – 01 (1.7*e* – 02)	1.597*e* – 01 (1.4*e* – 02)	1.645*e* – 01 (1.3*e* – 02)	1.585*e* – 01 (1.5*e* – 02)
WFG9	2.372*e* – 01 (5.6*e* – 04)	2.372*e* – 01 (4.9*e* – 04)	2.373*e* – 01 (5.6*e* – 04)	2.375*e* – 01 (7.0*e* – 04)
DTLZ1	7.721*e* – 01 (3.2*e* – 03)	7.487*e* – 01 (1.2*e* – 01)	7.710*e* – 01 (3.8*e* – 03)	7.696*e* – 01 (4.1*e* – 03)
DTLZ2	3.941*e* – 01 (3.7*e* – 03)	3.921*e* – 01 (3.5*e* – 03)	3.920*e* – 01 (5.0*e* – 03)	3.921*e* – 01 (4.1*e* – 03)
DTLZ3	3.947*e* – 01 (5.7*e* – 03)	3.916*e* – 01 (5.2*e* – 03)	3.914*e* – 01 (3.5*e* – 03)	3.895*e* – 01 (6.3*e* – 03)
DTLZ4	3.298*e* – 01 (6.2*e* – 02)	3.880*e* – 01 (1.9*e* – 02)	3.199*e* – 01 (7.2*e* – 02)	2.797*e* – 01 (5.9*e* – 02)
DTLZ5	9.402*e* – 02 (3.0*e* – 05)	9.405*e* – 02 (2.4*e* – 05)	9.404*e* – 02 (3.2*e* – 05)	9.404*e* – 02 (4.0*e* – 05)
DTLZ6	9.498*e* – 02 (2.7*e* – 05)	9.497*e* – 02 (3.6*e* – 05)	9.496*e* – 02 (4.0*e* – 05)	9.493*e* – 02 (4.4*e* – 05)
DTLZ7	2.936*e* – 01 (8.5*e* – 03)	2.958*e* – 01 (7.0*e* – 03)	2.803*e* – 01 (2.2*e* – 02)	2.755*e* – 01 (1.8*e* – 02)

It is apparent from Table 7 that the eMOABC is the most competitive algorithm obtaining the best or second best values on 21 out of the 31 problems, and then it is the GDE3 algorithm which has computed the best or the second best fronts regarding to this indicator on 12 of the 31 evaluated problems. MOEA/D, SMPSO, AbYSS, MOCell and OMOPSO perform similarly, while CellDE and PAES give poor results regarding this indicator.

Next, we pay attention to the SPREAD indicator (Table 8). In this case, the best algorithm is OMOPSO that has obtained the best or the second best values on 10 evaluated problems. Roughly, SMPSO, eMOABC, AbYSS, CellDE, MOCell, SPEA2 and GDE3 give competitive results regarding this indicator. The rest of the algorithms (MOEA/D, PAES, IBEA and NSGAII) obtain worse fronts compared to other algorithms.

We now analyze the computed fronts in terms of the EPSILON values (Table 9). Regarding this indicator, the eMOABC has been also the best algorithm obtaining the best or the second best values in 16 cases, followed by SMPSO algorithm, which performs the best or the second best on 13 evaluated problems. GDE3 may be the third best algorithm which outperforms other algorithms on considerable number of problems. CellDE, IBEA and PAES are three worst algorithms in terms of this indicator.

Finally, we compare algorithms in terms of IGD indicator (Table 10). Definitely, in this case, the eMOABC has performed the best. Other promising algorithms are MOEA/D, SMPSO, GDE3, MOCell, OMOPSO and SPEA2. At last, IBEA and PAES are two algorithms performing the worst on this indicator.

Table 11 lists the average rankings of all algorithms computed by applying the Friedman test. It is shown by the test results that eMOABC, SMPSO, GDE3 and OMOPSO are four promising algorithms that perform the best or the second best in terms of at least one indicator. Specially, our eMOABC gets the first ranking on three indicators: HV, EPSILON and IGD, indicating its great superiority over its competitors. By observing Tables 7–10 carefully, we find that the eMOABC is especially effective when approximating the Pareto fronts of CEC09 benchmarks (UF1–UF10), which are generally deemed to be the most difficult MOPs to be optimized. That is to say, the eMOABC is able to compute high-quality Pareto fronts when dealing with more difficult multi-objective problems.

The Wilcoxon test results for each pairwise comparison between the eMOABC and one of the referred algorithms are provided in Table 12. It gives the R^+, R^-, *p*-value and significance (Sig.) sign for all pairwise comparisons associated with the eMOABC. It can be seen from this table that the eMOABC significantly improves all referred algorithms on indicators HV, EPSILON and IGD, with a level of significance $\alpha = 0.01$ in most cases, and $\alpha = 0.05$ for the pairwise comparisons: eMOABC VS SMPSO on EPSILON, and eMOABC VS GDE3 on IGD. As for indicator SPREAD, the eMOABC shows an improvement over MOEA/D, IBEA, NSGAII and PAES, with a level of significance $\alpha = 0.01$, and achieves comparable results

Table 15
Mean and standard deviation of *eMOABC* and its variants on SPREAD (standard deviations are in brackets).

