Soft Computing (2018) 22:7833–7846
https://doi.org/10.1007/s00500-018-3499-9

FOCUS

External archive matching strategy for MOEA/D

Feng Wang[1] · Heng Zhang[1] · Yixuan Li[1] · Yaoyu Zhao[1] · Qi Rao[2]

Published online: 6 September 2018

Abstract

Multiobjective evolutionary algorithms based on decomposition (MOEA/D) decompose a multiobjective optimization problem (MOP) into a group of subproblems and optimizes them at the same time. The reproduction method in MOEA/D, which generates offspring solutions, has crucial effect on the performance of algorithm. As the difficulties of MOPs increases, it requires much higher efficiency for the reproduction methods in MOEA/D. However, for the complex optimization problems whose PS shape is complicated, the original reproduction method used in MOEA/D might not be suitable to generate excellent offspring solutions. In order to improve the property of the reproduction method for MOEA/D, this paper proposes an external archive matching strategy which selects solutions' most matching archive solutions as parent solutions. The offspring solutions generated by this strategy can maintain a good convergence ability. To balance convergence and diversity, a perturbed learning scheme is used to extend the search space of the solutions. The experimental results on three groups of test problems reveal that the solutions obtained by MOEA/D-EAM have better convergence and diversity than the other four state-of-the-art algorithms.

Keywords Reproduction method · Decomposition · Multiobjective optimization

1 Introduction

There are many optimization problems in real life. A number of optimization problems have multiple optimization objectives, and these problems are called multiobjective optimization problems (MOPs). MOPs arise in many field of science, including economics, engineering, and logistics (Gupta et al. 2016; Alzain et al. 2015; Yuan et al. 2017; Alsmirat et al. 2017). Since all objectives of MOP always conflict with each other and a single solution cannot optimize these objectives at the same time, all objectives need to be considered simultaneously. A multiobjective optimization problem can be mathematically formulated as follows:

$$\begin{array}{l} \text{minimize } \ F(x) = (f1(x), \ldots, fm(x)) \\ \text{subject to } \ x \in \pi \end{array} \tag{1}$$

where π is the decision space, $X = (1, \ldots, xn) \in R^n$ is a decision variable vector, and $F{:}\pi \to R^m$ consists m of objective functions $f1(x), \ldots, fm(x)$. If $x \in R^n$, all the objectives are continuous and this problem can be called a continuous MOP.

Suppose $u, v \in \pi$, u dominates v, denoted by $u \prec v$, if and only if $fi(u) \le fi(v)$, for all $i \in \{1, \ldots, m\}$, and $F(u) \neq F(v)$. A point $X^* \in \pi$ represents the Pareto optimal solution if no other solution $x \in \pi$ dominates X^*. 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) (Hillermeier 2001).

Since the objective functions of MOPs often conflict with each other, and the improvement of one objective may lead to deterioration of another, no single solution can optimize all objectives at the same time. Multiobjective optimiza-

Communicated by B. B. Gupta.

✉ Qi Rao
qirao@pku.edu.cn

Feng Wang
fengwang@whu.edu.cn

Heng Zhang
zhangheng96@whu.edu.cn

Yixuan Li
271623470@qq.com

Yaoyu Zhao
949670640@qq.com

1 School of Computer Science, Wuhan University, Wuhan, China

2 Institute of Computational Linguistics, Peking University, Beijing, China

tion evolutionary algorithms (MOEAs) treat a MOP as a whole optimization problem and use Pareto dominance relationship among solutions to drive the population evolution toward the Pareto front (Deb and Kalyanmoy 2001; Beng and Beng 2004). The strategy selection plays a crucial role in MOEAs. Based on the selection of the strategies, most existing MOEAs can be classified into three categories: (1) Pareto dominance-based MOEAs, in which solutions are selected based on their Pareto dominance relationship (Srinivas and Deb 2014); although Pareto dominance-based MOEAs are very powerful for solving bi-objective and tri-objective MOPs, their performance dramatically degrades when the number of objectives increases; (2) performance indicator-based MOEAs (Beume et al. 2007), which utilizes performance indicator as selection criterion, the main challenge of performance indicator-based MOEAs is high computational complex for computing the performance indicators, especially when the number of objectives increases; (3) decomposition-based MOEAs (Zhang and Li 2007), which converts a MOP into a set of subproblems and optimizes them in a collaborative manner.

