Hindawi Publishing Corporation
Discrete Dynamics in Nature and Society
Volume 2011, Article ID 569784, 37 pages
doi:10.1155/2011/569784

Research Article

Solving Multiobjective Optimization Problems Using Artificial Bee Colony Algorithm

Wenping Zou,[1,2] Yunlong Zhu,[1] Hanning Chen,[1] and Beiwei Zhang[1,2]

[1] *Key Laboratory of Industrial Informatics, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China*
[2] *Graduate School of the Chinese Academy of Sciences, Beijing 100039, China*

Correspondence should be addressed to Wenping Zou, zouwp@sia.cn

Received 10 May 2011; Revised 9 August 2011; Accepted 23 August 2011

Academic Editor: Binggen Zhang

Multiobjective optimization has been a difficult problem and focus for research in fields of science and engineering. This paper presents a novel algorithm based on artificial bee colony (ABC) to deal with multi-objective optimization problems. ABC is one of the most recently introduced algorithms based on the intelligent foraging behavior of a honey bee swarm. It uses less control parameters, and it can be efficiently used for solving multimodal and multidimensional optimization problems. Our algorithm uses the concept of Pareto dominance to determine the flight direction of a bee, and it maintains nondominated solution vectors which have been found in an external archive. The proposed algorithm is validated using the standard test problems, and simulation results show that the proposed approach is highly competitive and can be considered a viable alternative to solve multi-objective optimization problems.

1. Introduction

In the real world, many optimization problems have to deal with the simultaneous optimization of two or more objectives. In some cases, however, these objectives are in contradiction with each other. While in single-objective optimization the optimal solution is usually clearly defined, this does not hold for multiobjective optimization problems. Instead of a single optimum, there is rather a set of alternative trade-offs, generally known as Pareto optimal solutions. These solutions are optimal in the wider sense that no other solutions in the search space are superior to them when all objectives are considered.

In the 1950s, in the area of operational research, a variety of methods have been developed for the solution of multiobjective optimization problems (MOP). Some of the most representative classical methods are linear programming, the weighted sum method, and

the goal programming method [1]. Over the past two decades, a number of multiobjective evolutionary algorithms (MOEAs) have been proposed [2, 3]. A few of these algorithms include the nondominated sorting genetic algorithm II (NSGA-II) [4], the strength Pareto evolutionary algorithm 2 (SPEA2) [5], and the multiobjective particle swarm optimization (MOPSO) which is proposed by Coello and Lechuga [6]. MOEA's success is due to their ability to find a set of representative Pareto optimal solutions in a single run.

Artificial bee colony (ABC) algorithm is a new swarm intelligent algorithm that was first introduced by Karaboga in Erciyes University of Turkey in 2005 [7], and the performance of ABC is analyzed in 2007 [8]. The ABC algorithm imitates the behaviors of real bees in finding food sources and sharing the information with other bees. Since ABC algorithm is simple in concept, easy to implement, and has fewer control parameters, it has been widely used in many fields. For these advantages of the ABC algorithm, we present a proposal, called "Multiobjective Artificial Bee Colony" (MOABC), which allows the ABC algorithm to be able to deal with multiobjective optimization problems. We aim at presenting a kind of efficient and simple algorithm for multiobjective optimization, meanwhile filling up the research gap of the ABC algorithm in the field of multiobjective problems. Our MOABC algorithm is based on Nondominated Sorting strategy. We use the concept of Pareto dominance to determine which solution vector is better and use an external archive to maintain nondominated solution vectors. We also use comprehensive learning strategy which is inspired by comprehensive learning particle swarm optimizer (CLPSO) [9] to ensure the diversity of population. In order to evaluate the performance of the MOABC, we compared the performance of the MOABC algorithm with that of NSGA-II and MOCLPSO [10] on a set of well-known benchmark functions. Seven of these test functions SCH, FON, ZDT1 to ZDT4, and ZDT6 are of two objectives, while the other four DTLZ1 to DTLZ3 and DTLZ6 are of three objectives. Meanwhile, a version of MOABC with Nondominated Sorting strategy only is also compared to illustrate the comprehensive learning strategy effect for population diversity. This version algorithm is called nondominated Sorting artificial bee colony (NSABC). From the simulation results, the MOABC algorithm shows remarkable performance improvement over other algorithms in all benchmark functions.

The remainder of the paper is organized as follows. Section 2 gives relevant MOP concepts. In Section 3, we will introduce the ABC algorithm. In Section 4, the details of MOABC algorithm will be given. Section 5 presents a comparative study of the proposed MOABC with NSABC, NSGA-II, and MOCLPSO on a number of benchmark problems, and we will draw conclusions from the study. Section 6 summarizes our discussion.

2. Related Concepts

Definition 1 (multiobjective optimization problem). A multiobjective optimization problem (MOP) can be stated as follows:

$$\begin{aligned} &\text{Minimize} \quad \mathbf{F}(\mathbf{x}) = \left(f_1(\mathbf{x}), \ldots, f_m(\mathbf{x})\right) \\ &\text{Subject to} \quad \mathbf{x} \in X, \end{aligned} \tag{2.1}$$

where $\mathbf{x} = (x_1, \ldots, x_n)$ is called decision (variable) vector, $X \subset R^n$ is the decision (variable) space, R^m is the objective space, and $\mathbf{F} : X \to R^m$ consists of m ($m \geq 2$) real-valued objective functions. $\mathbf{F}(\mathbf{x})$ is the objective vector. We call problem (2.1) a MOP.

Main steps of the ABC algorithm
(1) cycle = 1
(2) Initialize the food source positions (solutions) x_i, $i = 1, \ldots, SN$
(3) Evaluate the nectar amount (fitness function fit_i) of food sources
(4) **repeat**
(5) Employed Bees' Phase
For each employed bee
Produce new food source positions v_i
Calculate the value fit_i
If new position better than previous position
Then memorizes the new position and forgets the old one.
End For.
(6) Calculate the probability values p_i for the solution.
(7) Onlooker Bees' Phase
For each onlooker bee
Chooses a food source depending on p_i
Produce new food source positions v_i
Calculate the value fit_i
If new position better than previous position
Then memorizes the new position and forgets the old one.
End For
(8) Scout Bee Phase
If there is an employed bee becomes scout
Then replace it with a new random source positions
(9) Memorize the best solution achieved so far
(10) cycle = cycle + 1.
(11) **until** cycle = Maximum Cycle Number

Algorithm 1: Pseudocode for ABC algorithm.

Main steps of the MOABC algorithm
(1) cycle = 1
(2) Initialize the food source positions (solutions) $\mathbf{x}_i$, $i = 1, \ldots, SN$
(3) Evaluate the nectar amount (fitness fit_i) of food sources
(4) The initialized solutions are sorted based on nondomination
(5) Store nondominated solutions in the external archive EA
(6) **repeat**
(7) Onlooker Bees' Phase
For each onlooker bee
Randomly chooses a solution from EA
Produce new solution v_i by using expression (4.1)
Calculate the value fit_i
Apply greedy selection mechanism in Algorithm 3 to decide which solution enters EA
EndFor
(8) The solutions in the EA are sorted based on nondomination
(9) Keep the nondomination solutions of them staying in the EA
(10) If the number of nondominated solutions exceeds the allocated the size of EA
Use crowding distance to remove the crowded members
(11) cycle = cycle + 1.
(12) **until** cycle = Maximum Cycle Number

Algorithm 2: Pseudocode for MOABC algorithm.

```
Greedy selection mechanism
If v_i dominates x_i
    Put v_i into EA
Else if x_i dominates v_i
    Do nothing
Else if x_i and v_i are not dominated by each other
    Put v_i into EA
    Produce a random number r drawn from a uniform distribution on the unit interval
    If r < 0.5
        The the original solution is replaced by the new solution as new food source position.
        That means x_i is replaced by v_i.
    Else
        Do nothing
    End If
End If
```

Algorithm 3: Greedy selection mechanism.

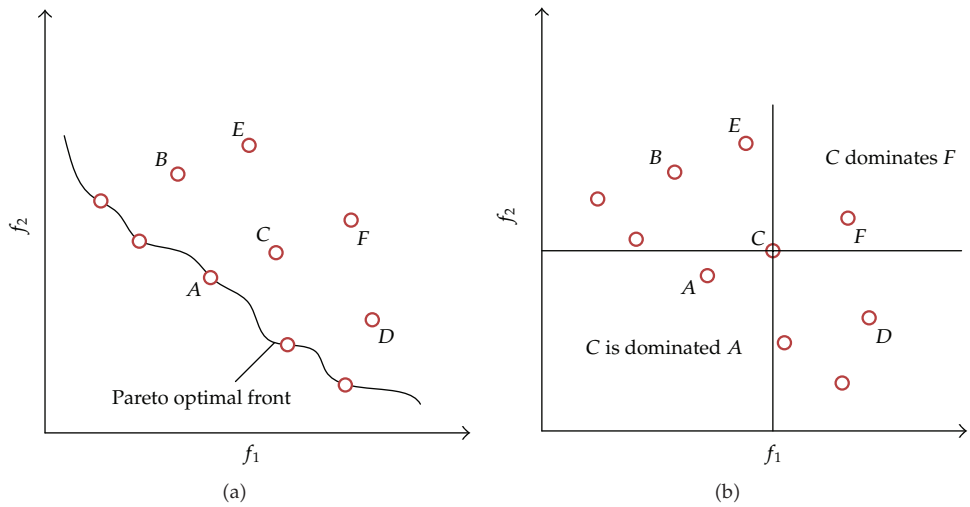

Figure 1: Illustrative example of Pareto optimality in objective space (a) and the possible relations of solutions in objective space (b).

Definition 2 (Pareto optimal). For (2.1), let $\mathbf{a} = (a_1, \ldots, a_m)$, $\mathbf{b} = (b_1, \ldots, b_m) \in R^m$ be two vectors, $\mathbf{a}$ is said to dominate $\mathbf{b}$ if $a_i \leq b_i$ for all $i = 1, \ldots, m$, and $\mathbf{a} \neq \mathbf{b}$. A point $\mathbf{x}^* \in X$ is called (globally) Pareto optimal if there is no $\mathbf{x} \in X$ such that $\mathbf{F}(\mathbf{x})$ dominates $\mathbf{F}(\mathbf{x}^*)$. Pareto optimal solutions are also called efficient, nondominated, and noninferior solutions. The set of all the Pareto optimal solutions, denoted by PS, is called the Pareto set. The set of all the Pareto objectives vectors, $\mathrm{PF} = \{\mathbf{F}(\mathbf{x}) \in R^m \mid \mathbf{x} \in \mathrm{PS}\}$, is called the Pareto front [11, 12]. Illustrative example can be seen in Figure 1.

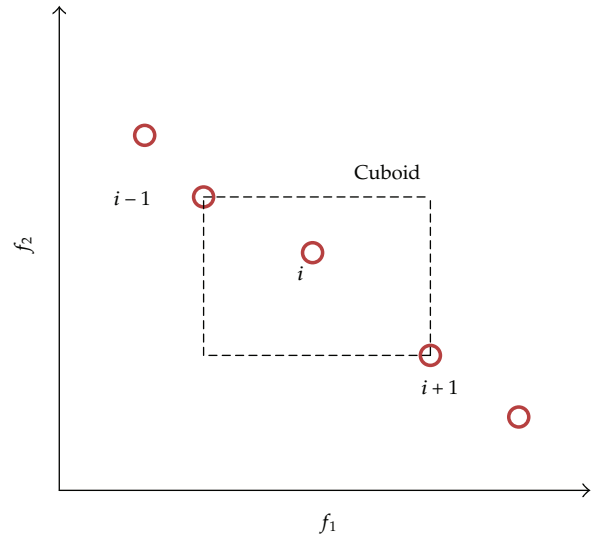

Figure 2: Crowding distance: Points are all nondominated solutions in the external archive.

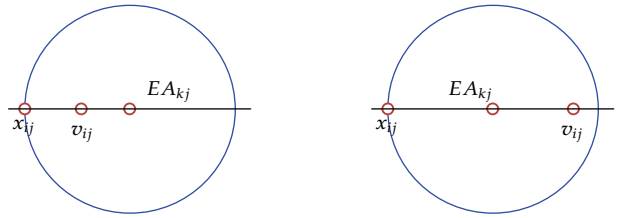

Figure 3: Possible search regions per dimensions: $0 < \phi_j < 1$ (left); $1 < \phi_j < 2$ (right).

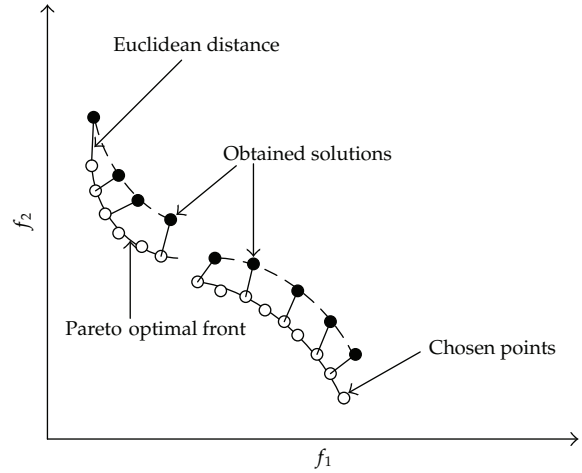

Figure 4: Convergence metric Υ.

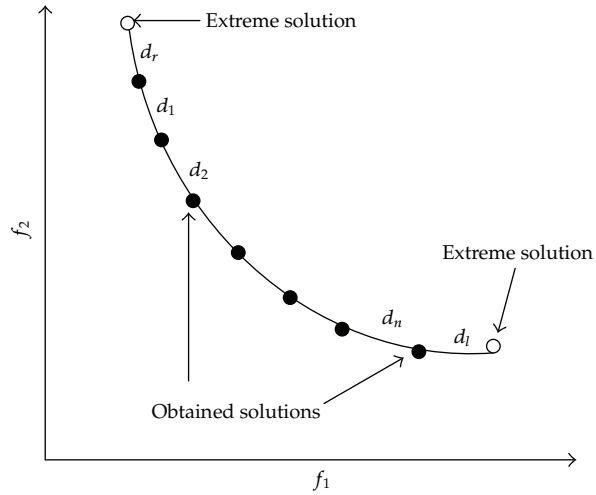

Figure 5: Diversity metric Δ.

3. The Original ABC Algorithm

The artificial bee colony algorithm is a new population-based metaheuristic approach, initially proposed by Karaboga [7] and Karaboga and Basturk [8] and further developed by Karaboga and Basturk [13] and Karaboga and Akay [14]. It has been used in various complex problems. The algorithm simulates the intelligent foraging behavior of honey bee swarms. The algorithm is very simple and robust. In the ABC algorithm, the colony of artificial bees is classified into three categories: employed bees, onlookers, and scouts. Employed bees are associated with a particular food source that they are currently exploiting or are "employed" at. They carry with them information about this particular source and share the information to onlookers. Onlooker bees are those bees that are waiting on the dance area in the hive for the information to be shared by the employed bees about their food sources and then make decision to choose a food source. A bee carrying out random search is called a scout. In the ABC algorithm, the first half of the colony consists of the employed artificial bees, and the second half includes the onlookers. For every food source, there is only one employed bee. In other words, the number of employed bees is equal to the number of food sources around the hive. The employed bee whose food source has been exhausted by the bees becomes a scout. The position of a food source represents a possible solution to the optimization problem, and the nectar amount of a food source corresponds to the quality (fitness) of the associated solution represented by that food source. Onlookers are placed on the food sources by using a probability-based selection process. As the nectar amount of a food source increases, the probability value with which the food source is preferred by onlookers increases, too [7, 8]. The main steps of the algorithm are given in Algorithm 1.