	eMOABC	*MOABCrandElite*	*MOABCcrowdedElite*	*MOABCnoElite*
UF1	7.576e − 01 (9.0e − 02)	8.216e − 01 (1.1e − 01)	8.495e − 01 (1.5e − 01)	1.088e + 00 (1.5e − 01)
UF2	7.886e − 01 (5.3e − 02)	7.667e − 01 (4.9e − 02)	8.035e − 01 (7.2e − 02)	7.475e − 01 (8.1e − 02)
UF3	7.714e − 01 (1.2e − 01)	7.216e − 01 (7.9e − 02)	7.637e − 01 (9.7e − 02)	8.965e − 01 (1.3e − 01)
UF4	5.923e − 01 (5.1e − 02)	6.090e − 01 (5.7e − 02)	6.228e − 01 (5.3e − 02)	6.665e − 01 (7.3e − 02)
UF5	6.204e − 01 (1.1e − 01)	5.392e − 01 (1.4e − 01)	5.270e − 01 (1.3e − 01)	5.701e − 01 (1.3e − 01)
UF6	7.535e − 01 (1.2e − 01)	7.076e − 01 (1.8e − 01)	6.308e − 01 (1.2e − 01)	6.927e − 01 (1.1e − 01)
UF7	8.840e − 01 (7.0e − 02)	9.238e − 01 (8.1e − 02)	9.639e − 01 (1.2e − 01)	1.055e + 00 (1.5e − 01)
UF8	8.608e − 01 (1.2e − 01)	8.832e − 01 (1.2e − 01)	9.208e − 01 (1.5e − 01)	8.922e − 01 (1.7e − 01)
UF9	9.066e − 01 (7.5e − 02)	9.041e − 01 (7.0e − 02)	9.325e − 01 (9.1e − 02)	9.275e − 01 (7.3e − 02)
UF10	1.094e + 00 (2.5e − 01)	1.145e + 00 (2.9e − 01)	1.067e + 00 (2.4e − 01)	1.201e + 00 (2.2e − 01)
ZDT1	1.164e − 01 (1.0e − 02)	1.255e − 01 (1.1e − 02)	1.342e − 01 (1.5e − 02)	1.418e − 01 (5.4e − 02)
ZDT2	1.124e − 01 (1.1e − 02)	1.171e − 01 (1.1e − 02)	1.173e − 01 (1.5e − 02)	1.282e − 01 (2.0e − 02)
ZDT3	7.997e − 01 (2.7e − 02)	8.547e − 01 (3.6e − 02)	8.966e − 01 (5.6e − 02)	9.225e − 01 (9.9e − 02)
ZDT4	9.142e − 02 (1.3e − 02)	9.867e − 02 (1.1e − 02)	1.021e − 01 (1.3e − 02)	1.069e − 01 (1.8e − 02)
ZDT6	1.309e − 01 (2.0e − 01)	1.263e − 01 (7.2e − 02)	1.348e − 01 (2.6e − 02)	2.258e − 01 (2.2e − 01)
WFG1	8.685e − 01 (8.4e − 02)	9.358e − 01 (3.4e − 02)	9.716e − 01 (4.5e − 02)	1.059e + 00 (3.3e − 02)
WFG2	8.984e − 01 (7.6e − 02)	9.662e − 01 (9.9e − 02)	9.387e − 01 (8.6e − 02)	9.248e − 01 (9.0e − 02)
WFG3	4.013e − 01 (1.6e − 02)	4.116e − 01 (2.1e − 02)	4.074e − 01 (1.9e − 02)	4.204e − 01 (2.6e − 02)
WFG4	1.355e − 01 (1.4e − 02)	1.423e − 01 (1.3e − 02)	1.381e − 01 (1.3e − 02)	1.454e − 01 (1.5e − 02)
WFG5	1.481e − 01 (1.8e − 02)	1.440e − 01 (1.9e − 02)	1.429e − 01 (1.4e − 02)	1.676e − 01 (4.0e − 02)
WFG6	1.483e − 01 (4.0e − 02)	1.685e − 01 (3.4e − 02)	1.655e − 01 (2.9e − 02)	1.879e − 01 (5.4e − 02)
WFG7	1.179e − 01 (1.2e − 02)	1.273e − 01 (1.1e − 02)	1.215e − 01 (1.3e − 02)	1.420e − 01 (1.4e − 02)
WFG8	6.672e − 01 (8.9e − 02)	6.832e − 01 (9.0e − 02)	7.066e − 01 (1.0e − 01)	6.914e − 01 (9.2e − 02)
WFG9	1.819e − 01 (1.6e − 02)	1.879e − 01 (1.7e − 02)	1.874e − 01 (1.8e − 02)	1.850e − 01 (1.8e − 02)
DTLZ1	7.164e − 01 (3.3e − 02)	7.227e − 01 (3.9e − 02)	7.274e − 01 (3.7e − 02)	7.187e − 01 (4.0e − 02)
DTLZ2	6.357e − 01 (3.7e − 02)	6.350e − 01 (3.4e − 02)	6.201e − 01 (3.5e − 02)	6.173e − 01 (4.4e − 02)
DTLZ3	6.223e − 01 (4.0e − 02)	6.140e − 01 (4.3e − 02)	6.290e − 01 (3.1e − 02)	6.605e − 01 (5.7e − 02)
DTLZ4	9.999e − 01 (3.3e − 01)	7.554e − 01 (1.4e − 01)	1.077e + 00 (3.6e − 01)	1.311e + 00 (2.2e − 01)
DTLZ5	1.261e − 01 (1.4e − 02)	1.227e − 01 (1.2e − 02)	1.254e − 01 (1.2e − 02)	1.292e − 01 (1.2e − 02)
DTLZ6	1.109e − 01 (1.4e − 02)	1.154e − 01 (1.6e − 02)	1.106e − 01 (1.7e − 02)	1.229e − 01 (1.4e − 02)
DTLZ7	7.205e − 01 (5.6e − 02)	6.994e − 01 (4.7e − 02)	7.648e − 01 (7.9e − 02)	7.968e − 01 (8.7e − 02)

compared with SMPSO, GDE3, AbYSS, CellDE, MOCell, OMOPSO and SPEA2.

To evaluate the speed of the algorithms, Table 13 records the mean of the runtime over 30 runs. It is shown by the table that PAES is the fastest algorithm, then it is OMOPSO and SMPSO. The eMOABC gets second best results when optimizing UF5 and UF6. Although PAES, OMOPSO and SMPSO are better than our algorithm in terms of the runtime, the eMOABC achieves much better results than its competitors as for indicators HV, EPSILON and IGD (see Table 12). On the whole, the eMOABC is an effective algorithm with an acceptable running speed.

For the purpose of intuitional comparison, we plot the computed fronts with the lowest IGD indicator values obtained in 30 runs of four promising algorithms (eMOABC, SMPSO, GDE3 and OMOPSO), together with NSGAII and SPEA2, on problems UF1, UF3, UF5–UF9 and WFG1 (see Figs. 7 and 8). NSGAII and SPEA2 are chosen because the newly proposed eMOABC uses the same fitness calculation method as in SPEA2, and the same crowding-distance based archive as in NSGAII. The reason of choosing the above eight problems is that the differences among the mentioned six algorithms on them are more obvious.

It can be seen from Figs. 7 and 8 that the eMOABC can well converge to the true Pareto fronts with better distribution on the considered problems. More detailed information can be obtained if we display the results by using box plots, which constitutes a useful way of depicting groups of numerical data. In Figs. 7 and 8, the left is the computed fronts of the mentioned algorithms, and the right is the box plots of IGD indicator over 30 runs. Some interesting facts can be worked out by observing the figures.

We first take a look at Fig. 7. It is observed that the eMOABC clearly outperforms other algorithms on problems UF1, UF3 and UF5. For problem UF3, GDE3 is the best algorithm, and the performances of eMOABC, NSGAII and SPEA2 are similar. However, the eMOABC is more stable than NSGAII and SPEA2 since it gets less extreme values over the 30 runs. Next, it can be seen from Fig. 8 that eMOABC ranks the first on UF7, UF8 and UF9. For problem WFG1, the best algorithm is GDE3, followed by the eMOABC. But, GDE3 archives unsteady results because of some observed extreme values.

Box plot graphs for indicators HV, SPREAD and EPSILON are also provided in Figs. 9 and 10. By observing these figures, we could find the eMOABC obtains the best results in most cases, indicating its great superiority when compared to the state-of-the-art multi-objective algorithms.