MOEA/D is an evolutionary algorithm for multiobjective optimization based on decomposition approach. It has been widely used for dealing with numbers of MOPs, such as manufacturing control in engineering optimization (He et al. 2016; Wang et al. 2018b; Wu et al. 2017; Li et al. 2017c, d), parameter tuning in pattern recognition (Zhou et al. 2018; Wang et al. 2017a; Huang et al. 2016; Cao et al. 2018; Xie et al. 2018; Zhang et al. 2017; Li et al. 2016, 2018a, b), and multi-source scheduling in cloud computing (Li et al. 2017a, b; Lin et al. 2017a, b; Wang et al. 2017b) (Lai et al. 2017), and shows a competitive performance (Wang et al. 2018a). In recent years, numerous variants of MOEA/D have been published. Qi (Qi et al. 2014) proposed a new weight vector generation method, called WS transformation demonstrated, which helps algorithm to obtain more uniformly distributed Pareto optimal solutions. Sato (Sato 2015) proposed a novel decomposition approach based on original PBI approach to overcome the difficulty in approximating Pareto front in some problems. Li et al. (2015) tried to overcome the difficulty in approximating Pareto front in some problems. Since there have been many literatures that focus on improving the efficiency of decomposition approach and weight vector generation approach, the reproduction methods, which are utilized to generate offspring solutions, are mostly the same as the reproduction method described in basic MOEA/D. Sufficient experimental results reveal that the reproduction methods are crucial for the performance of MOEA/D (Li and Zhang 2009). However, in basic MOEA/D, offspring solution is only generated from their two neighbor solutions by using genetic operators. As the difficulty of MOPs increases, the search space is complex and the neighbor solutions might not always be suitable for generating excellent offspring solutions.

To improve the efficiency of reproduction method and generate more excellent offspring solutions, in this paper, we propose an external archive matching (EAM) strategy for MOEA/D (MOEA/D-EAM). This strategy maintains an external archive to store the best solutions, and then, the current solution will select the two most matching archive solutions instead of neighbor solutions to generate offspring solution. On account of that the selected two solutions are non-dominated solutions found so far, the convergence of the generated offspring solution can be guaranteed. Furthermore, to balance population diversity and convergence, a perturbed learning scheme is proposed in this paper. Current solutions will select other unmatched archive solutions as parent solutions under a certain probability. The experimental results show that the proposed strategy can improve the exploitation ability and accelerate the convergence rate.

The rest of this paper is organized as follows. A brief introduction of the basic MOEA/D is present in Sect. 2. The detail of our proposed methodologies is described in Sect. 3. The experimental results and analysis on three groups of test problems are shown in Sect. 4, and conclusion is provided in Sect. 5.

2 Background of MOEA/D

In this section, we firstly introduce the three main components of MOEA/D, i.e., decomposition approaches, neighborhood, and reproduction methods. Then, we describe the classical framework of MOEA/D in detail.

2.1 Decomposition approaches

In MOEA/D, a MOP can be decomposed into N scalar subproblems by decomposition approach. Tchebycheff approach is a mostly used decomposition approach. Therefore, we take Tchebycheff approach as the decomposition approach in this paper. Then, the ith subproblem can be defined as follows:

$$\begin{aligned} &\text{minimize } g\left(x^i \middle| \lambda^i, Z^*\right) = \max\left\{\lambda_j^i \left| fj(x^i) - z_j^* \right|\right\} \\ &\text{subject to } x^i \in \pi \end{aligned} \tag{2}$$

where $i \in \{1, \ldots, N\}$ and N is the number of solutions. For all $j \in \{1, \ldots, M\}$, M is the number of objectives. Furthermore, $fj(*)$ is the jth objective function, x^i is the ith solution, $\lambda^i = (\lambda_1^i, \ldots, \lambda_m^i)$ is the ith weight vector corresponding to the ith subproblem, and $Z^* = (z1, \ldots, zm)$ is the reference vector.

To convert a MOP into N subproblems, N weight vectors should be selected. As detailed in (Zhang and Li 2007), the weight vector-generating approach is utilized in this paper.