In the initialization phase, the ABC algorithm generates randomly distributed initial food source positions of SN solutions, where SN denotes the size of employed bees or onlooker bees. Each solution $\mathbf{x}_i$ $(i = 1, 2, \ldots, SN)$ is a n-dimensional vector. Here, n is the number of optimization parameters. And then evaluate each nectar amount fit_i. In the ABC algorithm, nectar amount is the value of benchmark function.

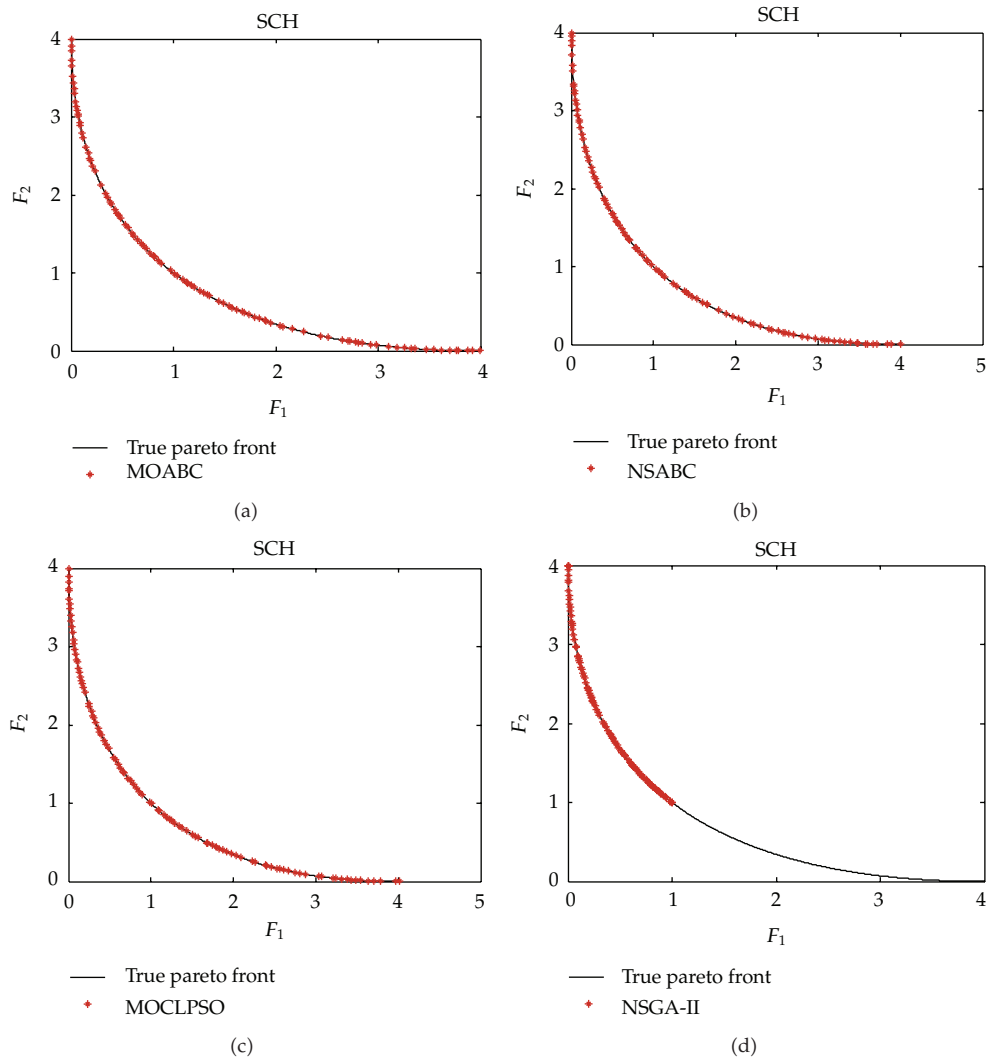

Figure 6: Pareto fronts obtained by NSABC, MOABC, MOCLPSO, and NSGA-II on SCH after 20000 FEs.

In the employed bees' phase, each employed bee finds a new food source v_i in the neighborhood of its current source $\mathbf{x}_i$. The new food source is calculated using the following expression:

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}), \tag{3.1}$$

where $k \in (1, 2, \ldots, SN)$ and $j \in (1, 2, \ldots, n)$ are randomly chosen indexes and $k \neq i$. ϕ_{ij} is a random number between $[-1, 1]$. It controls the production of a neighbour food source position around x_{ij}. And then employed bee compares the new one against the current solution and memorizes the better one by means of a greedy selection mechanism.

In the onlooker bees' phase, each onlooker chooses a food source with a probability which is related to the nectar amount (fitness) of a food source shared by employed bees.

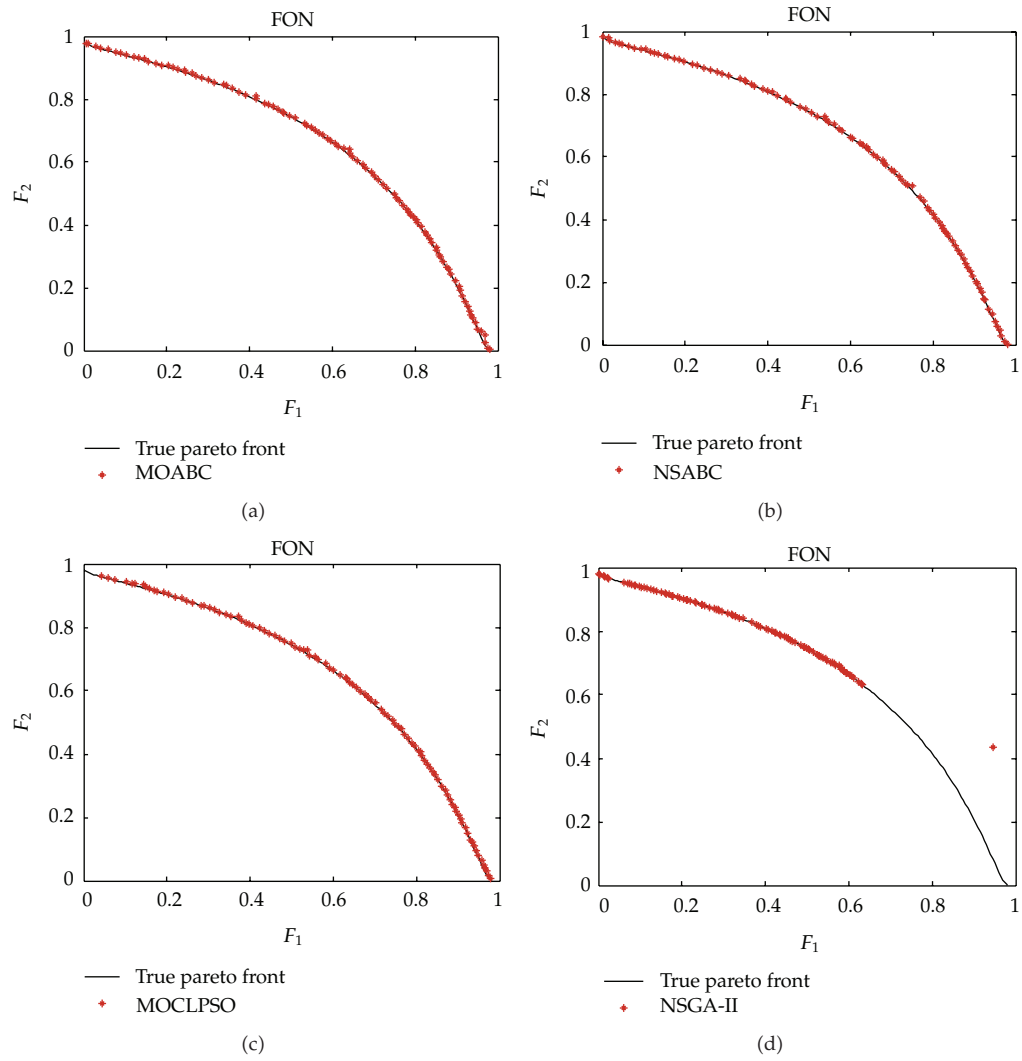

Figure 7: Pareto fronts obtained by NSABC, MOABC, MOCLPSO, and NSGA-II on FON after 20000 FEs.

Probability is calculated using the following expression:

$$p_i = \frac{\text{fit}_i}{\sum_{n=1}^{SN} \text{fit}_i}. \tag{3.2}$$

In the scout bee phase, if a food source cannot be improved through a predetermined cycles, called "limit", it is removed from the population, and the employed bee of that food source becomes scout. The scout bee finds a new random food source position using

$$x_i^j = x_{\min}^j + \text{rand}[0,1]\left(x_{\max}^j - x_{\min}^j\right), \tag{3.3}$$

where $x_{\min}^j$ and $x_{\max}^j$ are lower and upper bounds of parameter j, respectively.

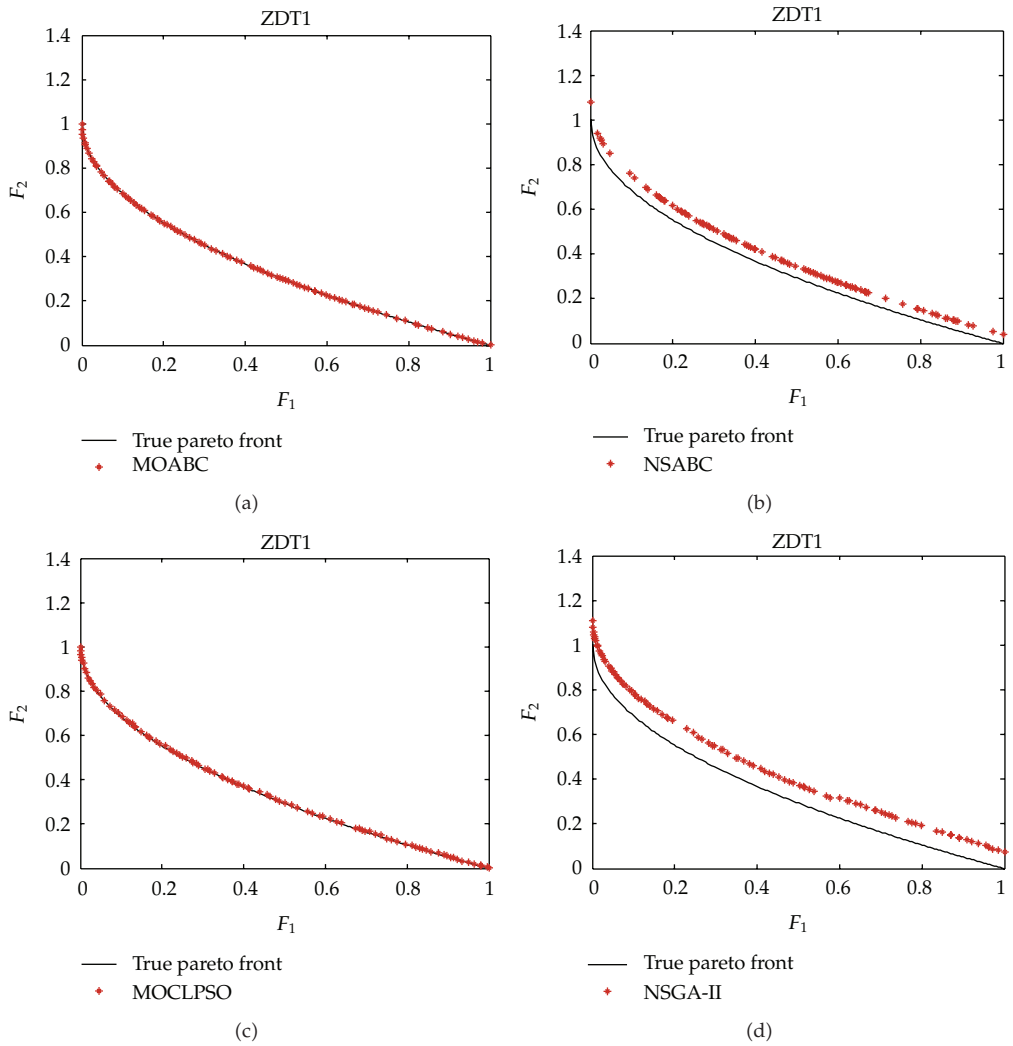

Figure 8: Pareto fronts obtained by NSABC, MOABC, MOCLPSO, and NSGA-II on ZDT1 after 20000 FEs.

These steps are repeated through a predetermined number of cycles, called maximum cycle number (MCN), or until a termination criterion is satisfied [7, 8, 15].

4. The Multiobjective ABC Algorithm

4.1. External Archive

As opposed to single-objective optimization, MOEAs usually maintain a nondominated solutions set. In multiobjective optimization, for the absence of preference information, none of the solutions can be said to be better than the others. Therefore, in our algorithm, we use an external archive to keep a historical record of the nondominated vectors found along the search process. This technique is used in many MOEAs [5, 16].

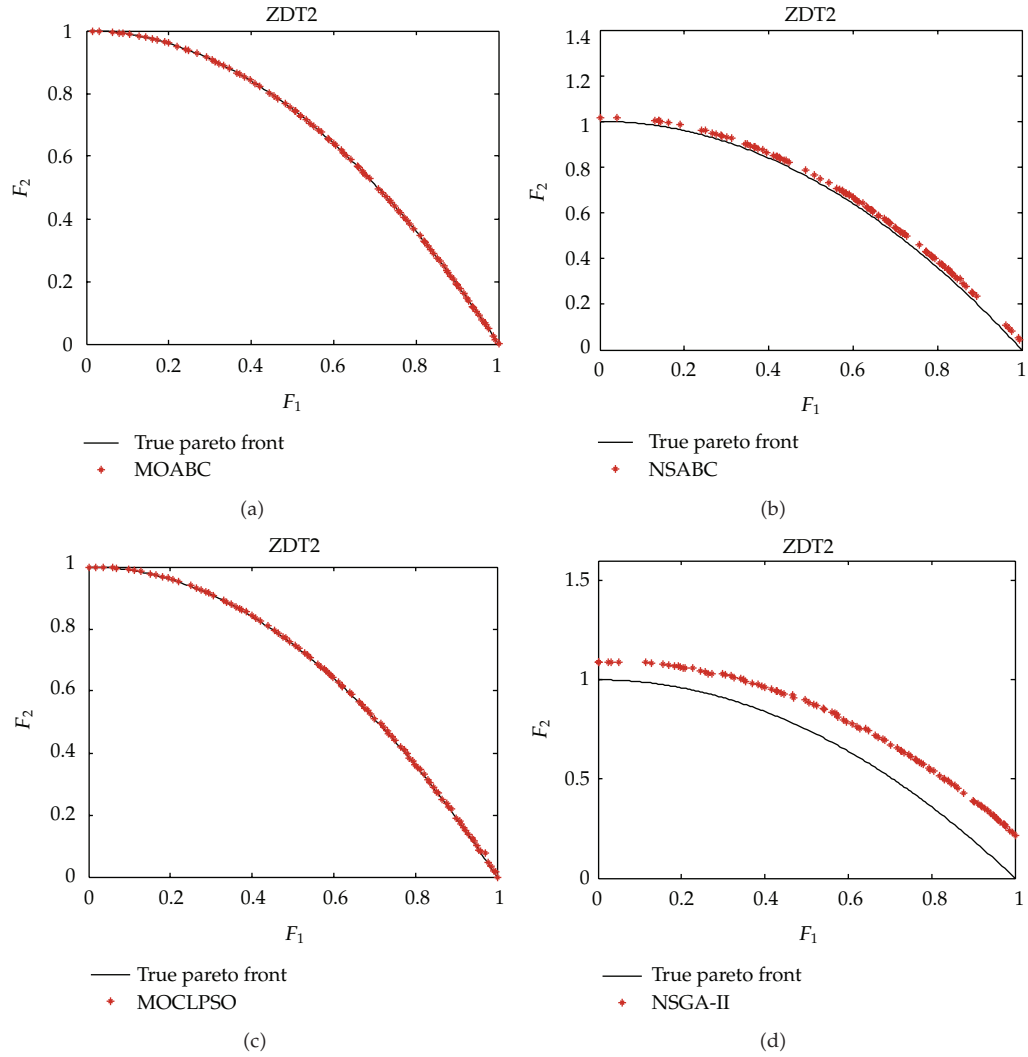

Figure 9: Pareto fronts obtained by NSABC, MOABC, MOCLPSO, and NSGA-II on ZDT2 after 20000 FEs.