4. The effect of the elitism strategy

In this section, we are going to show the effect of the elitism strategy introduced in eMOABC algorithm by a simulation experiment. As mentioned before, the eMOABC will select an elite from the archive which is defined as the intermediate solution with maximum crowding-distance value, and then use it to generate new food sources by Eq. (3) in the employed bees phase. Next, a greedy strategy is used

Table 16
Mean and standard deviation of *eMOABC* and its variants on EPSILON (standard deviations are in brackets).

	eMOABC	*MOABCrandElite*	*MOABCcrowdedElite*	*MOABCnoElite*
UF1	$4.387e-02$ ($9.9e-03$)	$4.774e-02$ ($1.1e-02$)	$4.992e-02$ ($1.1e-02$)	$6.547e-02$ ($1.7e-02$)
UF2	$4.746e-02$ ($1.5e-02$)	$5.056e-02$ ($1.3e-02$)	$4.662e-02$ ($1.1e-02$)	$5.921e-02$ ($1.8e-02$)
UF3	$2.366e-01$ ($5.7e-02$)	$2.354e-01$ ($5.0e-02$)	$2.474e-01$ ($5.6e-02$)	$2.561e-01$ ($6.0e-02$)
UF4	$5.669e-02$ ($4.1e-03$)	$5.630e-02$ ($5.8e-03$)	$5.818e-02$ ($5.0e-03$)	$6.547e-02$ ($5.8e-03$)
UF5	$2.302e-01$ ($3.9e-02$)	$2.290e-01$ ($3.4e-02$)	$2.379e-01$ ($3.5e-02$)	$2.059e-01$ ($3.1e-02$)
UF6	$2.794e-01$ ($4.3e-02$)	$2.546e-01$ ($5.4e-02$)	$2.804e-01$ ($7.6e-02$)	$2.170e-01$ ($8.1e-02$)
UF7	$6.577e-02$ ($1.7e-02$)	$6.880e-02$ ($1.6e-02$)	$9.339e-02$ ($3.7e-02$)	$9.935e-02$ ($8.3e-02$)
UF8	$2.222e-01$ ($7.4e-02$)	$2.197e-01$ ($6.2e-02$)	$2.646e-01$ ($1.3e-01$)	$2.576e-01$ ($1.1e-01$)
UF9	$1.945e-01$ ($6.2e-02$)	$2.020e-01$ ($8.2e-02$)	$1.947e-01$ ($4.8e-02$)	$2.217e-01$ ($7.5e-02$)
UF10	$6.110e-01$ ($1.1e-01$)	$6.440e-01$ ($9.5e-02$)	$6.757e-01$ ($8.1e-02$)	$6.408e-01$ ($8.5e-02$)
ZDT1	$6.177e-03$ ($3.3e-04$)	$6.353e-03$ ($3.7e-04$)	$6.851e-03$ ($9.6e-04$)	$7.690e-03$ ($3.4e-03$)
ZDT2	$6.033e-03$ ($2.5e-04$)	$6.255e-03$ ($3.7e-04$)	$6.425e-03$ ($5.9e-04$)	$6.644e-03$ ($9.1e-04$)
ZDT3	$7.807e-03$ ($2.3e-03$)	$7.915e-03$ ($2.7e-03$)	$1.047e-02$ ($4.7e-03$)	$3.585e-02$ ($9.6e-02$)
ZDT4	$5.734e-03$ ($2.3e-04$)	$5.721e-03$ ($2.8e-04$)	$5.893e-03$ ($2.9e-04$)	$6.025e-03$ ($7.0e-04$)
ZDT6	$5.343e-03$ ($3.3e-04$)	$5.863e-03$ ($5.0e-04$)	$6.726e-03$ ($1.4e-03$)	$8.077e-03$ ($3.1e-03$)
WFG1	$2.124e-01$ ($4.4e-02$)	$2.479e-01$ ($2.5e-02$)	$2.625e-01$ ($3.6e-02$)	$3.051e-01$ ($3.2e-02$)
WFG2	$4.448e-02$ ($4.7e-02$)	$4.839e-02$ ($2.7e-02$)	$5.042e-02$ ($2.6e-02$)	$9.993e-02$ ($6.5e-02$)
WFG3	$2.010e+00$ ($6.0e-03$)	$2.012e+00$ ($1.0e-02$)	$2.010e+00$ ($6.3e-03$)	$2.022e+00$ ($2.0e-02$)
WFG4	$1.403e-02$ ($5.9e-04$)	$1.394e-02$ ($5.2e-04$)	$1.410e-02$ ($6.4e-04$)	$1.367e-02$ ($5.0e-04$)
WFG5	$6.497e-02$ ($1.3e-02$)	$6.096e-02$ ($4.0e-03$)	$6.450e-02$ ($9.6e-03$)	$6.146e-02$ ($3.1e-03$)
WFG6	$2.545e-02$ ($6.1e-03$)	$2.759e-02$ ($6.6e-03$)	$2.803e-02$ ($6.4e-03$)	$3.424e-02$ ($1.0e-02$)
WFG7	$1.427e-02$ ($5.3e-04$)	$1.421e-02$ ($4.6e-04$)	$1.428e-02$ ($5.3e-04$)	$1.396e-02$ ($6.2e-04$)
WFG8	$3.054e-01$ ($8.0e-02$)	$3.317e-01$ ($7.5e-02$)	$3.098e-01$ ($7.7e-02$)	$3.606e-01$ ($1.2e-01$)
WFG9	$2.295e-02$ ($1.6e-03$)	$2.313e-02$ ($1.7e-03$)	$2.331e-02$ ($2.0e-03$)	$2.249e-02$ ($1.9e-03$)
DTLZ1	$4.751e-02$ ($5.0e-03$)	$5.320e-02$ ($3.6e-02$)	$4.656e-02$ ($4.2e-03$)	$4.565e-02$ ($3.5e-03$)
DTLZ2	$1.151e-01$ ($1.2e-02$)	$1.204e-01$ ($1.5e-02$)	$1.161e-01$ ($1.4e-02$)	$1.205e-01$ ($1.7e-02$)
DTLZ3	$1.129e-01$ ($1.4e-02$)	$1.214e-01$ ($2.0e-02$)	$1.165e-01$ ($1.4e-02$)	$1.242e-01$ ($2.0e-02$)
DTLZ4	$1.811e-01$ ($6.5e-02$)	$1.147e-01$ ($3.2e-02$)	$2.033e-01$ ($1.1e-01$)	$2.155e-01$ ($4.8e-02$)
DTLZ5	$4.752e-03$ ($2.8e-04$)	$4.426e-03$ ($2.1e-04$)	$4.485e-03$ ($2.4e-04$)	$4.477e-03$ ($2.3e-04$)
DTLZ6	$4.245e-03$ ($2.5e-04$)	$4.205e-03$ ($2.5e-04$)	$4.363e-03$ ($3.6e-04$)	$4.406e-03$ ($3.4e-04$)
DTLZ7	$1.863e-01$ ($2.2e-01$)	$1.501e-01$ ($2.1e-01$)	$7.870e-01$ ($6.4e-01$)	$6.743e-01$ ($4.7e-01$)

to update each food source. Thereafter, the crowding-distance values of solutions in the archive are re-computed. In the following onlooker bees phase, another elite is determined and adopted to guide the flight of the whole bee colony. If one food source's *trial* value exceeds *limit*, then the scout bee phase is carried out. The above procedures are repeated until the number of iterations reaches the maximum value *maxCycle*.

From the flowchart of the eMOABC, we can find there are two elites in each cycle, and each of them is used to update food sources *FN* times in the employed or onlooker bees phases. This operation could ensure more sufficient exploitation around the region where elites locate, and pull the whole bee colony toward the sparsest zone. The advantages of this lie in that it could: (1) exploit more potential solutions around the less crowded regions, and (2) maintain the diversity of solutions in the approximated set. Hence, benefiting from this elitism strategy, the eMOABC will produce the results not only have good convergence but also have an appropriate distribution over the Pareto front in the objective space.