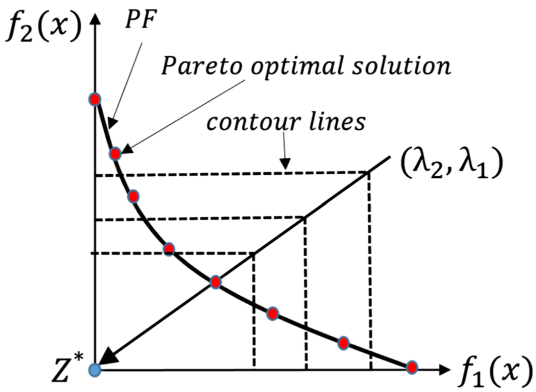

Fig. 1 The Tchebycheff decomposition approach

Figure 1 illustrates the Tchebycheff decomposition approach which uses a set of evenly distributed weight vectors, and the red points here are Pareto optimal solutions defined by weight vectors. Under the condition that weight vectors are evenly spread in the M dimensions objective space, the Pareto optimal solutions will be also evenly distributed along the PF.

2.2 Neighborhood

Since each subproblem is defined by a specific weight vector and its objective function is continuous of the weight vector, we can assume that subproblems have similar Pareto optimal solutions if their weight vectors are close. Under this assumption, MOEA/D introduces a neighbor concept (Zhang and Li 2007), a subproblem has T neighbor subproblems. It means that the weight vectors of these T neighbor subproblems are very close to the subproblem in terms of Euclidean distance. To make use of evolutionary information obtained by the neighbor subproblems, subproblem's offspring solution is generated from neighbor subproblems' solution using specific operators. Besides that, the generated offspring might be used to replace its neighbor subproblems' solution during the updating of population. Thus, two different neighborhood concepts are used in MOEA/D (Ishibuchi et al. 2009, 2013). One neighbor is used for generating offspring, called mating neighborhood, and another is used for replacing neighbor subproblems' solutions, i.e., replacement neighborhood.

2.3 Reproduction methods of MOEA/D

In the classical MOEA/D, parent solutions are selected from their neighbor solutions. Then offspring solutions are generated from these solutions by using genetic operators (Zhang and Li 2007). In MOEA/D-DE, which is a new version of MOEA/D based on differential evolution, offspring solutions are generated from neighbor solutions by using differential evolution operators. Besides that, offspring solutions may be produced by other solutions selected from the whole population. The parent solutions set in MOEA/D-DE can be described as follows:

$$P = \begin{cases} B(i) & \text{with probability } \delta \\ \{1, \ldots, N\} & \text{with probability } 1 - \delta \end{cases} \tag{3}$$

where δ is a random number from [0, 1] and $B(i)$ is the neighbor solutions' indexes of ith solution. Experimental results reveal that MOEA/D-DE has a promising ability in solving MOPs with complicated PS shape (Li and Zhang 2009).

2.4 Framework of MOEA/D

MOEA/D is an iterative algorithm. It maintains a set of solutions $x^1, x^2, \ldots, x^N$ at each generation, where solution x^i is the ith subproblem's best solution found so far in terms of the $g^i(x^i)$. In this paper, we utilize the Tchebycheff approach as decomposition approach, and then $\mathrm{FV}^1, \mathrm{FV}^2, \ldots, \mathrm{FV}^N$ are the F value of solutions. The algorithm works as follows:

Algorithm 1: Framework of MOEA/D

Input:
A MOP;
A stopping condition;
N, the number of solutions and weight vectors;
T, the number of neighbor subproblems;

Initialize:
Initialize N weight vectors $\lambda^1, \lambda^2, \ldots, \lambda^N$;
Set N subproblems defined by N weight vectors;
Set T neighbor subproblems $B(i) = i^1, i^2, \ldots, i^T$ for i^{th} subproblem, where $i^1, i^2, \ldots, i^T$ subproblems have the T closest weight vectors;
Generate N initial solutions $x^1, x^2, \ldots, x^N$;
Calculate FV^i of subproblem corresponding to x^i;

Update:
while Stop condition is not satisfied **do**
  for each $i \in [1, N]$ **do**
  Randomly select two solutions from $B(i)$ to generate offspring y^i using GA operators;
  Repair the solution y^i using a problem-specific method;
  If the subproblems value of $g^i(y^i) < g^i(x^i)$, replace x^i with y^i, $FV^i = g^i(y^i)$;
    for each $t \in [1, T]$ **do**
      If $g^{i^t}(y^i) < g^{i^t}(x^{i^t})$
      Replace x^{i^t} with y^i, $FV^{i^t} = g^i(y^i)$;
    end for
  end for
end while