Table 1: Comparison of performance on SCH after 10000 FEs.

SCH		NSABC	MOABC	MOCLPSO	NSGA-II
Converge metric	Average	$1.2621e-004$	$1.1716e-004$	$4.3657e-004$	$1.8141e-004$
	Median	$1.2791e-004$	$1.0273e-004$	$4.6393e-004$	$1.8131e-004$
	Best	$8.8590e-005$	$8.3119e-005$	$1.1484e-004$	$1.2924e-004$
	Worst	$2.1045e-004$	$2.9875e-004$	$7.5822e-004$	$2.3095e-004$
	Std	$3.8892e-005$	$6.4562e-005$	$2.1962e-004$	$3.2392e-005$
Diversity metric	Average	$3.5804e-001$	$6.6144e-001$	$8.1418e-001$	$7.5673e-001$
	Median	$3.5641e-001$	$6.6444e-001$	$8.0321e-001$	$7.5785e-001$
	Best	$3.0112e-001$	$6.4593e-001$	$7.5339e-001$	$7.2893e-001$
	Worst	$3.9650e-001$	$6.7325e-001$	$8.8260e-001$	$7.7713e-001$
	Std	$2.9056e-002$	$1.0953e-002$	$5.3376e-002$	$1.5143e-002$

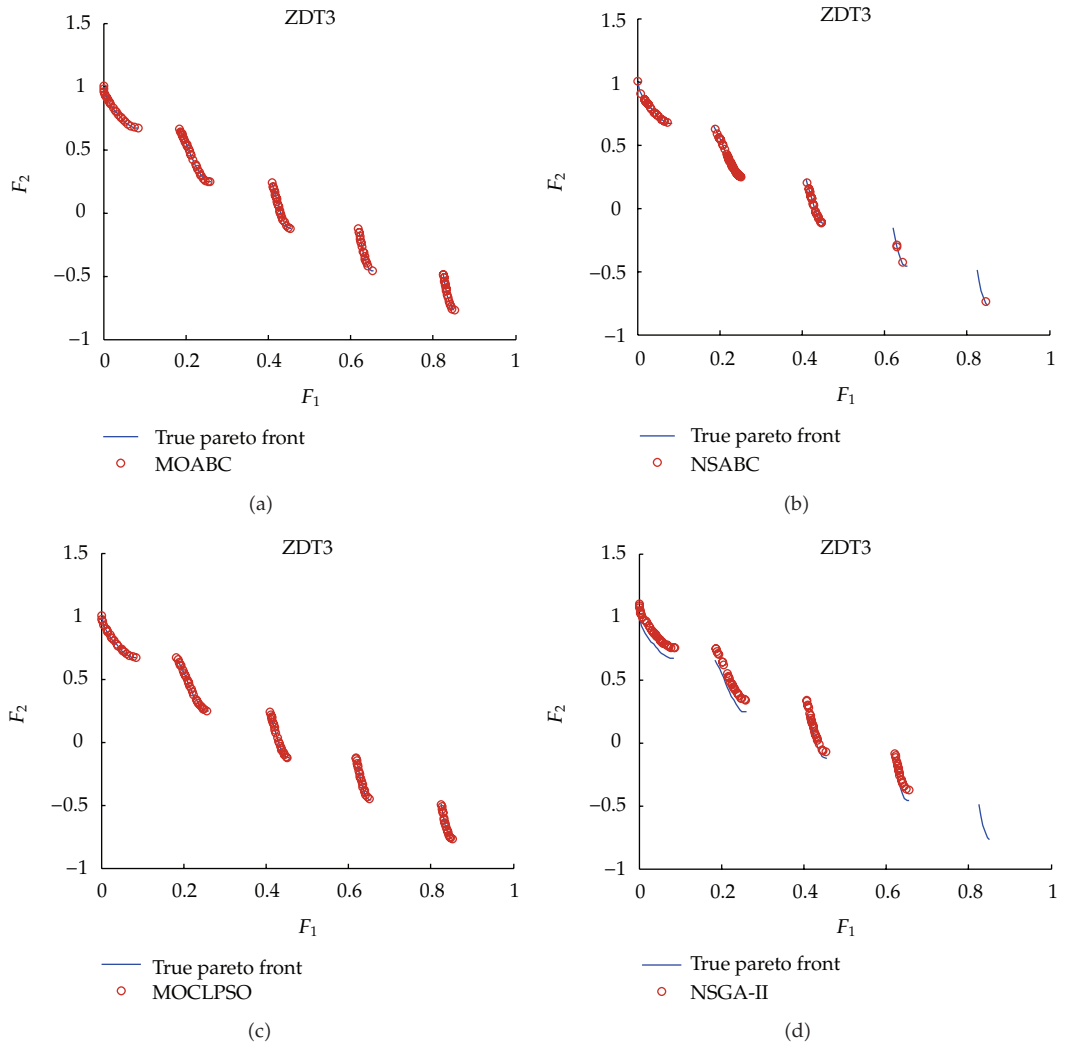

Figure 10: Pareto fronts obtained by NSABC, MOABC, MOCLPSO, and NSGA-II on ZDT3 after 20000 FEs.

Table 2: Comparison of performance on SCH after 20000 FEs.

SCH		NSABC	MOABC	MOCLPSO	NSGA-II
Converge metric	Average	1.4112$e-004$	1.2621$e-004$	3.5428$e-004$	2.1868$e-004$
	Median	1.2137$e-004$	1.2791$e-004$	2.4166$e-004$	2.0044$e-004$
	Best	8.5338$e-005$	8.8590$e-005$	1.1154$e-004$	1.2921$e-004$
	Worst	3.1097$e-004$	2.1045$e-004$	1.2674$e-003$	3.5484$e-004$
	Std	6.4495$e-005$	3.8892$e-005$	3.4346$e-004$	8.3707$e-005$
Diversity metric	Average	3.5641$e-001$	3.5804$e-001$	4.1216$e-001$	7.3525$e-001$
	Median	3.5852$e-001$	3.5641$e-001$	3.6338$e-001$	7.3941$e-001$
	Best	3.0512$e-001$	3.0112$e-001$	2.7346$e-001$	7.1568$e-001$
	Worst	4.1034$e-001$	3.9650$e-001$	6.7878$e-001$	7.5692$e-001$
	Std	3.0259$e-002$	2.9056$e-002$	1.3514$e-001$	1.4628$e-002$

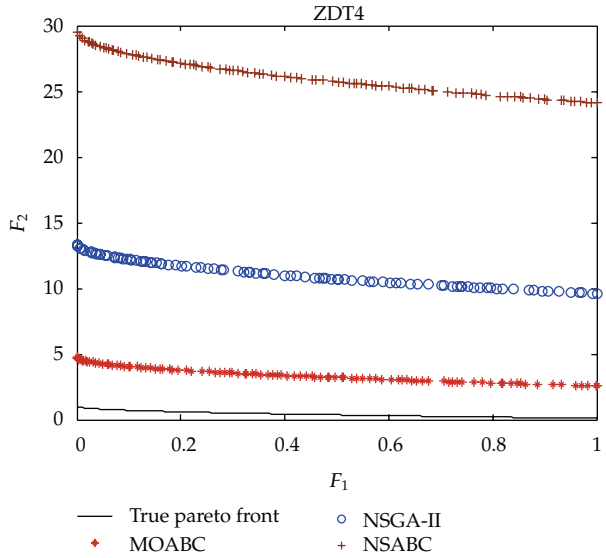

Figure 11: Pareto fronts obtained by NSABC, MOABC, and NSGA-II on ZDT4 after 20000 FEs.

Table 3: Comparison of performance on FON after 10000 FEs.

FON		NSABC	MOABC	MOCLPSO	NSGA-II
Converge metric	Average	1.8273e – 003	2.5970e – 003	2.5497e – 003	1.3845e – 003
	Median	1.7880e – 003	2.5151e – 003	2.5663e – 003	1.0898e – 003
	Best	1.4036e – 003	2.1860e – 003	2.2776e – 003	4.7267e – 004
	Worst	2.4229e – 003	3.2428e – 003	2.8616e – 003	3.1236e – 003
	Std	2.9514e – 004	3.1462e – 004	1.6583e – 004	9.5901e – 004
Diversity metric	Average	2.9810e – 001	2.9219e – 001	2.9647e – 001	7.9613e – 001
	Median	2.9468e – 001	2.9295e – 001	2.9470e – 001	8.3509e – 001
	Best	2.6703e – 001	2.7029e – 001	2.5571e – 001	6.9076e – 001
	Worst	3.4473e – 001	3.2248e – 001	3.3743e – 001	9.2519e – 001
	Std	2.6981e – 002	1.7651e – 002	2.3500e – 002	9.1073e – 002

Table 4: Comparison of performance on FON after 20000 FEs.

FON		NSABC	MOABC	MOCLPSO	NSGA-II
Converge metric	Average	1.7538e – 003	2.1794e – 003	2.2194e – 003	1.9599e – 003
	Median	1.6810e – 003	2.2196e – 003	2.3107e – 003	2.2283e – 003
	Best	1.4394e – 003	1.8691e – 003	1.5167e – 003	4.8366e – 004
	Worst	2.3287e – 003	2.4895e – 003	3.0426e – 003	3.1664e – 003
	Std	2.7467e – 004	2.1335e – 004	4.4931e – 004	9.8050e – 004
Diversity metric	Average	3.0233e – 001	2.3371e – 001	2.5764e – 001	8.2412e – 001
	Median	2.9958e – 001	2.3321e – 001	2.5815e – 001	8.3561e – 001
	Best	2.8358e – 001	1.9060e – 001	2.3411e – 001	6.8418e – 001
	Worst	3.4571e – 001	2.9052e – 001	2.9385e – 001	9.4643e – 001
	Std	1.8006e – 002	3.2232e – 002	2.0916e – 002	8.5596e – 002

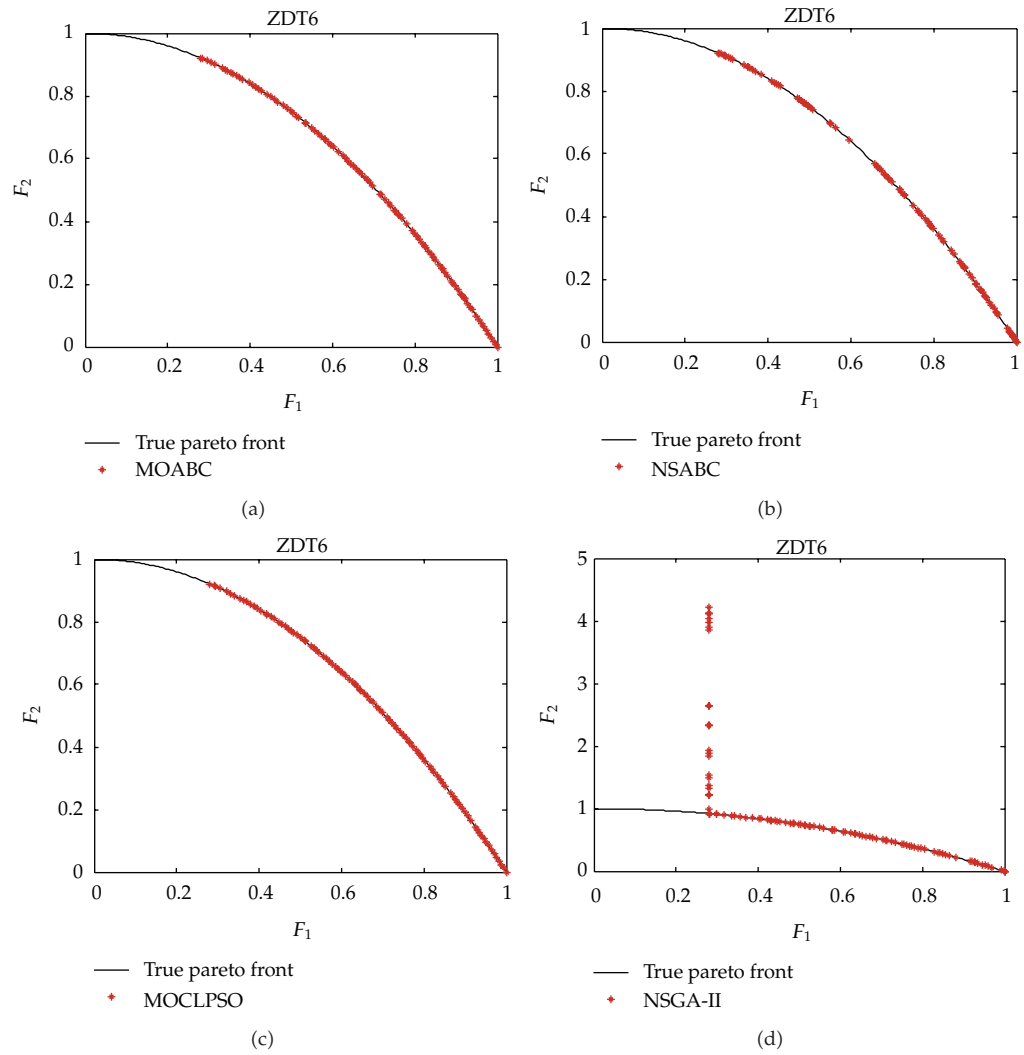

Figure 12: Pareto fronts obtained by NSABC, MOABC, MOCLPSO, and NSGA-II on ZDT6 after 20000 FEs.

Table 5: Comparison of performance on ZDT1 after 10000 FEs.