Now, we prove the above analysis through a simulation experiment. The performance of the eMOABC is compared against three MOABC variants: (1) the *MOABCrandElite* that uses randomly selected members as the elites; (2) the *MOABCcrowdedElite* that defines elites as the member located at the most crowded region (i.e. the solution with the minimum crowding-distance) , and (3) the *MOABCnoElite* which uses no elite at all. Tables 14–17 present the simulation results of the above four algorithms on each indicator and each problem under the same algorithm configurations. The Wilcoxon test results of each pairwise comparison between the eMOABC and its variants are summarized in Table 18, showing that the eMOABC significantly improves over *MOABCcrowdedElite* and *MOABCnoElite* in terms of all indicators, and over *MOABCrandElite* on the indicator HV. That is to say, the elitism strategy used in eMOABC algorithm indeed improves the algorithm's performance, and achieves better effects than other strategies as employed in *MOABCcrowdedElite* and *MOABCrandElite*.

The average rankings of the eMOABC and its variants are calculated by the Friedman test procedure and are shown in Table 19, indicating that the eMOABC is the most competitive algorithm which obtains the best rankings on four indicators. Besides, all MOABC variants with elitism get better average rankings than *MOABCnoElite*, implying the positive effect of introducing an elitism strategy in our MOABC framework. As a summary, it is true that the elitism strategy improves the performance of the MOABC algorithms which is depended on the selection method of elites. In each cycle, selecting the member located at the sparsest region as the elite will produce better results.

5. Further comparison on benchmarks in the CEC09 competition

In this section, the results of the eMOABC are further compared with some state-of-the-art algorithms on benchmarks in the CEC09 competition. We run the eMOABC algorithm under the standard

Table 17
Mean and standard deviation of *eMOABC* and its variants on IGD (standard deviations are in brackets).

	eMOABC	*MOABCrandElite*	*MOABCcrowdedElite*	*MOABCnoElite*
UF1	1.480e − 02 (1.9e − 03)	1.575e − 02 (2.2e − 03)	1.646e − 02 (2.8e − 03)	2.094e − 02 (3.5e − 03)
UF2	1.407e − 02 (3.4e − 03)	1.531e − 02 (3.9e − 03)	1.508e − 02 (2.7e − 03)	1.809e − 02 (5.0e − 03)
UF3	1.075e − 01 (1.6e − 02)	1.043e − 01 (1.1e − 02)	1.096e − 01 (1.3e − 02)	1.132e − 01 (2.1e − 02)
UF4	4.123e − 02 (9.8e − 04)	4.015e − 02 (8.4e − 04)	4.155e − 02 (1.2e − 03)	4.196e − 02 (1.7e − 03)
UF5	1.625e − 01 (2.2e − 02)	1.577e − 01 (2.3e − 02)	1.683e − 01 (2.6e − 02)	1.332e − 01 (2.0e − 02)
UF6	2.174e − 01 (3.7e − 02)	1.709e − 01 (4.5e − 02)	1.940e − 01 (5.0e − 02)	1.174e − 01 (5.1e − 02)
UF7	1.872e − 02 (5.4e − 03)	1.917e − 02 (5.5e − 03)	2.722e − 02 (1.3e − 02)	3.078e − 02 (3.2e − 02)
UF8	1.218e − 01 (1.6e − 02)	1.266e − 01 (1.9e − 02)	1.425e − 01 (3.4e − 02)	1.389e − 01 (3.4e − 02)
UF9	6.641e − 02 (1.5e − 02)	6.620e − 02 (1.9e − 02)	6.433e − 02 (1.1e − 02)	7.125e − 02 (2.6e − 02)
UF10	2.950e − 01 (3.4e − 02)	3.104e − 01 (4.7e − 02)	3.138e − 01 (4.3e − 02)	2.584e − 01 (2.9e − 02)
ZDT1	3.737e − 03 (2.8e − 05)	3.722e − 03 (2.1e − 05)	3.713e − 03 (3.2e − 05)	3.783e − 03 (2.0e − 04)
ZDT2	3.850e − 03 (3.2e − 05)	3.842e − 03 (3.8e − 05)	3.839e − 03 (4.9e − 05)	3.879e − 03 (5.0e − 05)
ZDT3	5.164e − 03 (3.3e − 04)	5.545e − 03 (7.2e − 04)	6.214e − 03 (1.0e − 03)	9.268e − 03 (1.1e − 02)
ZDT4	3.710e − 03 (1.9e − 05)	3.694e − 03 (2.5e − 05)	3.682e − 03 (2.3e − 05)	3.696e − 03 (2.8e − 05)
ZDT6	3.025e − 03 (1.4e − 05)	3.053e − 03 (1.8e − 05)	3.099e − 03 (5.6e − 05)	3.187e − 03 (1.3e − 04)
WFG1	1.018e − 01 (1.9e − 02)	1.101e − 01 (1.2e − 02)	1.191e − 01 (1.1e − 02)	1.346e − 01 (1.1e − 02)
WFG2	2.136e − 02 (7.7e − 03)	2.566e − 02 (8.4e − 03)	2.568e − 02 (9.2e − 03)	3.474e − 02 (1.2e − 02)
WFG3	1.462e − 01 (6.0e − 03)	1.486e − 01 (1.0e − 02)	1.464e − 01 (6.0e − 03)	1.584e − 01 (2.4e − 02)
WFG4	1.037e − 02 (1.4e − 04)	1.027e − 02 (1.7e − 04)	1.029e − 02 (1.8e − 04)	1.025e − 02 (1.9e − 04)
WFG5	6.461e − 02 (4.5e − 03)	6.303e − 02 (6.0e − 03)	6.559e − 02 (2.5e − 03)	6.204e − 02 (5.5e − 03)
WFG6	1.915e − 02 (5.5e − 03)	2.036e − 02 (6.0e − 03)	2.036e − 02 (4.7e − 03)	2.756e − 02 (1.1e − 02)
WFG7	1.182e − 02 (1.5e − 04)	1.167e − 02 (1.5e − 04)	1.172e − 02 (1.7e − 04)	1.154e − 02 (2.0e − 04)
WFG8	8.937e − 02 (2.5e − 02)	9.487e − 02 (1.9e − 02)	8.763e − 02 (2.2e − 02)	1.022e − 01 (2.5e − 02)
WFG9	1.504e − 02 (5.6e − 04)	1.507e − 02 (5.6e − 04)	1.500e − 02 (5.9e − 04)	1.465e − 02 (7.4e − 04)
DTLZ1	2.258e − 02 (6.4e − 04)	3.121e − 02 (4.9e − 02)	2.214e − 02 (5.4e − 04)	2.218e − 02 (5.0e − 04)
DTLZ2	6.050e − 02 (1.9e − 03)	6.046e − 02 (1.8e − 03)	5.970e − 02 (1.5e − 03)	6.033e − 02 (2.1e − 03)
DTLZ3	6.253e − 02 (1.9e − 03)	6.184e − 02 (1.9e − 03)	6.230e − 02 (1.6e − 03)	6.412e − 02 (3.0e − 03)
DTLZ4	6.262e − 02 (3.5e − 02)	4.408e − 02 (4.9e − 03)	7.578e − 02 (4.6e − 02)	8.013e − 02 (4.2e − 02)
DTLZ5	4.017e − 03 (9.3e − 05)	4.051e − 03 (9.6e − 05)	4.019e − 03 (9.8e − 05)	4.052e − 03 (8.5e − 05)
DTLZ6	3.934e − 03 (1.2e − 04)	3.963e − 03 (1.2e − 04)	3.905e − 03 (1.3e − 04)	3.963e − 03 (1.3e − 04)
DTLZ7	8.659e − 02 (4.8e − 02)	8.175e − 02 (4.9e − 02)	1.965e − 01 (1.5e − 01)	1.768e − 01 (1.1e − 01)

Table 18
The summary of the Wilcoxon test for eMOABC and its variants on each indicator. • means that the method in the row improves and the method of the column with a significance level $\alpha = 0.1$, while ○ indicates the opposite cases.