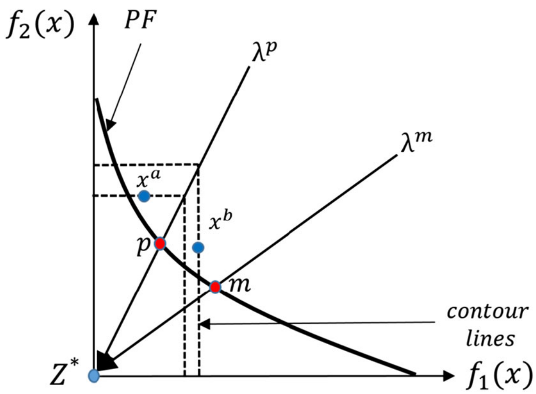

Fig. 2 The neighbor solutions of λ^m

3 MOEA/D-EAM algorithm

Suppose the PF of a MOP is continuous, since that each subproblem is defined by a specific weight vector, it can be assumed that subproblems produced by similar weight vector have similar Pareto optimal solutions. It has been proved that algorithms whose solutions are only allowed to mate with their neighbor solutions maintain an excellent ability of exploitation (Zhang et al. 2016). As a result, in the reproduction process of classical MOEA/D, parent solutions selected from neighboring solutions may improve the convergence of algorithm.

However, solution and its neighboring solutions in MOEA/D may not always be near to each other during evolutionary process. As detailed in Fig. 2, red points p and m are Pareto optimal points defined by weight vectors λ^p and λ^m, the dashed lines are contour lines produced by Tchebycheff decomposition approach, blue points x^a and x^b are two solutions to the pth subproblem, they are on two different contour lines and x^a has a smaller FV value than x^b in terms of the pth subproblem. It means that x^a will be reserved and x^b will be discarded in the following iterations. During the reproduction procedure of x^m, parent solutions are selected from its neighbor solutions. However, the reserved neighbor solution x^a may not be good for producing the mth subproblem's offspring since it is far away from mth subproblem's optimal solution in the objective space and decision space as well. The generated offspring by using x^a may search in the space around x^a, and the rate to find m will be slowed down. As a conclusion, the original reproduction methods used in MOEA/D would discourage convergence.

3.1 External archive matching strategy

To improve the efficiency of the reproduction method for MOEA/D, we propose the following external matching strategy, i.e., EAM. In MOEA/D-EAM, an external archive will be established to record N non-dominated solutions during iterations. For each offspring solution, it will be produced as follows:

- Choose the two most matching solution from archive as parent solutions;
- Generate offspring using the DE operators.

As mentioned above, the original reproduction method in which offsprings are generated by using neighbor solutions may discourage the convergence of algorithm. In order to reduce the adverse impact caused by these methods, we choose the two most matching solutions as parent solutions. Here, we define the index of the most matching archive solution j for a given ith solution as follows:

$$j = \arg \min_{1\le k\le N} \{\text{distance}(k, i)\}$$
$$i = 1, \ldots, N \tag{4}$$

where distance() is a function that calculates the distance in objective space between x^i and all archive solutions. The jth solution in archive will be utilized as one parent solution. Now, we can make some remarks on EAM strategy. As detailed in Fig. 2, the subproblem defined by weight vector λ^p may find solutions that are more approximate to optimal point p, i.e., x^a. However, x^a may be improper for generating the offspring of the mth subproblem. In EAM strategy, solutions will evolve toward its two most matching solutions.