ZDT1		NSABC	MOABC	MOCLPSO	NSGA-II
Converge metric	Average	$1.0497e-001$	$2.3149e-002$	$2.4169e-003$	$2.2278e-001$
	Median	$5.0541e-002$	$2.2455e-002$	$2.4156e-003$	$1.4410e-001$
	Best	$1.1793e-002$	$1.8867e-002$	$1.7281e-003$	$7.2072e-002$
	Worst	$3.0166e-001$	$2.7309e-002$	$3.1017e-003$	$8.7348e-001$
	Std	$1.1094e-001$	$2.6057e-003$	$4.9069e-004$	$2.3806e-001$
Diversity metric	Average	$6.5022e-001$	$3.3061e-001$	$2.9773e-001$	$5.9362e-001$
	Median	$6.4527e-001$	$3.3108e-001$	$2.9207e-001$	$5.2694e-001$
	Best	$5.5856e-001$	$2.8898e-001$	$2.6434e-001$	$4.5810e-001$
	Worst	$7.6730e-001$	$3.7446e-001$	$3.5083e-001$	$9.1451e-001$
	Std	$5.8890e-002$	$2.4324e-002$	$2.4574e-002$	$1.5649e-001$

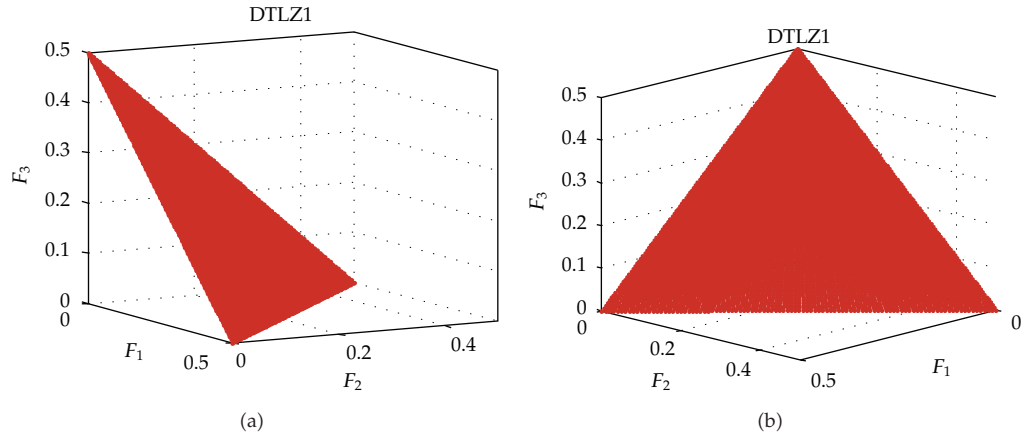

Figure 13: The true Pareto optimal front on DTLZ1.

Table 6: Comparison of performance on ZDT1 after 20000 FEs.

ZDT1		NSABC	MOABC	MOCLPSO	NSGA-II
Converge metric	Average	9.9284e – 002	1.5898e – 004	2.0862e – 003	1.1987e – 001
	Median	6.2180e – 002	1.6574e – 004	2.0433e – 003	6.6423e – 002
	Best	1.1536e – 003	9.8722e – 005	1.1930e – 003	2.6298e – 002
	Worst	5.3893e – 001	2.0130e – 004	2.8804e – 003	5.1706e – 001
	Std	1.5779e – 001	4.1578e – 005	5.1917e – 004	1.5036e – 001
Diversity metric	Average	8.0721e – 001	3.4882e – 001	3.1462e – 001	4.9501e – 001
	Median	8.0156e – 001	3.5069e – 001	3.1627e – 001	4.6744e – 001
	Best	6.0949e – 001	2.9640e – 001	2.9149e – 001	4.2585e – 001
	Worst	9.1995e – 001	4.1645e – 001	3.3882e – 001	6.5378e – 001
	Std	9.4948e – 002	3.5944e – 002	1.7857e – 002	7.4913e – 002

In the initialization phase, the external archive will be initialized. After initializing the solutions and calculating the value of every solution, they are sorted based on nondomination. We compare each solution with every other solution in the population to find which one is nondominated solution. We then put all nondominated solutions into external archive. The external archive will be updated at each generation.

4.2. Diversity

MOEA's success is due to their ability to find a set of representative Pareto optimal solutions in a single run. In order to approximate the Pareto optimal set in a single optimization run, evolutionary algorithms have to perform a multimodal search where multiple, widely different solutions are to be found. Therefore, maintaining a diverse population is crucial for the efficacy of an MOEA. ABC algorithm has been demonstrated to possess superior performance in the single-objective domain. However, NSABC that we presented as the first version multiobjective ABC algorithm cannot get satisfactory results in terms of diversity metric. So as to apply ABC algorithm to multiobjective problems, we use the comprehensive learning strategy which is inspired by comprehensive learning particle swarm optimizer (CLPSO) [9] to ensure the diversity of population.

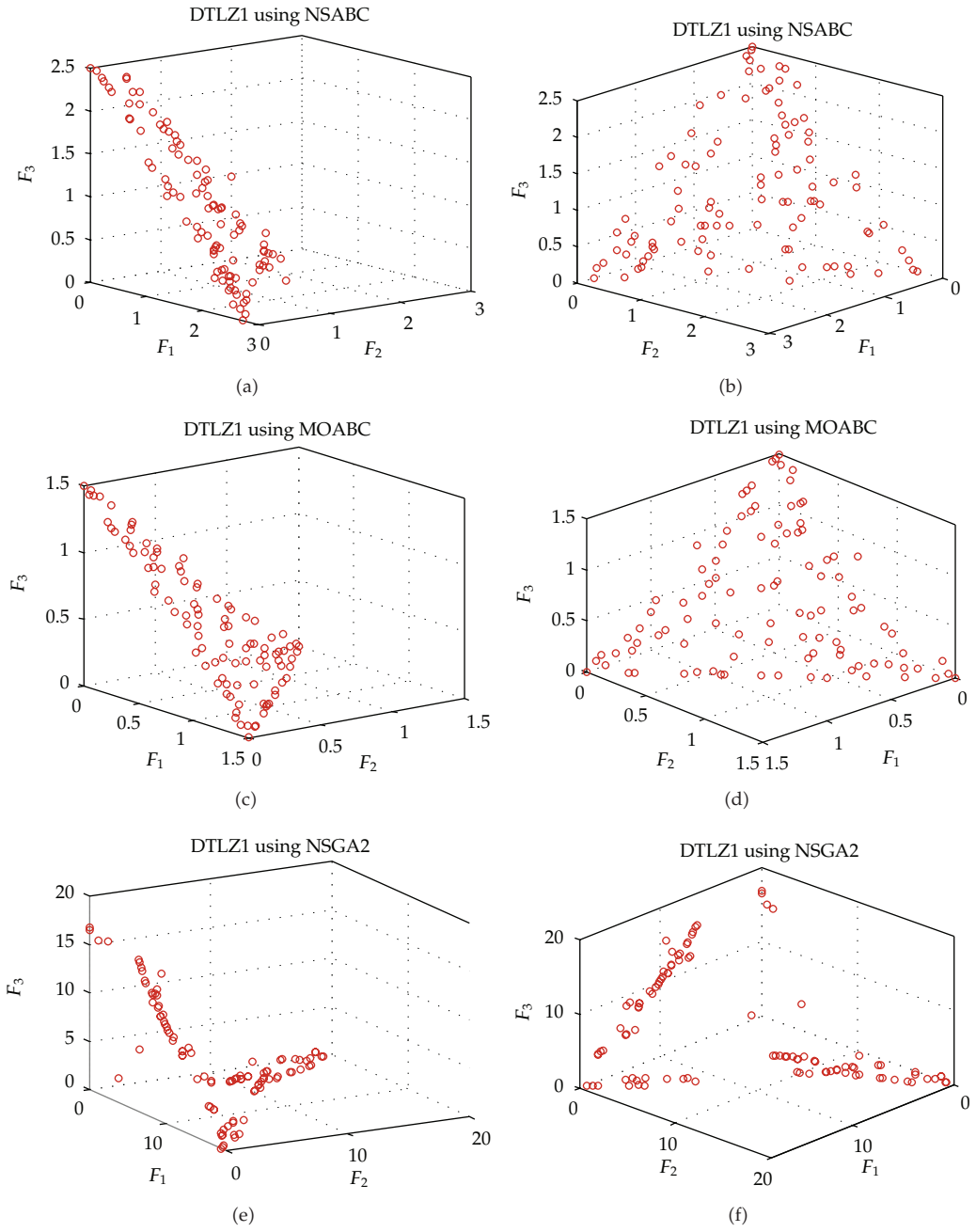

Figure 14: Pareto fronts obtained by NSABC, MOABC, and NSGA-II on DTLZ1 after 20000 FEs.

In our algorithm, all solutions in the external archive are regarded as food source positions, and all bees are regarded as onlooker bees. There do not exist employed bees and scouts. In each generation, each onlooker randomly chooses a food source from external archive, goes to the food source area, and then chooses a new food source. In original ABC algorithm, each bee finds a new food source by means of the information in the neighborhood

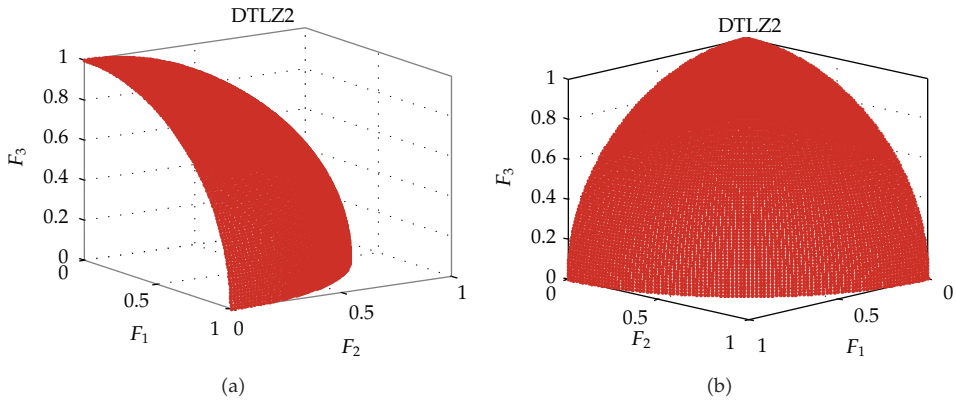

Figure 15: The true Pareto optimal front on DTLZ2.

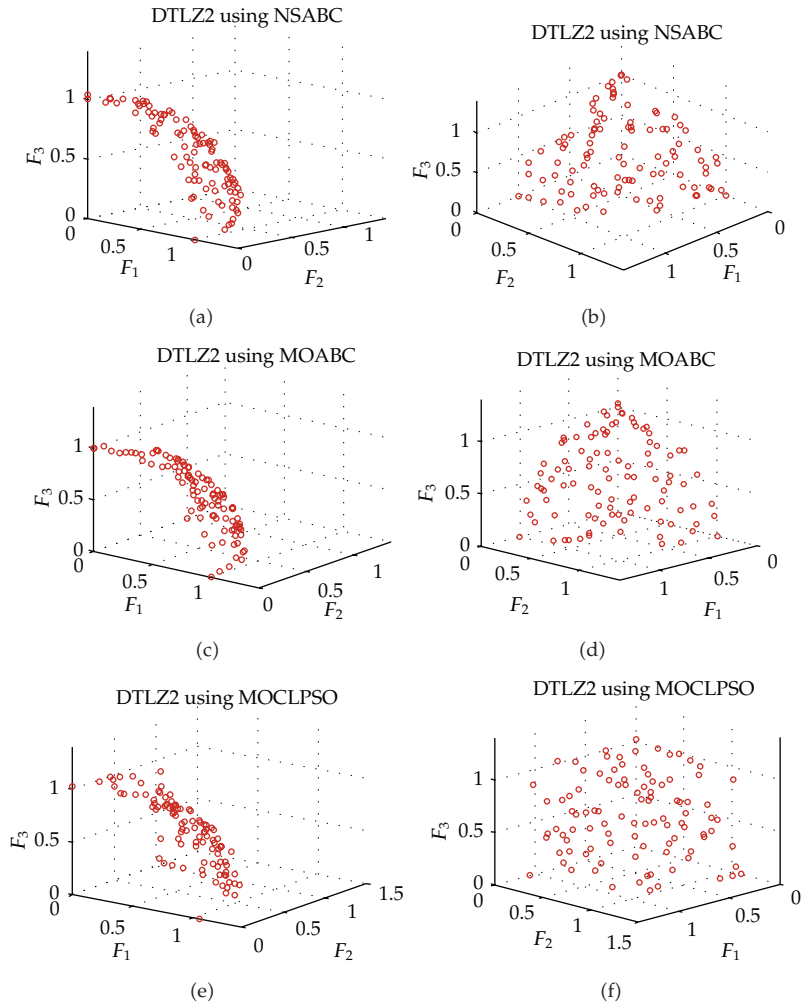

Figure 16: Continued.

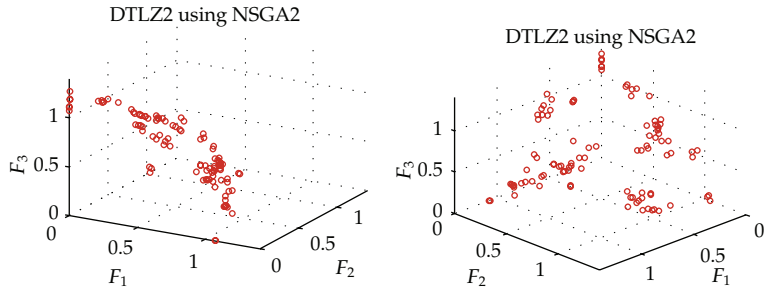

Figure 16: Pareto fronts obtained by NSABC, MOABC, MOCLPSO, and NSGA-II on DTLZ2 after 20000 FEs.

Table 7: Comparison of performance on ZDT2 after 10000 FEs.

ZDT2		NSABC	MOABC	MOCLPSO	NSGA-II
Converge metric	Average	1.9550e + 000	1.0023e – 003	9.7365e – 004	2.9476e – 001
	Median	2.0100e + 000	9.5537e – 004	1.0350e – 003	1.8105e – 001
	Best	3.8637e – 001	7.2183e – 004	3.3829e – 004	1.0668e – 001
	Worst	3.6481e + 000	1.5690e – 003	1.4223e – 003	9.8218e – 001
	Std	9.2515e – 001	2.5380e – 004	3.7382e – 004	2.6288e – 001
Diversity metric	Average	8.9986e – 001	2.9352e – 001	3.1760e – 001	7.6302e – 001
	Median	9.2174e – 001	2.9185e – 001	3.1660e – 001	7.2999e – 001
	Best	7.6513e – 001	2.5003e – 001	2.6717e – 001	4.6617e – 001
	Worst	9.6541e – 001	3.3111e – 001	3.5091e – 001	1.0542e + 000
	Std	5.8934e – 002	2.2147e – 002	2.6171e – 002	2.3567e – 001

of its current source. In our proposed MOABC algorithm, however, we use the comprehensive learning strategy. Like CLPSO, m dimensions of each individual are randomly chosen to learn from a random nondominated solution which comes from external archive. And each of the other dimensions of the individual learns from the other nondominated solutions. In our proposed NSABC algorithm, just one dimension of each individual is randomly chosen to learn from a random nondominated solution.