	Indicator HV				Indicator SPREAD			
	(1)	(2)	(3)	(4)	(1)	(2)	(3)	(4)
eMOABC(1)	–	•	•	•	–		•	•
MOABCrandElite(2)	○	–	•	•		–	•	•
MOABCcrowdedElite(3)	○	○	–	•	○	○	–	•
MOABCnoElite(4)	○	○	○	–	○	○	○	–
	Indicator EPLISON				Indicator IGD			
	(1)	(2)	(3)	(4)	(1)	(2)	(3)	(4)
eMOABC(1)	–		•	•	–		•	•
MOABCrandElite(2)		–	•	•		–	•	•
MOABCcrowdedElite(3)	○	○	–		○	○	–	
MOABCnoElite(4)	○	○		–	○	○		–

configurations described in CEC09 technical report (Zhang et al., 2009b). Specifically, the maximal number of the solutions in the approximate set for computing the IGD indicator is 100 for two objective problems, and 150 for three objective problems. The maximal number of function evaluations is set to be 300K for all the problems. The eMOABC algorithm is run independently 30 times for each test problem, and the mean values of IGD are presented in Table 20. In this table, the data for the algorithms MOEA/D (Zhang et al., 2009a), MTS (Tseng & Chen, 2009), DMOEADD (Liu et al., 2009) and LiuLiAlgorithm (Liu & Li, 2009) are directly taken from the final report on the CEC09 MOEA competition (Zhang & Suganthan, 2009). The results for S-MOABC/NS are provided by Akay in Akay (2013), and those for dMOABC and NSGAII are obtained by running our JAVA codes and the jMetal software package under the standard CEC09 competition configurations, respectively. The S-MOABC/NS and dMOABC are two relatively new heuristic algorithms that are based on the artificial bee

Table 19
Average rankings of the eMOABC and its variants on each indicator.

Algorithm	Ranking (HV)	Ranking (SPREAD)	Ranking (EPSILON)	Ranking (IGD)
eMOABC	**3.194**	**1.774**	**1.806**	**2.226**
eMOABCrandElite	**2.806**	**2.194**	**2.032**	**2.226**
eMOABCcrowdedElite	2.161	2.613	3.000	2.452
MOABCnoElite	1.839	3.419	3.161	3.097

Table 20
Mean values of IGD for *eMOABC* and the state-of-the-art algorithms on benchmarks in the CEC09 competition, $maxFEs = 300K$.

	eMOABC	MOEA/D	MTS	DMOEADD	LiuLiAlgorithm	S-MOABC/NS	dMOABC	NSGAII
UF1	**0.00494**	**0.00435**	0.00646	0.01038	0.00785	0.02537	0.005897	0.1038
UF2	**0.004324**	0.00679	0.00615	0.00679	0.0123	0.008963	**0.005714**	0.03858
UF3	0.0655	**0.00742**	0.0531	0.03337	**0.01497**	0.268	0.05256	0.1581
UF4	**0.03424**	0.06385	**0.02356**	0.04268	0.0435	0.03829	0.03531	0.05123
UF5	0.1052	0.18071	**0.01489**	0.31454	0.16186	0.3964	0.1046	**0.0801**
UF6	0.1436	**0.00587**	**0.05917**	0.06673	0.17555	0.3347	0.1141	0.258
UF7	**0.006161**	**0.00444**	0.04079	0.01032	0.0073	0.01639	0.01043	0.1854
UF8	0.07508	**0.0584**	0.11251	**0.06841**	0.08235	0.1444	0.1378	0.2213
UF9	**0.03694**	0.07896	0.11442	**0.04896**	0.09391	0.2188	0.1014	0.3159
UF10	0.2486	0.47415	**0.15306**	0.32211	0.44691	0.5022	**0.225**	0.3855

Table 21
Results obtained by the Wilcoxon test for the pairwise comparison between eMOABC and the state-of-the-art algorithms on benchmarks in the CEC09 competition.

eMOABC VS	R^+	R^-	Exact *p*-value	Asymptotic *p*-value
MOEA/D	32.0	23.0	≥0.2	0.610299
MTS	21.0	34.0	≥0.2	1.000000
DMOEADD	35.0	20.0	≥0.2	0.414823
LiuLiAlgorithm	48.0	7.0	0.0371	0.032313
S-MOABC/NS	55.0	0.0	0.0019532	0.004317
dMOABC	33.0	22.0	≥0.2	0.540818
NSGAII	53.0	2.0	0.00586	0.008045

colony algorithm. In the table, the best results are underlined and bolded, and the second best results are bolded only.

It is shown by Table 20 that our eMOABC achieves the best results on UF2 and UF9, and obtains the second best results on UF1, UF4, UF5 and UF7. MOEA/D ranks the first on five problems (UF1, UF3, UF6, UF7 and UF8), and MTS gets the best or the second best results in four cases (UF4, UF5, UF6 and UF10). As far as the number of the best and the second best results one algorithm obtained, the DMOEADD and dMOABC are better than LiuLiAlgorithm, NSGAII and S-MOABC/NS.

Again, by using Wilcoxon test procedure, we get the R^+, R^-, exact and asymptotic *p*-values for each pairwise comparison between eMOABC and the referred algorithms (see Table 21). The table states that the eMOABC shows a significant improvement over S-MOABC/NS and NSGAII with a level of significance $\alpha = 0.01$, and over LiuLiAlgorithm with $\alpha = 0.05$. The eMOABC achieves comparable results when compared with the famous MOEA/D, MTS, DMOEADD and dMOABC. However, our eMOABC algorithm employs less control parameters than its competitors. Besides two common parameters, i.e., the number of returned non-dominated solutions used for computing indicators (or the size of the external archive) and the number of maximum function evaluations (or the termination condition), the algorithms' additional parameters are listed below: eMOABC (two parameters): *CS* and *limit*; dMOABC (three parameters): *depth*, *ColonySize* and *limit* (Zhong et al., 2014); MOEA/D (seven parameters): T, n_r, δ, CR, F, η and p_m (Zhang et al., 2009a); MTS (five parameters):*BONUS*1, *BOUND*2, *#ofLocalSearchTest*, *#ofLocalSearch*, *#ofForeground* (Tseng & Chen, 2009); DMOEADD (five parameters): *the number of multi-parent crossover*, *the number of the new individuals by crossover*, *the mutation rate*, *the temperatureT* and *the number of subdomains N* (Liu et al., 2009). In the practical application of one algorithm, much more efforts may be made by the users on the adjustment of the control parameters. It is especially true when the algorithm employs more parameters or it is sensitive to some of them. Thus, we are confident that our eMOABC is more suitable for applications in real-world problems from this perspective. What is more, the eMOABC gets better results than both MOEA/D and MTS on five problems, and outperforms dMOABC and DMOEADD on six and seven test functions, respectively. As a summary, our proposed eMOABC can be a prime choice when dealing with practical problems for its inherent properties of being simple and effective.