As shown in Fig. 3, suppose the blue points are sets of archive solutions obtained by other subproblems, x^m is the current solution found by the mth subproblem. When generating the offspring of the mth subproblem, mth would choose its most matching solution, i.e., x^b and x^c from archive as parent solutions. Then the offspring is obtained as follows:

$$y_k^m = \begin{cases} x_k^m + F\left(x_k^b - x_k^c\right) & \text{with probability CR} \\ x_k^m & \text{with probability } 1 - \text{CR} \end{cases} \tag{5}$$

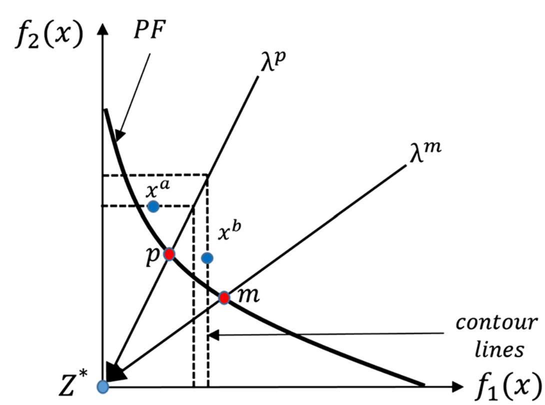

Fig. 3 The reproduction process using most matching solutions

where $k \in [1, M]$, M is the number of variables, F and CR are the control parameter in DE operators detailed in (Li and Zhang 2009). In this case, x^b would be used as parent solution for generating the mth offspring. Consequently, solutions produced by the whole population during iterations would be considered as parent solution, which promote the convergence.

3.2 Perturbed learning scheme

The proposed EAM strategy takes the solutions' most matching archive solutions as parents, and the generated offspring solutions may search around their parents and maintain a good convergence. However, as described in (Liu et al. 2014), the population diversity is more important than convergence in multiobjective evolutionary algorithm for solving some MOPs. Since MOEA/D-EAM takes most matching archive solutions of solution to generate offspring, the diversity of population is depending on archive solutions to a certain degree.

In order to promote the diversity of population, offspring will be generated by some randomly selected solutions from archive under a certain probability; the search space of solutions will be extended. The process of this scheme can be detailed as follows:

Algorithm 2: Procedure of perturbed learning scheme

```
Uniformly generate a random number rand from [0, 1];
if rand < δ then
  Randomly select two solutions, A^p and A^q from archive as
  parent solutions;
  Generate offspring solution using DE operators;
end if
```

where δ is the control parameter present in MOEA/D-DE (Li and Zhang 2009) and A^p and A^q are two solutions selected from archive set A. Since these randomly selected archive solutions are approximated to PF, the offspring's convergence can be guaranteed. Consequently, in MOEA/D-EAM, an external archive is maintained to efficiently produce excellent offspring with great convergence and to ensure the diversity of population as well.

3.3 Framework of MOEA/D-EAM

The main difference between MOEA/D-EAM and MOEA/D is that MOEA/D-EAM maintains an external archive as mating pool. Neighbor solutions can be replaced by a good solution, but they are no more used for generating offspring solutions. The framework of MOEA/D-EAM is shown as follows:

Algorithm 3: Framework of MOEA/D-EAM

```
Input:
A MOP;
A stopping condition;
N, the number of solutions and weight vectors;
T, the number of neighbor subproblems;
An external archive, A, the maximum number of solutions in
archive is set to be N.
Initialize:
Initialize N weight vectors λ^1, λ^2, ..., λ^N;
Set N subproblems defined by N weight vectors;
Set T neighbor subproblems B(i) = i^1, i^2, ..., i^T for i^th
subproblem, where i^1, i^2, ..., i^T subproblems have the T
closest weight vectors;
Generate N initial solutions x^1, x^2, ..., x^N;
Calculate FV^i of subproblem corresponding to x^i;
Update:
while Stop condition is not satisfied do
 for each i ∈ [1, N] do
  if rand > δ then
   Find two most matching solutions of x^i, A^p, A^q;
   Generate the m^th offspring y^i using A^p, A^q following
   the DE operators described in equation.5;
  end if
  if rand ≤ δ then
   Randomly selected two solutions from archive, A^p, A^q;
   Generate the m^th offspring y^i using A^p, A^q following
   the DE operators described in equation.5;
  end if
  Repair the solution y^i using a problem-specific method;
  Update external archive;
  If the subproblems value of g^i(y^i) < g^i(x^i), replace
  with y^i, FV^i = g^i(y^i);
  for each t ∈ [1, T] do
   If g^{i^t}(y^i) < g^{i^t}(x^{i^t})
   Replace x^{i^t} with y^i, FV^{i^t} = g^i(y^i);
  end for
 end for
end while
```

3.4 Computational complexity of MOEA/D-EAM

The main difference between MOEA/D-EAM and original MOEA/D is the reproduction method. Since MOEA/D-EAM maintains an external archive to record N non-dominated solutions, the computational complexity of the archive-updating process is $O(N)$. To select the current solution's most matching archive solution, MOEA/D-EAM will calculate distance between the current solution and all archive solutions in the objective space, so the computational complexity of the archive matching process is $O(N \cdot M)$, M is the number of objectives. As a result, the entire computational complexity of the proposed approach is $O(N \cdot M)$.