4.3. *Update External Archive*

As the evolution progresses, more and more new solutions enter the external archive. Considering that each new solution will be compared with every nondominated solution in the external archive to decide whether this new solution should stay in the archive, and the computational time is directly proportional to the number of comparisons, the size of external archive must be limited.

In our algorithm, each individual will find a new solution in each generation. If the new solution dominates the original individual, then the new solution is allowed to enter the external archive. On the other hand, if the new solution is dominated by the original individual, then it is denied access to the external archive. If the new solution and the original individual do not dominate each other, then we randomly choose one of them to enter the external archive. After each generation, we update external archive. We select nondominated

Table 8: Comparison of performance on ZDT2 after 20000 FEs.

ZDT2		NSABC	MOABC	MOCLPSO	NSGA-II
Converge metric	Average	8.6101e – 002	1.0592e – 004	1.0238e – 003	1.1274e – 001
	Median	7.2305e – 002	9.5653e – 005	1.0092e – 003	9.8693e – 002
	Best	4.3562e – 002	7.4842e – 005	6.2493e – 004	3.5737e – 002
	Worst	3.1215e – 001	1.4481e – 004	1.4683e – 003	3.3153e – 001
	Std	9.1252e – 001	2.5134e – 005	2.6058e – 004	8.5585e – 002
Diversity metric	Average	8.4325e – 001	3.4310e – 001	3.1718e – 001	5.5863e – 001
	Median	8.5472e – 001	3.4839e – 001	2.9621e – 001	5.1789e – 001
	Best	7.3456e – 001	2.9056e – 001	3.7037e – 001	4.4479e – 001
	Worst	9.3123e – 001	3.8678e – 001	2.0841e – 002	1.0050e + 000
	Std	7.0645e – 002	3.1759e – 002	3.1718e – 001	1.6478e – 001

Table 9: Comparison of performance on ZDT3 after 10000 FEs.

ZDT3		NSABC	MOABC	MOCLPSO	NSGA-II
Converge metric	Average	6.1941e – 003	1.5551e – 003	2.5320e – 003	2.1662e – 001
	Median	1.2600e – 003	1.5186e – 003	2.4152e – 003	1.5563e – 001
	Best	1.0374e – 003	1.2901e – 003	1.6893e – 003	7.9710e – 002
	Worst	3.8055e – 002	2.1061e – 003	3.1702e – 003	6.3936e – 001
	Std	1.1842e – 002	2.2384e – 004	5.3581e – 004	1.7221e – 001
Diversity metric	Average	9.0080e – 001	6.6057e – 001	7.2017e – 001	7.0351e – 001
	Median	9.0374e – 001	6.6325e – 001	7.2578e – 001	6.9255e – 001
	Best	8.0100e – 001	6.3194e – 001	6.5412e – 001	6.5156e – 001
	Worst	1.0213e + 000	6.8535e – 001	7.8368e – 001	8.1501e – 001
	Std	6.9121e – 002	1.8462e – 002	3.3709e – 002	4.6110e – 002

solutions from the archive and keep them in the archive. If the number of nondominated solutions exceeds the allocated archive size, then crowding distance [4] is applied to remove the crowded members and to maintain uniform distribution among the archive members.

4.4. Crowding Distance

Crowding distance is used to estimate the density of solutions in the external archive. Usually, the perimeter of the cuboid formed by using the nearest neighbors as the vertices is called crowding distance. In Figure 2, the crowding distance of the ith solution is the average side length of the cuboid (shown with a dashed box) [4].

The crowding distance computation requires sorting the population in the external archive according to each objective function value in ascending order of magnitude. Thereafter, for each objective function, the boundary solutions (solutions with smallest and largest function values) are assigned as an infinite distance value. All other intermediate solutions are assigned a distance value equal to the absolute normalized difference in the function values of two adjacent solutions. This 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. Each objective function is normalized before calculating the crowding distance [4].

Table 10: Comparison of performance on ZDT3 after 20000 FEs.

ZDT3		NSABC	MOABC	MOCLPSO	NSGA-II
Converge metric	Average	1.5543$e-004$	3.5783$e-004$	1.0331$e-003$	6.9862$e-002$
	Median	1.1933$e-004$	3.6670$e-004$	1.0744$e-003$	2.9758$e-002$
	Best	7.5450$e-005$	2.6324$e-004$	7.3706$e-004$	1.6795$e-002$
	Worst	4.0513$e-004$	4.5799$e-004$	1.1982$e-003$	3.0308$e-001$
	Std	9.9259$e-005$	6.2428$e-005$	1.6711$e-004$	9.2471$e-002$
Diversity metric	Average	1.0943$e+000$	6.2668$e-001$	6.3686$e-001$	6.8500$e-001$
	Median	1.1146$e+000$	6.2607$e-001$	6.3902$e-001$	6.7755$e-001$
	Best	9.4478$e-001$	6.0343$e-001$	6.0709$e-001$	6.4228$e-001$
	Worst	1.1992$e+000$	6.5011$e-001$	6.5457$e-001$	7.9273$e-001$
	Std	8.7036$e-002$	1.5613$e-002$	1.3501$e-002$	4.1238$e-002$

Table 11: Comparison of performance on ZDT4 after 10000 FEs.

ZDT4		NSABC	MOABC	MOCLPSO	NSGA-II
Converge metric	Average	2.1278$e+001$	5.3904$e+000$	N/A	3.2868$e+001$
	Median	1.3447$e+001$	5.2482$e+000$	N/A	3.3219$e+001$
	Best	7.0411$e+000$	3.2012$e+000$	N/A	2.9451$e+001$
	Worst	5.4023$e+001$	8.3100$e+000$	N/A	3.6560$e+001$
	Std	1.7517$e+001$	1.6624$e+000$	N/A	2.1154$e+000$
Diversity metric	Average	9.2818$e-001$	9.0817$e-001$	N/A	9.8813$e-001$
	Median	9.2668$e-001$	8.9116$e-001$	N/A	9.9170$e-001$
	Best	8.9696$e-001$	8.3111$e-001$	N/A	9.6282$e-001$
	Worst	9.6675$e-001$	9.9888$e-001$	N/A	1.0172$e+000$
	Std	2.5872$e-002$	6.5026$e-002$	N/A	1.6292$e-002$

4.5. The Multiobjective ABC Algorithm

The ABC algorithm is very simple when compared to the existing swarm-based algorithms. Therefore, we develop it to deal with multiobjective optimization problems. The main steps of the MOABC algorithm are shown in Algorithm 2.

In the initialization phase, we evaluate the fitness of the initial food source positions and sort them based on nondomination. Then we select nondominated solutions and store them in the external archive EA. This is the initialization of the external archive.

In the onlooker bees' phase, we use comprehensive learning strategy to produce new solutions v_i. For each bee x_i, it randomly chooses m dimensions and learns from a nondominated solution which is randomly selected from EA. And the other dimensions learn from the other nondominated solutions. The new solution is produced by using the following expression:

$$v_{i,f(m)} = x_{i,f(m)} + \phi(m)\left(\mathrm{EA}_{k,f(m)} - x_{i,f(m)}\right), \tag{4.1}$$

where $k \in (1, 2, \ldots, p)$ is randomly chosen index and p is the number of solutions in the EA. $f(m)$ is the first m integers of a random permutation of the integers $1:n$, and $f(m)$ defines which x_i's dimensions should learn from EA_k. As opposed to ϕ_{ij} in original ABC algorithm,

Table 12: Comparison of performance on ZDT4 after 20000 FEs.

ZDT4		NSABC	MOABC	MOCLPSO	NSGA-II
Converge metric	Average	2.8242e + 001	3.9615e + 000	N/A	1.5026e + 001
	Median	2.8297e + 001	3.7351e + 000	N/A	1.3504e + 001
	Best	5.3764e + 000	3.9397e − 001	N/A	9.2747e + 000
	Worst	5.5253e + 001	7.6236e + 000	N/A	2.8434e + 001
	Std	1.4687e + 001	2.1985e + 000	N/A	5.3418e + 000
Diversity metric	Average	9.3404e − 001	8.2135e − 001	N/A	9.4884e − 001
	Median	9.4428e − 001	8.5428e − 001	N/A	9.4495e − 001
	Best	8.6520e − 001	5.8305e − 001	N/A	9.1134e − 001
	Worst	9.5992e − 001	8.8787e − 001	N/A	1.0004e + 000
	Std	2.8531e − 002	9.0785e − 002	N/A	2.4879e − 002

Table 13: Comparison of performance on ZDT6 after 10000 FEs.

ZDT6		NSABC	MOABC	MOCLPSO	NSGA-II
Converge metric	Average	9.1927e − 001	3.9988e − 003	1.5839e − 002	3.3642e + 000
	Median	1.8809e − 005	2.8035e − 005	8.1531e − 003	3.9688e + 000
	Best	1.1153e − 005	1.9642e − 005	2.3569e − 005	6.9186e − 001
	Worst	3.2480e + 000	2.6604e − 002	5.3588e − 002	4.7179e + 000
	Std	1.4822e + 000	8.9540e − 003	1.9448e − 002	1.4822e + 000
Diversity metric	Average	9.0099e − 001	6.0121e − 001	8.1936e − 001	1.1219e + 000
	Median	9.5394e − 001	4.7318e − 001	7.5949e − 001	1.1187e + 000
	Best	6.9802e − 001	4.3742e − 001	3.9620e − 001	1.0231e + 000
	Worst	1.0392e + 000	1.2171e + 000	1.3417e + 000	1.2849e + 000
	Std	1.1021e − 001	2.7989e − 001	4.2953e − 001	8.8914e − 002

$\phi(m)$ produce m random numbers which are all between $[0,2]$. And the m random numbers correspond to the m dimensions above. This modification makes the potential search space around EA_k. The potential search spaces of MOABC on one dimension are plotted as a line in Figure 3. Each remaining dimension learns from the other nondominated solutions by using

$$v_{ij} = x_{ij} + \phi_{ij}\left(\mathrm{EA}_{lj} - x_{ij}\right), \tag{4.2}$$

where $l \neq k$, $j \in (1,2,\ldots,p)$ and $j \notin f(m)$. For NSABC algorithm, each bee x_i randomly chooses a dimension and learns from a nondominated solution which is randomly selected from EA. The new solution v_i is produced by just using expression (4.2). After producing new solution, we calculate the fitness and apply greedy selection mechanism to decide which solution enters EA. The selection mechanism is shown in Algorithm 3.

In nondominated sorting phase, after a generation, the solutions in the EA are sorted based on nondomination, and we keep the nondomination solutions of them staying in the

Table 14: Comparison of performance on ZDT6 after 20000 FEs.

ZDT6		NSABC	MOABC	MOCLPSO	NSGA-II
Converge metric	Average	4.3038$e-005$	6.7609$e-004$	1.4469$e-002$	4.6871$e-001$
	Median	4.2428$e-005$	2.4510$e-005$	2.4791$e-005$	4.1021$e-001$
	Best	3.8692$e-006$	2.1576$e-005$	2.2898$e-005$	7.1529$e-002$
	Worst	4.9351$e-003$	6.5408$e-003$	9.1647$e-002$	1.2285$e+000$
	Std	2.9946$e-004$	2.0606$e-003$	2.9694$e-002$	3.7985$e-001$
Diversity metric	Average	7.7320$e-001$	4.9926$e-001$	6.3819$e-001$	1.1519$e+000$
	Median	7.3399$e-001$	4.5840$e-001$	4.1180$e-001$	1.1497$e+000$
	Best	6.4696$e-001$	4.4708$e-001$	3.5906$e-001$	1.0412$e+000$
	Worst	1.0856$e+000$	8.1645$e-001$	1.2848$e+000$	1.2939$e+000$
	Std	1.2741$e-001$	1.1284$e-001$	3.9246$e-001$	8.1261$e-002$

Table 15: Comparison of performance on DTLZ1 after 10000 FEs.

DTLZ1		NSABC	MOABC	MOCLPSO	NSGA-II
Converge metric	Average	5.6686$e+000$	6.9637$e+000$	N/A	1.4780$e+001$
	Median	4.3841$e+000$	7.4922$e+000$	N/A	1.4598$e+001$
	Best	1.0581$e+000$	1.5215$e+000$	N/A	9.8158$e+000$
	Worst	9.7176$e+000$	1.1092$e+001$	N/A	1.8511$e+001$
	Std	3.5367$e+000$	4.1292$e+000$	N/A	2.7797$e+000$
Diversity metric	Average	4.6157$e-001$	4.5170$e-001$	N/A	5.5704$e-001$
	Median	4.6061$e-001$	4.4991$e-001$	N/A	5.4700$e-001$
	Best	3.9987$e-001$	4.0565$e-001$	N/A	4.8393$e-001$
	Worst	5.1209$e-001$	4.9551$e-001$	N/A	6.5260$e-001$
	Std	3.3377$e-002$	3.1004$e-002$	N/A	5.1337$e-002$

EA. If the number of nondominated solutions exceeds the allocated size of EA, we use crowding distance to remove the crowded members. Crowding distance algorithm is seen in [4].

5. Experiments

In the following, we will first describe the benchmark functions used to compare the performance of MOABC with NSABC, NSGA-II, and MOCLPSO. And then we will introduce performance measures. For every algorithm, we will give the parameter settings. At last, we will present simulation results for benchmark functions.

5.1. Benchmark Functions

In order to illustrate the performance of the proposed MOABC algorithm, we used several well-known test problems SCH, FON, ZDT1 to ZDT4, and ZDT6 as the two-objective test functions. And we used Deb-Thiele-Laumanns-Zitzler (DTLZ) problem family as the three-objective test functions.

Table 16: Comparison of performance on DTLZ1 after 20000 FEs.

DTLZ1		NSABC	MOABC	MOCLPSO	NSGA-II
Converge metric	Average	4.5813e + 000	2.3609e + 000	N/A	1.6197e + 001
	Median	4.2016e + 000	1.3339e + 000	N/A	1.6815e + 001
	Best	1.1025e + 000	6.9640e − 001	N/A	9.7891e + 000
	Worst	1.0148e + 001	7.9295e + 000	N/A	2.2004e + 001
	Std	2.7762e + 000	2.2890e + 000	N/A	3.7379e + 000
Diversity metric	Average	4.5833e − 001	4.6144e − 001	N/A	5.7586e − 001
	Median	4.6618e − 001	4.4855e − 001	N/A	5.9588e − 001
	Best	4.0286e − 001	4.0314e − 001	N/A	4.0561e − 001
	Worst	4.8518e − 001	5.4159e − 001	N/A	6.8804e − 001
	Std	2.6031e − 002	5.0967e − 002	N/A	8.7688e − 002

Table 17: Comparison of performance on DTLZ2 after 10000 FEs.