Beside the quantitative comparisons of the investigated algorithms, the graphical representations of the Pareto fronts produced by the eMOABC algorithm are given in Figs. 11 and 12. These figures show the quality of the Pareto fronts produced by the eMOABC algorithm. It can be seen from Fig. 11 that the Pareto front produced by the eMOABC algorithm over the UF1 test problem not only have good convergence but also have an appropriate distribution over the Pareto front in the objective space.

The eMOABC algorithm performs well when optimizing the UF2 test problem. The eMOABC algorithm obtains the first rank on this test problem. Fig. 11 shows that the Pareto front produced can converge to the true front well, but does not have an appropriate distribution. The bottom-right corner of the Pareto front is not successfully covered by the eMOABC algorithm.

For the UF3 test problem, the best convergence is obtained by the MOEA/D algorithm. The results in Fig. 11 show that the approximated Pareto front obtained by the eMOABC covers some parts of the objective space, but the number of returned non-dominated solutions is not sufficient. Hence, the value of the IGD indicator increases over this test problem.

The eMOABC algorithm obtains the second rank on the UF4 test problem compared to other considered algorithms. The best results are found by the MTS algorithm. It is apparent from Fig. 11 that the eMOABC produces a set of solutions which have both good convergence and an appropriate distribution along the true Pareto front.

It seems that the UF5 is a hard problem to be solved. The UF5 has a discontinuous Pareto front. The proposed eMOABC algorithm obtains the fourth rank among eight algorithms. As can be seen from Fig. 11, the eMOABC algorithm produces considerable solutions in the archive which are well distributed. However, it has difficulties in converging to the true Pareto front.

The UF6 has a discontinuous Pareto front, too. The results show that most of the algorithms have difficulty in optimizing this type of test problems. The MOEA/D obtains the best performance on the UF6 test problem. The MTS and DMOEADD algorithms produce competitive results on this test problem. Our eMOABC and dMOABC achieve similar results. Fig. 11 shows that the produced front contains a smaller number of points in the objective space, but has an acceptable distribution.

The eMOABC algorithm obtains the second rank on the UF7 test problem. As shown in Fig. 12, the front produced by the eMOABC has both good convergence and good distribution over the optimal Pareto front.

The UF8 is the first three objectives test problem. Usually, the complexity of multi-objective problems increases by the number of objectives to be optimized. The eMOABC obtains the third rank on this test problem. The best result is obtained by the MOEA/D algorithm. The quality of the approximated Pareto front is shown in Fig. 12. It is apparent from the results that the eMOABC produces a set of

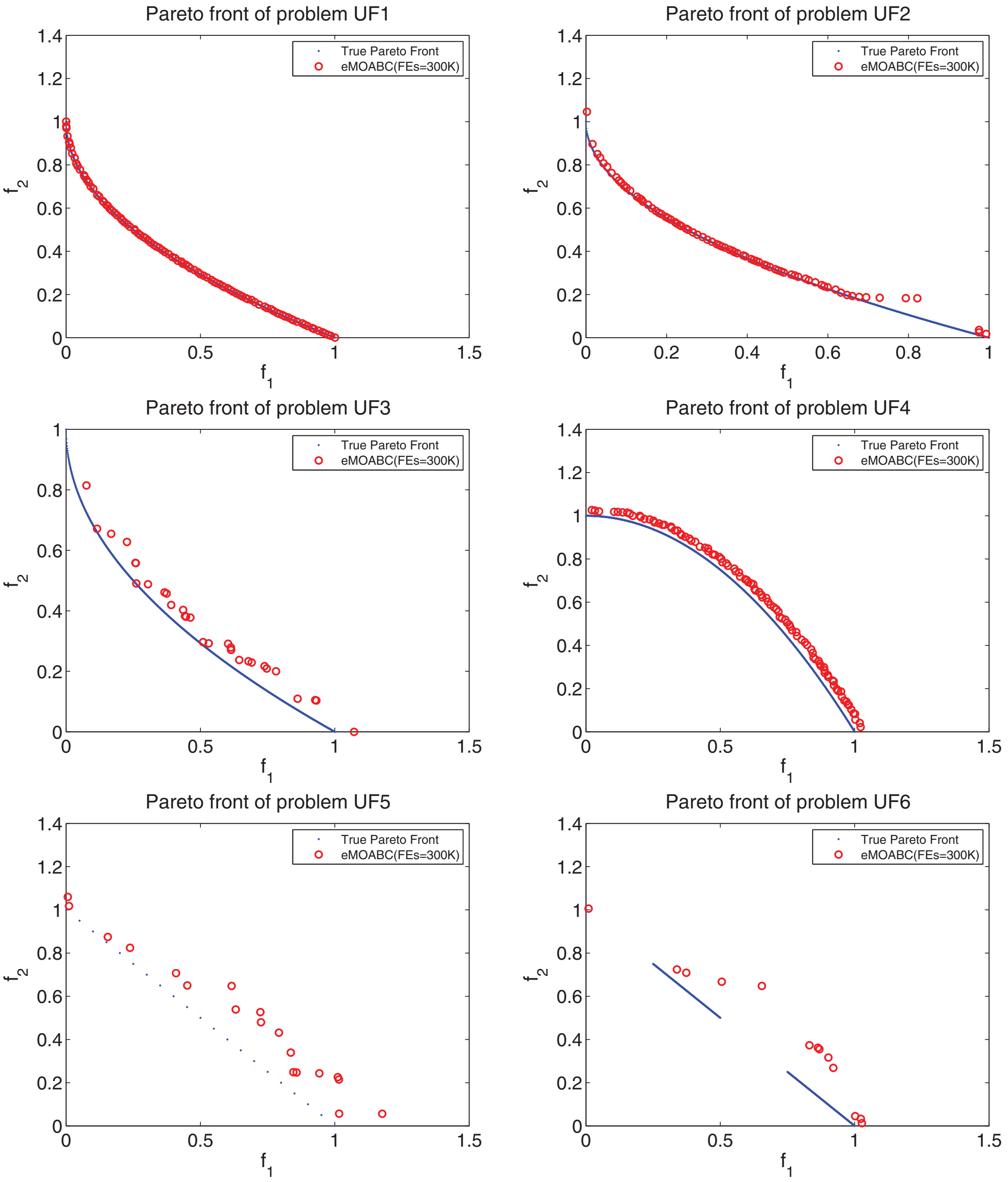

Fig. 11. Pareto fronts obtained by the eMOABC on the CEC09 competition benchmarks (UF1–UF6, maxFEs = 300K).

solution points which have an appropriate distribution in the three dimensional objective space.

The eMOABC algorithm has the ability to solve the UF9 test problem well, and it gets the first rank. The second rank is obtained by the MTS algorithm. The quality of the approximated Pareto front can be seen in Fig. 12. The results show that the eMOABC produces a set of well distributed non-dominated points which cover many parts of the objective space. Thus, the IGD value for this problem decreases naturally.