4 Experimental studies

To test the performance of MOEA/D-EAM, we choose four state-of-the-art MOEAs to compare with MOEA/D-EAM, including Pareto dominance-based algorithm NSGA-II (Deb et al. 2002), decomposition-based algorithm MOEA/D (Zhang and Li 2007), a version of MOEA/D based on differential evolution MOEA/D-DE (Li and Zhang 2009) and MOEA/D with adaptive replacement strategy MOEA/D-AGR (Wang et al. 2017c). NSGA-II is a widely used Pareto dominance-based MOEA. MOEA/D is a classical decomposition-based MOEA which uses scalaring function to convert a MOP into a number of optimization subproblems aiming to get uniformly distributed Pareto optimal solutions. MOEA/D-DE is a kind of MOEA/D based on differential evolutions, and it has an outstanding ability to deal with MOPs with complicated PS shapes. MOEA/D-AGR is a modification of MOEA/D-DE, it utilizes an adaptive global replacement scheme to accelerate the convergence of algorithm.

4.1 Test problems

In our experimental study, we choose five test instances from ZDT (Zitzler et al. 2000) and four test instances from DTLZ (Deb et al. 2005) to validate the ability of MOEA/D-EAM to solve bi-objective or tri-objective optimization problems. Moreover, five test instances from UF (Zhang et al. 2008) are chosen to verify the efficiency of MOEA/D-EAM in solving MOPs with complicated PS shape. The parameter values and characteristics of all test instances are shown in Table 1, where n is the number of variables and m is the number of objectives.

4.2 Performance metric

The widely used performance metric, i.e., inverted generational distance (IGD) (Coello and Corts 2005; Schutze et al. 2012) indicator is utilized to evaluate the performance of algorithms. Let S^* be a set of Pareto optimal solutions, they are uniformly distributed along the PF. Let S be a set of the approximate solutions found by an algorithm, the IGD-metric can be obtained by Eq. 6:

$$\mathrm{IGD}(S^*, S) = \frac{\sum_{F \in S^*} d(F, S)}{S^*} \quad (6)$$

Table 1 Test instances

Name	n	m	Range
ZDT1	30	2	$x^i \in [0, 1], 1 \leq i \leq n$
ZDT2	30	2	$x^i \in [0, 1], 1 \leq i \leq n$
ZDT3	30	2	$x^i \in [0, 1], 1 \leq i \leq n$
ZDT4	10	2	$x^1 \in [0, 1], x^i \in [-5, 5], 2 \leq i \leq n$
ZDT6	10	2	$x^i \in [0, 1], 1 \leq i \leq n$
DTLZ1	12	3	$x^i \in [0, 1], 1 \leq i \leq n$
DTLZ2	12	3	$x^i \in [0, 1], 1 \leq i \leq n$
DTLZ3	12	3	$x^i \in [0, 1], 1 \leq i \leq n$
DTLZ4	12	3	$x^i \in [0, 1], 1 \leq i \leq n$
UF1	30	2	$x^1 \in [0, 1], x^i \in [-1, 1], 2 \leq i \leq n$
UF2	30	2	$x^1 \in [0, 1], x^i \in [-1, 1], 2 \leq i \leq n$
UF3	30	2	$x^i \in [0, 1], 1 \leq i \leq n$
UF4	30	2	$x^1 \in [0, 1], x^i \in [-2, 2], 2 \leq i \leq n$
UF5	30	2	$x^1 \in [0, 1], x^i \in [-1, 1], 2 \leq i \leq n$

where $d(F, S)$ is the minimum Euclidean distance from a Pareto optimal solution to the solutions in S. Suppose $|S^*|$ is large enough, IGD(S^*, S) can be used to evaluate both diversity and convergence of S. A very small IGD value means that the solutions obtained by an algorithm have a great diversity and convergence. Here, we choose 500 uniformly distributed Pareto optimal solutions for test instances with two optimization objectives and 1000 for test instances with three optimization objectives.