DTLZ2		NSABC	MOABC	MOCLPSO	NSGA-II
Converge metric	Average	2.1341e − 002	7.6836e − 003	1.0976e − 001	8.7363e − 002
	Median	2.1827e − 002	6.7337e − 003	1.1294e − 001	8.9419e − 002
	Best	1.8141e − 002	4.9233e − 003	9.4400e − 002	6.0319e − 002
	Worst	2.4931e − 002	1.2708e − 002	1.1705e − 001	1.1310e − 001
	Std	2.3165e − 003	2.4500e − 003	7.4233e − 003	1.6418e − 002
Diversity metric	Average	4.1414e − 001	4.1177e − 001	4.1629e − 001	6.0903e − 001
	Median	4.1040e − 001	4.1107e − 001	4.1578e − 001	5.0881e − 001
	Best	3.9349e − 001	3.5942e − 001	3.7554e − 001	3.7879e − 001
	Worst	4.7171e − 001	4.7267e − 001	4.7455e − 001	7.3358e − 001
	Std	2.1464e − 002	3.1091e − 002	3.0363e − 002	1.8498e − 002

SCH

Although simple, the most studied single-variable test problem is Schaffer's two-objective problem [17, 18]

$$\text{SCH}: \begin{cases} \text{Minimize} & f_1(x) = x^2, \\ \text{Minimize} & f_2(x) = (x-2)^2 \\ & -10^3 \le x \le 10^3. \end{cases} \tag{5.1}$$

This problem has Pareto optimal solution $x^* \in [0, 2]$, and the Pareto optimal set is a convex set:

$$f_2^* = \left(\sqrt{f_1^*} - 2\right)^2. \tag{5.2}$$

Table 18: Comparison of performance on DTLZ2 after 20000 FEs.

DTLZ2		NSABC	MOABC	MOCLPSO	NSGA-II
Converge metric	Average	1.6461$e-002$	2.9095$e-003$	1.0504$e-001$	8.8402$e-002$
	Median	1.6400$e-002$	2.8372$e-003$	1.0476$e-001$	9.4425$e-002$
	Best	1.3958$e-002$	2.6826$e-003$	9.5225$e-002$	6.6176$e-002$
	Worst	1.8900$e-002$	3.3161$e-003$	1.1362$e-001$	1.0847$e-001$
	Std	1.6644$e-003$	2.3986$e-004$	5.5136$e-003$	1.3819$e-002$
Diversity metric	Average	4.1713$e-001$	4.1041$e-001$	4.1786$e-001$	6.2601$e-001$
	Median	4.1918$e-001$	4.0989$e-001$	4.1565$e-001$	6.1671$e-001$
	Best	3.6375$e-001$	3.9217$e-001$	3.8277$e-001$	3.8379$e-001$
	Worst	4.8046$e-001$	4.3465$e-001$	4.7103$e-001$	7.3020$e-001$
	Std	3.6245$e-002$	1.3692$e-002$	2.7804$e-002$	4.3631$e-002$

Table 19: Comparison of performance on DTLZ3 after 10000 FEs.

DTLZ3		NSABC	MOABC	MOCLPSO	NSGA-II
Converge metric	Average	1.0604$e+002$	5.2341$e+001$	N/A	1.0995$e+002$
	Median	1.0317$e+002$	5.0666$e+001$	N/A	1.0583$e+002$
	Best	5.0475$e+001$	2.2050$e+001$	N/A	6.3606$e+001$
	Worst	1.5592$e+002$	8.7997$e+001$	N/A	1.5401$e+002$
	Std	3.0211$e+001$	1.9430$e+001$	N/A	2.6229$e+001$
Diversity metric	Average	4.3775$e-001$	4.4570$e-001$	N/A	9.6799$e-001$
	Median	4.3628$e-001$	4.4365$e-001$	N/A	9.0815$e-001$
	Best	3.9653$e-001$	4.2066$e-001$	N/A	5.7383$e-001$
	Worst	4.6588$e-001$	4.9020$e-001$	N/A	1.3928$e+000$
	Std	2.2679$e-002$	2.0921$e-002$	N/A	2.6257$e-001$

FON

Schaffer and Fonseca and Fleming used a two-objective optimization problem (FON) [18, 19] having n variables:

$$\text{FON}: \begin{cases} \text{Minmize} \quad f_1(x) = 1 - \exp\left(-\sum_{i=1}^{n}\left(x_i - \frac{1}{\sqrt{n}}\right)^2\right), \\ \text{Minmize} \quad f_2(x) = 1 - \exp\left(-\sum_{i=1}^{n}\left(x_i + \frac{1}{\sqrt{n}}\right)^2\right) \\ \qquad\qquad -4 \le x_i \le 4 \quad i = 1, 2, \ldots, n. \end{cases} \tag{5.3}$$

The Pareto optimal solution to this problem is $x_i^* = [-1/\sqrt{n}, 1/\sqrt{n}]$ for $i = 1, 2, \ldots, n$. These solutions also satisfy the following relationship between the two function values:

$$f_2^* = 1 - \exp\left\{-\left[2 - \sqrt{-\ln(1 - f_1^*)}\right]^2\right\} \tag{5.4}$$

Table 20: Comparison of performance on DTLZ3 after 20000 FEs.

DTLZ3		NSABC	MOABC	MOCLPSO	NSGA-II
Converge metric	Average	9.3970e + 001	2.9514e + 001	N/A	8.9755e + 001
	Median	8.9480e + 001	1.9201e + 001	N/A	8.7459e + 001
	Best	5.8510e + 001	1.5806e + 001	N/A	5.8992e + 001
	Worst	1.3147e + 002	6.5907e + 001	N/A	1.2936e + 002
	Std	2.3911e + 001	1.8973e + 001	N/A	2.2698e + 001
Diversity metric	Average	4.3399e − 001	4.2490e − 001	N/A	9.0585e − 001
	Median	4.3223e − 001	4.1868e − 001	N/A	8.7254e − 001
	Best	4.0358e − 001	3.7111e − 001	N/A	7.2752e − 001
	Worst	4.8145e − 001	4.7686e − 001	N/A	1.2918e + 000
	Std	2.5498e − 002	3.5592e − 002	N/A	1.7067e − 001

Table 21: Comparison of performance on DTLZ6 after 10000 FEs.

DTLZ6		NSABC	MOABC	MOCLPSO	NSGA-II
Converge metric	Average	5.5914e − 001	2.9901e − 002	2.3412e − 002	6.6160e − 001
	Median	2.9461e − 002	2.7952e − 002	2.5935e − 002	2.8640e − 001
	Best	2.0189e − 002	1.9106e − 002	5.6145e − 003	1.6724e − 001
	Worst	4.8909e + 000	3.7997e − 002	3.4578e − 002	3.1497e + 000
	Std	1.5271e + 000	6.7314e − 003	8.1996e − 003	9.2366e − 001
Diversity metric	Average	5.4587e − 001	5.4715e − 001	5.2084e − 001	5.7305e − 001
	Median	5.4631e − 001	5.5144e − 001	4.9934e − 001	5.5839e − 001
	Best	4.7349e − 001	5.0172e − 001	4.6254e − 001	4.5813e − 001
	Worst	6.3240e − 001	6.0581e − 001	6.2609e − 001	7.0776e − 001
	Std	4.9975e − 002	2.9597e − 002	5.1694e − 002	7.2624e − 002

in the range $0 \le f_1^* \le 1-\exp(-4)$. The interesting aspect is that the search space in the objective space and the Pareto optimal function values do not depend on the dimensionality (the parameter n) of the problem. In our paper, we set $n = 3$. And the optimal solutions are $x_1 = x_2 = x_3 \in [-1/\sqrt{3}, 1/\sqrt{3}]$.

ZDT1

This is a 30-variable ($n = 30$) problem having a convex Pareto optimal set. The functions used are as follows:

$$\text{ZDT1}: \begin{cases} \text{Minimize} & f_1(x) = x_1, \\ \text{Minimize} & f_2(x) = g(x)\left[1 - \sqrt{\dfrac{x_1}{g(x)}}\right], \\ & g(x) = 1 + \dfrac{9\left(\sum_{i=2}^{n} x_i\right)}{n-1}. \end{cases} \tag{5.5}$$

All variables lie in the range $[0, 1]$. The Pareto optimal region corresponds to $0 \le x_1^* \le 1$ and $x_1^* = 0$ for $i = 2, 3, \ldots, 30$ [20].

Table 22: Comparison of performance on DTLZ6 after 20000 FEs.

DTLZ6		NSABC	MOABC	MOCLPSO	NSGA-II
Converge metric	Average	1.4560$e-001$	2.2293$e-002$	3.0709$e-002$	4.1934$e-001$
	Median	1.1240$e-001$	2.2006$e-002$	3.1585$e-002$	2.7839$e-001$
	Best	5.4431$e-002$	1.8476$e-002$	2.1425$e-002$	1.2079$e-001$
	Worst	4.6437$e-001$	2.7389$e-002$	3.9060$e-002$	1.9577$e+000$
	Std	1.1962$e-001$	2.8880$e-003$	4.9988$e-003$	5.4684$e-001$
Diversity metric	Average	5.1947$e-001$	5.2313$e-001$	5.2114$e-001$	5.8184$e-001$
	Median	5.1032$e-001$	5.2860$e-001$	5.1470$e-001$	5.9360$e-001$
	Best	4.9349$e-001$	4.7949$e-001$	4.9429$e-001$	4.9585$e-001$
	Worst	5.4816$e-001$	5.8122$e-001$	5.8158$e-001$	6.7556$e-001$
	Std	1.9948$e-002$	3.2645$e-002$	2.9098$e-002$	5.8605$e-002$

ZDT2

This is also an $n = 30$ variable problem having a nonconvex Pareto optimal set:

$$\text{ZDT2}: \begin{cases} \text{Minimize} & f_1(x) = x_1, \\ \text{Minimize} & f_2(x) = g(x)\left[1 - \left(\dfrac{x_1}{g(x)}\right)^2\right], \\ & g(x) = 1 + \dfrac{9\left(\sum_{i=2}^{n} x_i\right)}{n-1}. \end{cases} \tag{5.6}$$

All variables lie in the range $[0,1]$. The Pareto optimal region corresponds to $0 \le x_1^* \le 1$ and $x_1^* = 0$ for $i = 2,3,\ldots,30$ [20].

ZDT3

This is an $n = 30$ variable problem having a number of disconnected Pareto optimal fronts:

$$\text{ZDT3}: \begin{cases} \text{Minimize} & f_1(x) = x_1, \\ \text{Minimize} & f_2(x) = g(x)\left[1 - \sqrt{\dfrac{x_1}{g(x)}} - \dfrac{x_1}{g(x)}\sin(10\pi x_1)\right], \\ & g(x) = 1 + \dfrac{9\left(\sum_{i=2}^{n} x_i\right)}{n-1}. \end{cases} \tag{5.7}$$

All variables lie in the range $[0,1]$. The Pareto optimal region corresponds to $x_1^* = 0$ for $i = 2,3,\ldots,30$, and hence not all points satisfying $0 \le x_1^* \le 1$ lie on the Pareto optimal front [20].

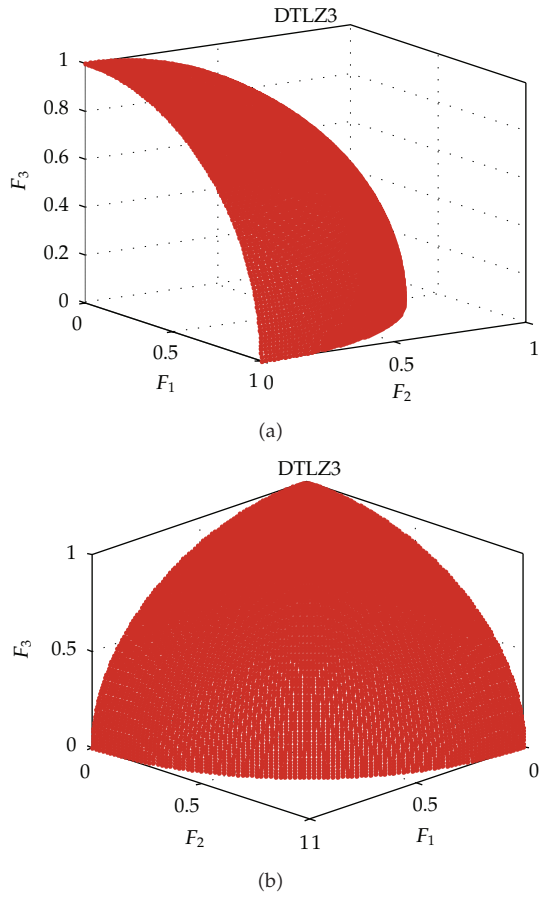

Figure 17: The true Pareto optimal front on DTLZ3.

ZDT4

This is an $n = 10$ variable problem having a convex Pareto optimal set:

$$\text{ZDT4}: \begin{cases} \text{Minimize} & f_1(x) = x_1, \\ \text{Minimize} & f_2(x) = g(x)\left[1 - \sqrt{\dfrac{x_1}{g(x)}}\right], \\ & g(x) = 1 + 10(n-1) + \displaystyle\sum_{i=2}^{n}\left[x_i^2 - 10\cos(4\pi x_i)\right]. \end{cases} \tag{5.8}$$

The variable x_1 lies in the range $[0, 1]$, but all others in the range $[-5, 5]$. The Pareto optimal region corresponds to $0 \le x_1^* \le 1$ and $x_1^* = 0$ for $i = 2, 3, \ldots, 10$ [20].

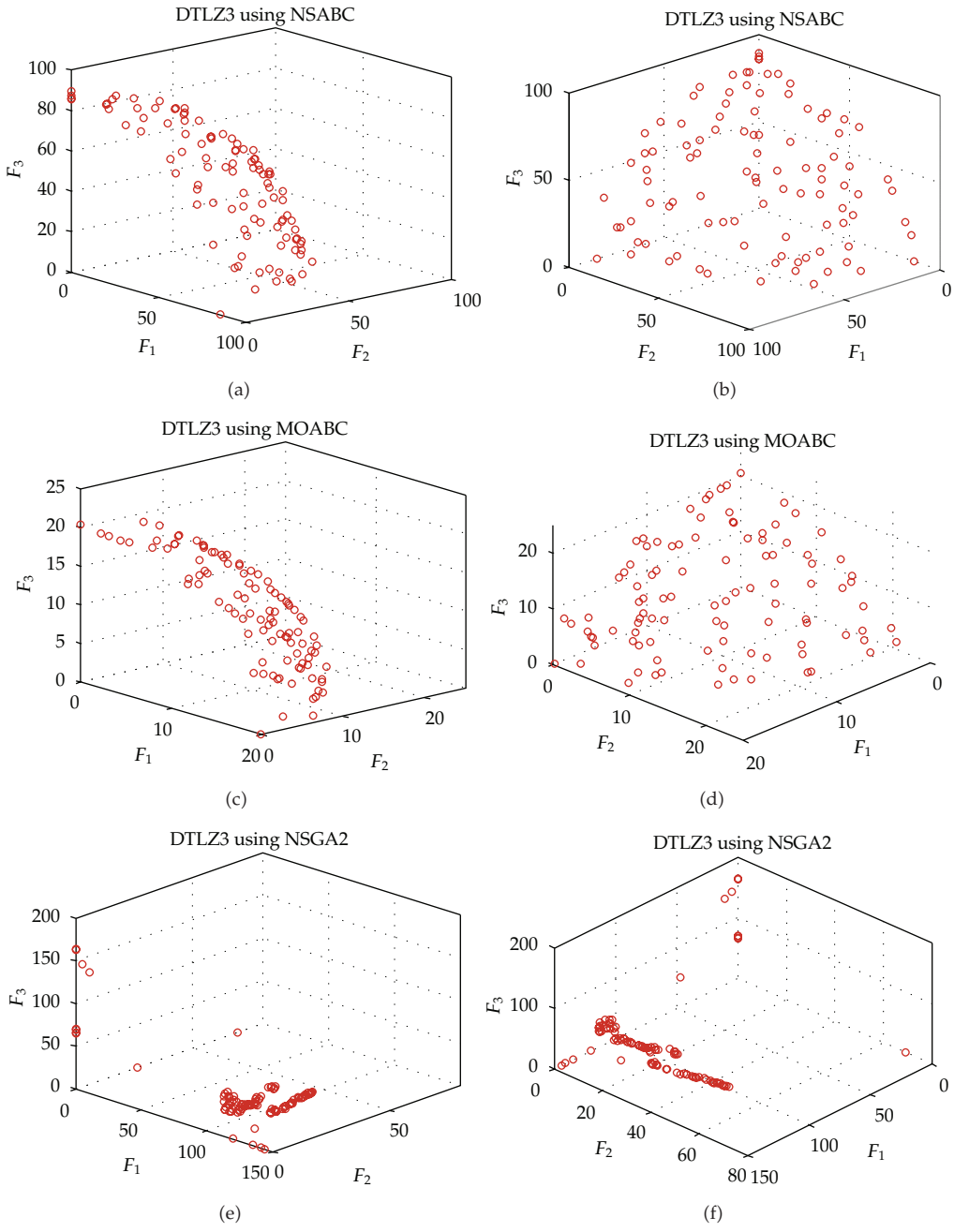

Figure 18: Pareto fronts obtained by NSABC, MOABC, and NSGA-II on DTLZ3 after 20000 FEs.