The MTS algorithm obtains the best result on the UF10 test problem. The eMOABC produces competitive results compared to the dMOABC algorithm. Our eMOABC algorithm obtains the third rank. Fig. 12 shows the quality of the approximated Pareto front by the eMOABC algorithm. The results show that the approximated solutions converge to the true front well, but they only cover a small part of the objective space, resulting in a worse distribution of the computed front.

6. Conclusions

This paper suggests a new multi-objective artificial bee colony algorithm, denoted as eMOABC. In this proposed algorithm, a crowding-distance based archive is used to keep and maintain the non-dominated solutions found so far. In order to keep the diversity of the archive, the solution to be added will automatically replace the member with the minimum value of the crowding-distance in the case when the archive is full. For the purpose of avoiding premature convergence, we adopt an improved ABC algorithm which uses an elitism strategy, that is, the member located at the least crowded region (called the elite) will be selected and used to generate new food sources in both employed and onlooker bees phases. We design a sensitivity variation analysis experiment to explore the parameter selection of *limit* and *CS*.

The eMOABC is evaluated on 31 test problems and compared with 11 state-of-the-art algorithms in terms of five often-used indicators:

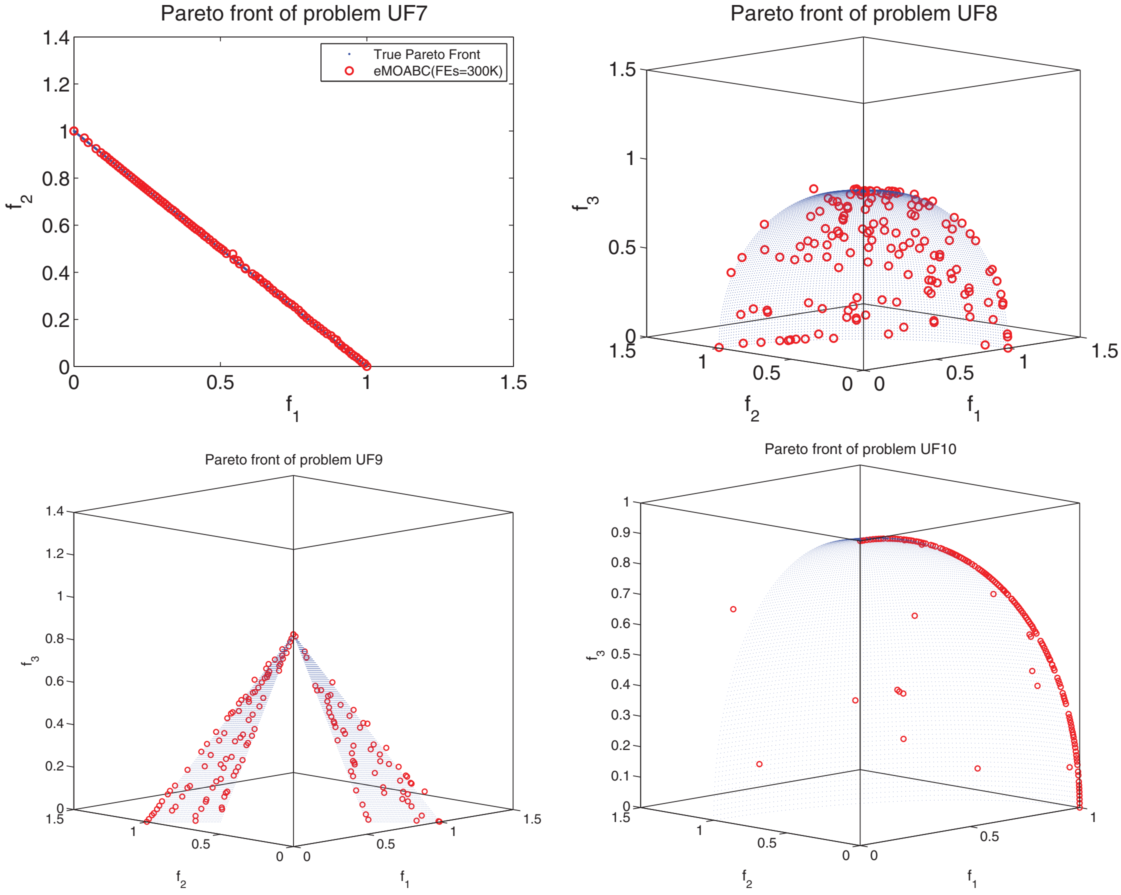

Fig. 12. Pareto fronts obtained by the eMOABC on the CEC09 competition benchmarks (UF7–UF10, maxFEs = 300K).

HV, SPREAD, EPSILON, IGD and RUNTIME. It is proved by the simulation results that our eMOABC is the most salient algorithm since it could provide approximated fronts with good convergence and appropriate distribution in an acceptable runtime.

The results produced by the eMOABC on problems from UF1 to UF10 are compared with those obtained by some well-known algorithms in the CEC09 competition, and by the latest artificial bee colony based multi-objective algorithms. It is shown by the Wilcoxon test procedure that the proposed eMOABC gets comparable results compared to the famous MOEA/D, MTS, DMOEADD and dMOABC, and shows a significant improvement over LiuLiAlgorithm, NSGAII and S-MOABC/NS. However, the number of control parameters used in the eMOABC is less than those employed in MOEA/D, MTS, DMOEADD and dMOABC. In the practical application of one algorithm, it is not an easy work for the users to find the best configurations of the algorithm if it employs many control parameters. It is especially true when the algorithm is sensitive to some of these parameters. Thus, we are confident that our eMOABC is more suitable for applications in real-world problems from this perspective.

Summarizing, according to the problems, the number of parameters, and quality indicators used, the eMOABC is the most promising algorithm in our study. In our future work, we will focus on the practical applications of the eMOABC. Furthermore, it is also a meaningful work to extend the algorithm for many-objective optimization.

Acknowledgment

The authors would like to thank the anonymous reviewers for providing valuable comments to improve this paper, and add special thanks to Durillo and Nebro for their excellent jMetal software package.

References

Akay, B. (2013). Synchronous and asynchronous pareto-based multi-objective artificial bee colony algorithms. *Journal of Global Optimization, 57*(2), 415–445.

Akay, B., & Karaboga, D. (2012). A modified artificial bee colony algorithm for real-parameter optimization. *Information Sciences, 192*(1), 120–142.

Akbari, R., Hedayatzadeh, R., Ziarati, K., & Hassanizadeh, B. (2012). A multi-objective artificial bee colony algorithm. *Swarm and Evolutionary Computation, 2*, 39–52.

Alcalá-Fdez, J., Fernández, A., Luengo, J., Derrac, J., García, S., Sánchez, L., et al. (2011). KEEL data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. *Journal of Multiple-Valued Logic and Soft Computing, 17*(2–3), 255–287.

Coello Coello, C. A., Pulido, G. T., & Lechuga, M. S. (2004). Handling multiple objectives with particle swarm optimization. *IEEE Transactions on Evolutionary Computation, 8*(3), 256–279.

Deb, K. (2001). *Multi-objective optimization using evolutionary algorithms* (pp. 13–46). Wiley.

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002a). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation, 6*(2), 182–197.