ZDT6

This is a 10-variable problem having a nonconvex Pareto optimal set. Moreover, the density of solutions across the Pareto optimal region is nonuniform, and the density towards the Pareto

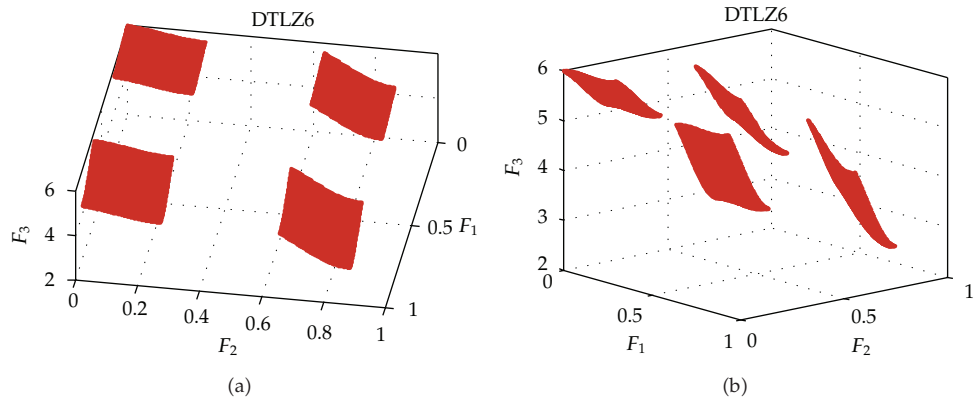

Figure 19: The true Pareto optimal front on DTLZ6.

optimal front is also thin:

$$\text{ZDT6}: \begin{cases} \text{Minimize} & f_1(x) = 1 - \exp(-4x_1)\sin^6(6\pi x_1), \\ \text{Minimize} & f_2(x) = g(x)\left[1 - \left(\dfrac{f_1(x)}{g(x)}\right)^2\right], \\ & g(x) = 1 + 9\left[\dfrac{\sum_{i=2}^{n} x_i}{n-1}\right]^{0.25}. \end{cases} \tag{5.9}$$

All variables lie in the range $[0,\ 1]$. The Pareto optimal region corresponds to $0 \le x_1^* \le 1$ and $x_1^* = 0$ for $i = 2, 3, \ldots, 10$ [20].

DTLZ1

A simple test problem uses M objectives with a linear Pareto optimal front:

$$\text{DTLZ1}: \begin{cases} \text{Minimize} & f_1(x) = \dfrac{1}{2}x_1x_2\cdots x_{M-1}\left(1 + g(x_M)\right), \\ \text{Minimize} & f_2(x) = \dfrac{1}{2}x_1x_2\cdots(1 - x_{M-1})\left(1 + g(x_M)\right), \\ \vdots & \vdots \\ \text{Minimize} & f_{M-1}(x) = \dfrac{1}{2}x_1(1 - x_2)\left(1 + g(x_M)\right), \\ \text{Minimize} & f_M(x) = \dfrac{1}{2}(1 - x_1)\left(1 + g(x_M)\right) \\ \text{Subject to} & 0 \le x_i \le 1, \quad \text{for } i = 1, \ldots, n, \\ \text{Where} & g(x_M) = 100\left(|x_M| + \displaystyle\sum_{x_i \in x_M} (x_i - 0.5)^2 - \cos(20\pi(x_i - 0.5))\right). \end{cases} \tag{5.10}$$

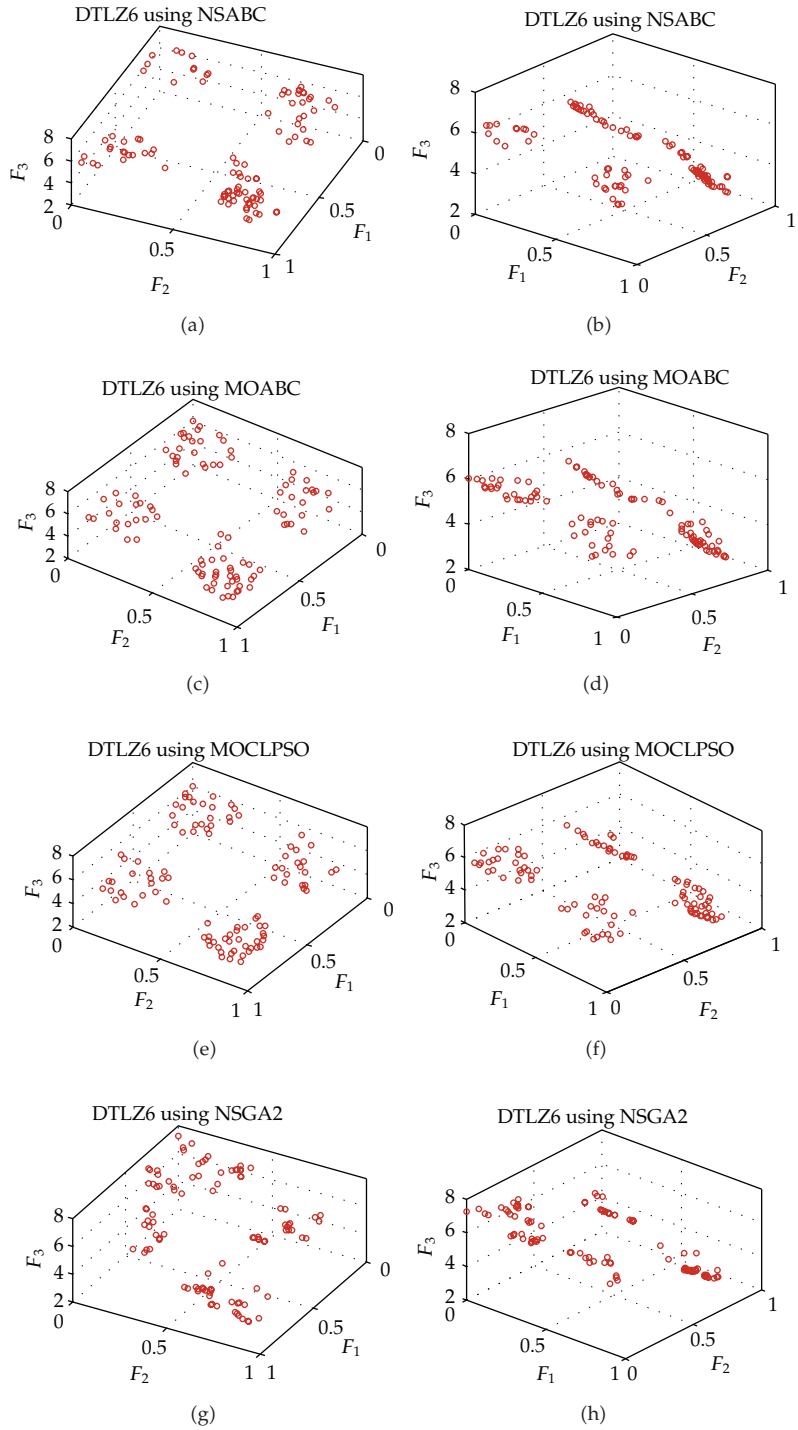

Figure 20: Pareto fronts obtained by NSABC, MOABC, MOCLPSO, and NSGA-II on DTLZ6 after 20000 FEs.

The Pareto optimal solution corresponds to $x_i^* = 0.5$ ($x_i^* \in x_M$), and the objective function values lie on the linear hyperplane: $\sum_{m=1}^{M} f_m^* = 0.5$. A value of $k = 5$ is suggested here. In the above problem, the total number of variables is $n = M + k - 1$. The difficulty in this problem is to converge to the hyperplane [21, 22].

DTLZ2

This test problem has a spherical Pareto optimal front:

$$\text{DTLZ2}: \begin{cases} \text{Minimize} & f_1(x) = (1 + g(x_M)) \cos\left(\frac{x_1\pi}{2}\right) \cdots \cos\left(\frac{x_{M-1}\pi}{2}\right), \\ \text{Minimize} & f_2(x) = (1 + g(x_M)) \cos\left(\frac{x_1\pi}{2}\right) \cdots \sin\left(\frac{x_{M-1}\pi}{2}\right), \\ \quad \vdots & \quad \vdots \\ \text{Minimize} & f_M(x) = (1 + g(x_M)) \sin\left(\frac{x_1\pi}{2}\right) \\ \text{Subject to} & 0 \le x_i \le 1, \quad \text{for } i = 1, \ldots, n, \\ \text{Where} & g(x_M) = \sum_{x_i \in x_M} (x_i - 0.5)^2. \end{cases} \tag{5.11}$$

The Pareto optimal solutions correspond to $x_i^* = 0.5$ ($x_i^* \in x_M$), and all objective function values must satisfy the $\sum_{m=1}^{M} (f_m^*)^2 = 1$. As in the previous problem, it is recommended to use $k = |x_M| = 10$. The total number of variables is $n = M + k - 1$ as suggested [21, 22].

DTLZ3

In order to investigate an MOEA's ability to converge to the global Pareto optimal front, the g function given in ZDTL1 is used in ZDTL2 problem:

$$\text{DTLZ3}: \begin{cases} \text{Minimize} & f_1(x) = (1 + g(x_M)) \cos\left(\frac{x_1\pi}{2}\right) \cdots \cos\left(\frac{x_{M-1}\pi}{2}\right), \\ \text{Minimize} & f_2(x) = (1 + g(x_M)) \cos\left(\frac{x_1\pi}{2}\right) \cdots \sin\left(\frac{x_{M-1}\pi}{2}\right), \\ \quad \vdots & \quad \vdots \\ \text{Minimize} & f_M(x) = (1 + g(x_M)) \sin\left(\frac{x_1\pi}{2}\right) \\ \text{Subject to} & 0 \le x_i \le 1, \text{ for } i = 1, \ldots, n, \\ \text{Where} & g(x_M) = 100\left(|x_M| + \sum_{x_i \in x_M} (x_i - 0.5)^2 - \cos(20\pi(x_i - 0.5))\right). \end{cases} \tag{5.12}$$

It is suggested that $k = |x_M| = 10$. There are a total of $n = M + k - 1$ decision variables in this problem. The Pareto optimal solution corresponds to $x_i^* = 0.5$ ($x_i^* \in x_M$) [21, 22].

DTLZ6

This test problem has $2^M - 1$ disconnected Pareto optimal regions in the search space:

$$\text{DTLZ6}: \begin{cases} \text{Minimize} & f_1(x) = x_1, \\ \text{Minimize} & f_2(x) = x_2, \\ \quad \vdots & \quad \vdots \\ \text{Minimize} & f_{M-1}(x) = x_{M-1}, \\ \text{Minimize} & f_M(x) = \left(1 + g(x_M)\right) h\left(f_1, f_2, \ldots, f_{M-1}, g\right) \\ \text{Subject to} & 0 \le x_i \le 1, \text{ for } i = 1, \ldots, n, \\ \text{Where} & g(x_M) = 1 + \dfrac{g}{|x_M|} \displaystyle\sum_{x_i \in x_M} x_i, \\ & h = M - \displaystyle\sum_{i=1}^{M-1} \left[\frac{f_i}{1+g} \left(1 + \sin\left(3\pi f_i\right)\right) \right]. \end{cases} \tag{5.13}$$

The functional g requires $k = |x_M|$ decision variables, and the total number of variables is $n = M + k - 1$. It is suggested that $k = 20$ [21, 22].

5.2. Performance Measures

In order to facilitate the quantitative assessment of the performance of a multiobjective optimization algorithm, two performance metrics are taken into consideration: (1) convergence metric Υ; (2) diversity metric Δ [4].

5.2.1. Convergence Metric

This metric measures the extent of convergence to a known set of Pareto optimal solutions, as follows:

$$\Upsilon = \frac{\sum_{i=1}^{N} d_i}{N}, \tag{5.14}$$

where N is the number of nondominated solutions obtained with an algorithm and d_i is the Euclidean distance between each of the nondominated solutions and the nearest member of the true Pareto optimal front. To calculate this metric, we find a set of $H = 10000$ uniformly spaced solutions from the true Pareto optimal front in the objective space. For each solution obtained with an algorithm, we compute the minimum Euclidean distance of it from H chosen solutions on the Pareto optimal front. The average of these distances is used as the convergence metric Υ. Figure 4 shows the calculation procedure of this metric [4].

5.2.2. Diversity Metric

This metric measure the extent of spread achieved among the obtained solutions. Here, we are interested in getting a set of solutions that spans the entire Pareto optimal region. This metric is defined as:

$$\Delta = \frac{d_f + d_l + \sum_{i=1}^{N-1} \left| d_i - \overline{d} \right|}{d_f + d_l + (N-1)\overline{d}}, \tag{5.15}$$

where d_i is the Euclidean distance between consecutive solutions in the obtained nondominated set of solutions and N is the number of nondominated solutions obtained with an algorithm. $\overline{d}$ is the average value of these distances. d_f and d_l are the Euclidean distances between the extreme solutions and the boundary solutions of the obtained nondominated set, as depicted in Figure 5.

5.3. Compared Algorithms and Parameter Settings

5.3.1. Nondominated Sorting Genetic Algorithm II (NSGA-II)

This algorithm was proposed by Deb et al. It is a revised version of the NSGA proposed by Guria [23]. NSGA-II is a computationally fast and elitist MOEA based on a nondominated sorting approach. It replaced the use of sharing function with the new crowded comparison approach, thereby eliminating the need of any user-defined parameter for maintaining the diversity among the population members. NSGA-II has proven to be one of the most efficient algorithms for multiobjective optimization due to simplicity, excellent diversity-preserving mechanism, and convergence near the true Pareto optimal set.

The original NSGA-II algorithm uses simulated binary crossover (SBX) and polynomial crossover. We use a population size of 100. Crossover probability $p_c = 0.9$ and mutation probability is $p_m = 1/n$, where n is the number of decision variables. The distribution indices for crossover and mutation operators are $\eta_c$mu = 20 and $\eta_m$mum = 20, respectively [4].

5.3.2. Multiobjective Comprehensive Learning Particle Swarm Optimizer (MOCLPSO)

MOCLPSO was proposed by Huang et al. [10]. It extends the comprehensive learning particle swarm optimizer (CLPSO) algorithm to solve multiobjective optimization problems. MOCLPSO incorporates Pareto dominance concept into CLPSO. It also uses an external archive technique to keep a historical record of the nondominated solutions obtained during the search process.The MOCLPSO uses population size $NP = 50$, archive size $A = 100$, learning probability $p_c = 0.1$, elitism probability $p_m = 0.4$ [10].

For the MOABC, a colony size of 50, archive size $A = 100$, elitism probability $p_m = 0.4$. NSABC algorithm does not need elitism probability. In the experiment, so as to compare the different algorithms, a fair time measure must be selected. It is, therefore, decided to use the number of function evaluations (FEs) as a time measure [24]. Thereby FEs in a time measure is our termination criterion.

5.4. Simulation Results for Benchmark Functions

The experimental results, including the best, worst, average, median, and standard deviation of the convergence metric and diversity metric values found in 10 runs are proposed in Tables 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, and 22 and all algorithms are terminated after 10000 and 20000 function evaluations, respectively. Figures 6, 7, 8, 9, 10, and 12 show the optimal front obtained by four algorithms for two-objective problems. The continuous lines represent the Pareto optimal front; star spots represent nondominated solutions found. Figure 11 shows the results of NSABC, MOABC, and NSGA-II algorithms optimizing the test function ZDT4. Figures 13, 14, 15, 16, 17, 18, 19, and 20 show the true Pareto optimal front and the optimal front obtained by four algorithms for DTLZ series problems.

From Tables 1 and 2, we can see that the performances of all the four algorithms in convergence metric have been competitively good over this problem. All solutions of them are almost on the true Pareto front. In diversity metric aspect, however, the performance of NSABC is better than the other algorithms. When given 20000 function evaluations for them, MOABC and MOCLPSO algorithms improve diversity metric. MOABC is as good as NSABC in diversity metric. From Figure 6, it can be seen that the front obtained from MOABC, NSABC, and MOCLPSO are found to be uniformly distributed. However, NSGA-II algorithm is not able to cover the full Pareto front. On the whole the MOABC and NSABC algorithms are a little better than MOCLPSO and NSGA-II on SCH problem.

For FON problem, like SCH, it can be observed from Tables 3 and 4 that all algorithms perform very well in convergence metric. In diversity metric aspect, MOABC, NSABC, and MOCLPSO algorithms can guarantee a good performance. On the other hand, even though NSGA-II is able to find the true Pareto front for this problem, it cannot maintain a relatively good diversity metric. It is only able to cover the half Pareto front, supporting the results of Figure 7. From Tables 3 and 4, we can also see that the performances of all the four algorithms could not be improved even in 20000 function evaluations.

On ZDT1 function, when given 10000 function evaluations for four algorithms, Table 5 shows that the performance of MOABC in convergence metric is one order of magnitude better than that of NSABC and NSGA-II, but it is one order of magnitude worse than that of MOCLPSO. However, when the number of function evaluations is 20000, we can see from Tables 5 and 6 that MOABC can greatly improve the performance of convergence and it has been two orders of magnitude better than that of MOCLPSO. For diversity metric, one can note from tables and algorithms that MOABC outperform NSABC in terms of diversity metric. Figure 8 shows that MOABC and MOCLPSO can discover a well-distributed and diverse solution set for this problem. However, NSABC and NSGA-II only find a sparse distribution, and they cannot archive the true Pareto front for ZDT1.

On ZDT2 function, the results of the performance measures show that MOABC and MOCLPSO have better convergence and diversity compared to the NSABC and NSGA-II. From Table 8, one can note that the performance of MOABC in convergence metric is one order of magnitude better than that of MOCLPSO after 20000 function evaluations. Figure 9 shows that NSGA-II produces poor results on this test function and it cannot achieve the true Pareto front. In spite of NSABC getting a relatively good convergence metric, it cannot maintain a good diversity metric.

ZDT3 problem has a number of disconnected Pareto optimal fronts. The situation is the same on ZDT2 problem. The performances of MOABC and MOCLPSO are almost the same after 10000 function evaluations, and they are a little better than that of NSABC, and two

orders of magnitude better than that of NSGA-II in convergence metric. However, when the number of function evaluations is 20000, it is found from Table 10 that the performances of MOCLPSO and NSGA-II could not be improved while MOABC and NSABC can improve the convergence metric by almost one order of magnitude. In diversity metric aspect, the results show that NSABC has worse diversity compared to the other algorithms, as can be seen in Figure 10.

The problem ZDT4 has 21^9 or about $8(10^{11})$ different local Pareto optimal fronts in the search space, of which only one corresponds to the global Pareto optimal front. The Euclidean distance in the decision space between solutions of two consecutive local Pareto optimal sets is 0.25 [20]. From Tables 11 and 12, it can be seen that MOABC, NSABC, and NSGA-II produce poor results in convergence metric. These three algorithms get trapped in one of its 21^9 local Pareto optimal fronts. Even though they cannot find the true Pareto front, they are able to find a good spread of solutions at a local Pareto optimal front, as shown in Figure 11. Since MOCLPSO converges poorly on this problem, we do not show MOCLPSO results on Figure 11 and Tables 11 and 12.

From Tables 13 and 14, we can see that the performances of MOABC, NSABC, and MOCLPSO are very good on this problem, and they outperform NSGA-II in terms of convergence metric and diversity metric. NSGA-II has difficulty in converging near the global Pareto optimal front. Considering the average and median values, we can observe that the proposed MOABC algorithm has better convergence in most test runs. For diversity metric, Figure 12 shows that MOABC and MOCLPSO effectively find nondominated solutions spread over the whole Pareto front. In spite of NSABC archiving the true Pareto front, it cannot maintain a good diversity metric. NSGA-II gets the worse distribution of solutions on the set of optimal solutions found.

From Tables 15 and 16, we can see that the performances of MOABC and NSABC are one order of magnitude better than that of NSGA-II in convergence metric. Since MOCLPSO converges poorly on this problem, we do not show MOCLPSO results. For diversity metric, Figure 14 shows that NSABC and MOABC effectively find nondominated solutions spreading over the whole Pareto front.

However, NSGA-II gets the worse distribution of solutions on the set of optimal solutions found.

On DTLZ2 function, from Tables 16 and 17, we can see that the performances of all the four algorithms in convergence metric have been competitively good over this problem. However, we can see that the performance of MOABC in convergence metric is one order of magnitude better than that of NSABC and NSGA-II and two orders of magnitude better than that of MOCLPSO. From Figure 16, it can be seen that the front obtained from MOABC, NSABC, and MOCLPSO is found to be uniformly distributed. However, NSGA-II algorithm is not able to cover the full Pareto front.

Like the DTLZ1 function, MOCLPSO converges poorly on DTLZ3 problem. For this problem, all algorithms could not quite converge on to the true Pareto optimal fronts after 20000 FEs. MOABC is a little better than the other algorithms in convergence metric. From tables, algorithms, and Figure 16, we have found that NSGA-II is bad in solving DTLZ3 in diversity metric. However, MOABC and NSABC can discover a well-distributed and diverse solution set for this problem.

DTLZ6 problem has $2^M - 1$ disconnected Pareto optimal regions in the search space. This problem will test an algorithm's ability to maintain subpopulation in different Pareto optimal regions. From Tables 21 and 22, we can observe that the performance of MOABC and MOCLPSO in convergence metric is one order of magnitude better than that of NSABC

and NSGA-II. From Figure 20, MOABC and MOCLPSO can discover a well-distributed and diverse solution set for this problem. However, the performances of NSABC and NSGA-II in diversity metric are a little worse than that of MOABC and MOCLPSO.

6. Summary and Conclusion

In this paper, we present a novel article bee colony (ABC) algorithm to solving multiobjective optimization problems, namely, multiobjective artificial bee colony (MOABC). In our algorithm, we use Pareto concept and external archive strategy to make the algorithm converge to the true Pareto optimal front and use comprehensive learning strategy to ensure the diversity of population. The best advantage of MOABC is that it could use less control parameters to get the most competitive performance. In order to demonstrate the performance of the MOABC algorithm, we compared the performance of the MOABC with those of NSABC (with nondominated sorting strategy only), MOCLPSO, and NSGA-II optimization algorithms on several two-objective test problems and three-objective functions.

For the two-objective test problems, from the simulation results, it is concluded that MOABC algorithm could converge to the true Pareto optimal front and maintain good diversity along the Pareto front. We can also see that MOABC is much better than NSABC in terms of diversity metric. We believe that comprehensive learning strategy could help find more nondominated solutions and improve the capabilities of the algorithm to distribute uniformly the nondominated vectors found. Additionally, from the simulation result we also know that MOABC has the same performance as MOCLPSO after 10000 FEs. They are much better than the other two algorithms. NSABC is a little better than NSGA-II. However, when the number of function evaluations is 20000, we can see that MOABC can greatly improve the performance of convergence, and it is 1-2 orders of magnitude better than that of MOCLPSO. NSABC also has a great improvement in convergence metric. However, it still has a bad diversity metric.

For DTLZ series problems, we can see that the performances of all algorithms are improved very small, when we increase the number of function evaluation. Simulation results show that MOCLPSO converges poorly on DTLZ1 and DTLZ3 problems. That means that MOCLPSO performs badly in terms of convergence for three-objective problem. We also know that MOABC and NSABC perform the best in terms of convergence and MOABC is a little better than NSABC. In diversity metric aspect, MOABC and NSABC can discover a well-distributed and diverse solution set for DTLZ series problems. However, In spite of NSGA-II getting a relatively good convergence metric, it cannot maintain a good diversity metric for DTLZ series problems.

On the whole, the proposed MOABC significantly outperforms three other algorithms in terms of convergence metric and diversity metric. Therefore, the proposed MOABC optimization algorithm can be considered as a viable and efficient method to solve multiobjective optimization problems. As our algorithm is not applied in the real problems, robustness of the algorithm needs to be verified. In the future work, we will apply MOABC algorithm to the real engineering problems.

Acknowledgment

This work is supported by National Natural Science Foundation of China under nos. 61174164, 61003208, and 61105067.

References

[1] G. B. Dantzig and M. N. Thapa, *Linear Programming I: Introduction*, Springer Series in Operations Research, Springer-Verlag, New York, NY, USA, 1997.

[2] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*, Wiley-Interscience Series in Systems and Optimization, John Wiley & Sons, Chichester, UK, 2001.

[3] C. A. Coello Coello, "Recent trends in evolutionary multi-objective optimization," in *Evolutionary Multi-Objective Optimization: Theoretical Advances and Applications*, A. Abraham, L. C. Jain, and R. Goldberg, Eds., pp. 7–32, Springer, Berlin, Germany, 2005.

[4] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[5] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: improving the strength Pareto evolutionary algorithm," Tech. Rep., Computer Engineering and Networks Laboratory [TIK], Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, May 2001.

[6] C. A. C. Coello and M. S. Lechuga, "MOPSO: a proposal for multiple objective particle swarm optimization," in *Proceedings of Congresson Evolutionary Computation (CEC '02)*, vol. 2, pp. 1051–1056, IEEE Press, 2002.

[7] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," Tech. Rep. TR06, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.

[8] D. Karaboga and B. Basturk, "On the performance of artificial bee colony (ABC) algorithm," *Applied Soft Computing Journal*, vol. 8, no. 1, pp. 687–697, 2008.

[9] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 3, pp. 281–295, 2006.

[10] V. L. Huang, P. N. Suganthan, and J. J. Liang, "Comprehensive learning particle swarm optimizer for solving multiobjective optimization problems," *International Journal of Intelligent Systems*, vol. 21, no. 2, pp. 209–226, 2006.

[11] K. Miettinen, *Nonlinear multiobjective optimization*, International Series in Operations Research & Management Science, 12, Kluwer Academic Publishers, Boston, MA, 1999.

[12] H. Li and Q. Zhang, "Multiobjective optimization problems with complicated pareto sets, MOEA/D and NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 284–302, 2009.

[13] D. Karaboga and B. Basturk, "Artificial Bee Colony (ABC) optimization algorithm for solving constrained optimization problems," in *Proceedings of the 12th international Fuzzy Systems Association world congress on Foundations of Fuzzy Logic and Soft Computing (IFSA '07)*, vol. 4529, pp. 789–798, Springer, 2007.

[14] D. Karaboga and B. Akay, "Artificial Bee Colony (ABC) Algorithm on training artificial neural networks," in *Proceedings of the IEEE 15th Signal Processing and Communications Applications (SIU '07)*, June 2007.

[15] B. Akay and D. Karaboga, "Parameter tuning for the artificial bee colony algorithm," in *Proceedings of the 1st International Conference on Computational Collective Intelligence (ICCCI '09)*, vol. 5796 LNAI, pp. 608–619, Wroclaw, Poland, 2009.

[16] J. D. Knowles and D. W. Corne, "Approximating the nondominated front using the Pareto Archived Evolution Strategy," *Evolutionary computation*, vol. 8, no. 2, pp. 149–172, 2000.

[17] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*, Wiley-Interscience Series in Systems and Optimization, John Wiley & Sons, Chichester, UK, 2001.

[18] J. D. Schaffer, "Multipleobjective optimization with vector evaluated genetic algorithms," in *Proceedings of the First International Conference on Genetic Algorithms*, J. J. Grefensttete, Ed., pp. 93–100, Lawrence Erlbaum, Hillsdale, NJ, USA, 1987.

[19] C. M. Fonseca and P. J. Fleming, "Multiobjective optimization and multiple constraint handling with evolutionary algorithms—part II: application example," *IEEE Transactions on Systems, Man, and Cybernetics Part A*, vol. 28, no. 1, pp. 38–47, 1998.

[20] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: empirical results," *Evolutionary computation*, vol. 8, no. 2, pp. 173–195, 2000.

[21] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable test problems for evolutionary multi-objective optimization," Tech. Rep. 2001001, Kanpur Genetic Algorithms Laboratory (KanGAL), Indian Institute of Technology, Kanpur, India, 2001.

[22] C. A. Coello Coello, G. B. Lamont, and D. A. Van Veldhuizen, *Evolutionary Algorithms for Solving Multi-Objective Problems*, Genetic and Evolutionary Computation Series, Springer, New York, NY, USA, 2nd edition, 2007.

[23] C. Guria, P. K. Bhattacharya, and S. K. Gupta, "Multi-objective optimization of reverse osmosis desalination units using different adaptations of the non-dominated sorting genetic algorithm (NSGA)," *Computers and Chemical Engineering*, vol. 29, no. 9, pp. 1977–1995, 2005.

[24] M. P. Wachowiak, R. Smolíková, Y. Zheng, J. M. Zurada, and A. S. Elmaghraby, "An approach to multimodal biomedical image registration utilizing particle swarm optimization," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 289–301, 2004.