Deb, K., Thiele, L., Laumanns, M., & Zitzler, E. (2002b). Scalable multi-objective optimization test problems. In *Proceedings of the congress on evolutionary computation (CEC'02), Honolulu, USA* (pp. 825–830).

Derrac, J., Garcia, S., Molina, D., & Herrera, F. (2011). A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation, 1*(7), 3–18.

Durillo, J. J., & Nebro, A. J. (2011). jMetal: A java framework for multi-objective optimization. *Advances in Engineering Software, 42*, 760–771.

Durillo, J. J., Nebro, A. J., & Alba, E. (2010). The jMetal framework for multi-objective optimization: Design and architecture. In *CEC'10* (pp. 4138–4325).

Durillo, J. J., Nebro, A. J., Luna, F., & Alba, E. (2008). Solving three-objective optimization problems using a new hybrid cellular genetic algorithm. In G. Rudolph, T. Jensen, S. Lucas, C. Poloni, & N. Beume (Eds.), *Parallel problem solving from nature (PPSN x). Lecture Notes in Computer Science, vol. 5199* (pp. 661–670). Springer.

Friedman, M. (1937). The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association, 32*(200), 675–701.

Hedayatzadeh, R., Hasanizadeh, B., Akbari, R., & Ziarati, K. (2010). A multi-objective artificial bee colony for optimizing multi-objective problems. In *The 3rd international conference on advanced computer theory and engineering (ICACTE), vol. 5* (pp. 277–281).

Huband, S., Hingston, P., Barone, L., & While, L. (2006). A review of multiobjective test problems and a scalable test problem toolkit. *IEEE Transactions on Evolutionary Computation, 10*(5), 477–506.

Karaboga, D., & Akay, B. (2009). A comparative study of artificial bee colony algorithm. *Applied Mathematics and Computation, 214*(1), 687–697.

Karaboga, D., & Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm. *Journal of Global Optimization, 39*(3), 459–471.

Kennedy, J., & Eberhart, R. C. (1995). Particle swarm optimization. In *Proceedings of the IEEE international conference on neural networks* (pp. 1942–1948). Washington, DC, USA.

Knowles, J., & Corne, D. (1999). The pareto archived evolution strategy: A new baseline algorithm for pareto multiobjective optimisation. In *Proceeding of congress on evolutionary computation (CEC'99), vol. 1* (pp. 98–105). IEEE.

Kukkonen, S., & Lampinen, J. (2009). Performance assessment of generalized differential evolution 3 with a given set of constrained multi-objective test problems. In *Proceeding of congress on evolutionary computation (CEC'09)* (pp. 1943–1950). IEEE.

Liu, H.-L., Gu, F., & Zhang, Q. (2014). Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems. *IEEE Transactions on Evolutionary Computation, 18*(3), 450–455.

Liu, H.-L., & Li, X. (2009). The multiobjective evolutionary algorithm based on determined weight and sub-regional search. In *IEEE congress on evolutionary computation (CEC'09)* (pp. 1928–1934).

Liu, M., Zou, X., Chen, Y., & Wu, Z. (2009). Performance assessment of DMOEA-DD with CEC 2009 MOEA competition test instances. In *Proceeding of congress on evolutionary computation (CEC'09)* (pp. 2913–2918). IEEE.

Nebro, A. J., Durillo, J., Garcia-Nieto, J., Coello Coello, C., Luna, F., & Alba, E. (2009a). Smpso: A new pso-based metaheuristic for multi-objective optimization. In *IEEE symposium on computational intelligence in miulti-criteria decision-making* (pp. 66–73). IEEE.

Nebro, A. J., Durillo, J. J., Luna, F., Dorronsoro, B., & Alba, E. (2009b). Mocell: A cellular genetic algorithm for multiobjective optimization. *International Journal of Intelligent Systems, 24*(7), 726–746.

Nebro, A. J., Luna, F., Alba, E., Dorronsoro, B., Durillo, J. J., & Beham, A. (2008). Abyss: Adapting scatter search to multiobjective optimization. *IEEE Transactions on Evolutionary Computation, 12*(4), 439–457.

Omkar, S. N., Senthilnath, J., Khandelwal, R., Naik, G. N., & Gopalakrishnan, S. (2011). Artificial bee colony (ABC) for multi-objective design optimization of composite structures. *Applied Soft Computing, 11*(1), 489–499.

Sierra, M. R., & Coello, C. A. C. (2005). Improving PSO-based multi-objective optimization using crowding, mutation and ϵ -dominance. In *Evolutionary multi-criterion optimization* (pp. 505–519). Springer.

Szeto, W. Y., Wu, Y., & Ho, S. C. (2011). An artificial bee colony algorithm for the capacitated vehicle routing problem. *European Journal of Operational Research, 215*(1), 126–135.

Tseng, L.-Y., & Chen, C. (2009). Multiple trajectory search for unconstrained/constrained multi-objective optimization. In *Proceeding of congress on evolutionary computation (CEC'09)* (pp. 1951–1958). IEEE.

Xiang, Y., Peng, Y., Zhong, Y., Chen, Z., Lu, X., & Zhong, X. (2014). A particle swarm inspired multi-elitist artificial bee colony algorithm for real-parameter optimization. *Computational Optimization and Applications, 57*(2), 493–516.

Zhang, Q., Liu, W., & Li, H. (2009a). The performance of a new version of MOEA/D on CEC09 unconstrained MOP instances. In *IEEE congress on evolutionary computing (CEC), Trondheim* (pp. 18–21).

Zhang, Q., & Suganthan, P. N. (2009). Final report on CEC'09 MOEA competition. School of CS & EE, University of Essex, UK; School of EEE, Nanyang Technological University, Singapore. http://www.ntu.edu.sg/home/EPNSugan/

Zhang, Q., Zhou, A., Zhao, S., Suganthan, P. N., Liu, W., & Tiwari, S. (2009b). Multiobjective optimization test instances for the CEC 2009 special session and competition. In: Special Session on Performance Assessment of Multi-Objective Optimization Algorithms, Technical Report. University of Essex, Colchester, UK and Nanyang Technological University, Singapore.

Zhong, Y.-B., Xiang, Y., & Liu, H.-L. (2014). A multi-objective artificial bee colony algorithm based on division of the searching space. *Applied Intelligence, 41*(4), 987–1011.

Zitzler, E., Deb, K., & Thiele, L. (2000). Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation, 8*(2), 173–195.

Zitzler, E., & Künzli, S. (2004). Indicator-based selection in multiobjective search. In *Proceedings of the 8th international conference on parallel problem solving from nature, ppsn viii* (pp. 832–842). Springer.

Zitzler, E., Laumanns, M., & Thiele, L. (2001). SPEA2: Improving the strength Pareto evolutionary algorithm. Technical Report Computer Engineering and Networks Laboratory, Department of Electrical Engineering, Swiss Federal Institute of Technology (ETH) Zurich, Switzerland.

Zitzler, E., & Thiele, L. (1999). Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach. *IEEE Transactions on Evolutionary Computation, 3*(4), 257–271.

Zou, W., Zhu, Y., Chen, H., & Zhang, B. (2011). Solving multiobjective optimization problems using artificial bee colony algorithm. *Discrete Dynamics in Nature and Society, 2011*, 1–37.