4.3 Parameter settings

The parameter settings of all algorithms are listed as follows: The simulated binary crossover (SBX) operator and polynomial mutation (Deb and Beyer 2001) are utilized as reproduction method in NSGA-II and MOEA/D. The differential evolution (DE) operator and polynomial mutation are adopted as reproduction method in MOEA/D-DE, MOEA/D-AGR and MOEA/D-EAM.

– The population size N is set to be 100 for ZDT problems and 300 for DTLZ problems and UF problems.
– Each algorithm will run on each test instance for 20 times independently, and the algorithms will stop after the number of iterations exceeds a maximal number. The maximal number of iterations is set to be 250 for ZDT problems, 300 for DTLZ problems and 500 for UF problems.
– The control parameters in reproduction process are the same as in NSGA-II (Deb et al. 2002), MOEA/D-DE (Li

and Zhang 2009), MOEA/D-EAM and MOEA/D-AGR (Wang et al2017c). To be specific, CR = 1.0 and F = 0.5 for DE operator, η is set to be 20 and Pm is set to be $1/n$ for polynomial mutation.
- The size of neighborhood for mating and replacement, i.e., Tm and Tr in MOEA/D, MOEA/D-DE and MOEA/D-EAM are set as $0.1 \times N$, Tm in MOEA/D-AGR is set as $0.1 \times N$ and Tr in MOEA/D-AGR is adaptively changed.
- The maximal number of solutions replaced by offspring solutions, i.e., nr in MOEA/D-DE is set as 2.
- All the variants of MOEA/D take the Tchebycheff decomposition as the decomposition approach.

4.4 Experimental results

To verify the performance of the proposed MOEA/D-EAM algorithm, we choose three groups of test problems. These test problems include 9 problems with simple PS shape and 5 problems with complicated PS shape. The comparison results are shown in Tables 2, 3 and 4 that present the mean and standard deviation of the IGD-metric values acquired by each algorithm after 20 times independently run on each test instances, respectively. Here, we mark the smallest value obtained by these algorithms on each test function with bold font and mark the second smallest value with underline.

Table 2 Results comparison on five ZDT problems

Problem	IGD	NSGA-II	MOEA/D	MOEA/D-DE	MOEA/D-AGR	MOEA/D-EAM
ZDT1	Mean	5.02E – 02	6.93E – 02	1.26E – 01	**7.69E – 03**	8.37E – 03
	SD	9.08E – 03	3.52E – 02	6.39E – 02	**9.40E – 04**	1.42E – 03
ZDT2	Mean	6.30E – 02	3.82E – 02	7.24E – 02	8.02E – 03	**7.54E – 03**
	SD	1.89E – 02	2.09E – 02	4.22E – 02	8.05E – 04	**5.31E – 04**
ZDT3	Mean	**8.51E – 02**	2.48E – 01	2.08E – 01	1.21E – 01	1.15E – 01
	SD	1.89E – 02	9.69E – 02	6.73E – 02	**1.06E – 02**	4.57E – 02
ZDT4	Mean	3.16E – 02	1.74E – 01	1.23E – 01	6.21E – 02	**4.94E – 02**
	SD	3.39E – 02	7.95E – 02	5.30E – 02	2.86E – 02	**1.67E – 02**
ZDT6	Mean	9.92E – 02	**4.57E – 02**	4.58E – 02	4.93E – 02	4.95E – 02
	SD	5.52E – 02	1.90E – 03	1.30E – 03	9.49E – 04	**7.02E – 04**

Table 3 Results comparison on four DTLZ problems

Problem	IGD	NSGA-II	MOEA/D	MOEA/D-DE	MOEA/D-AGR	MOEA/D-EAM
DTLZ1	Mean	5.56E – 01	3.11E – 01	2.18E – 01	2.67E – 01	**7.97E – 02**
	SD	2.31E – 01	2.56E – 01	2.23E – 01	2.53E – 01	**1.34E – 01**
DTLZ2	Mean	5.59E – 02	5.41E – 02	5.28E – 02	5.17E – 02	**4.98E – 02**
	SD	2.89E – 03	3.06E – 03	2.17E – 03	2.23E – 03	**2.14E – 03**
DTLZ3	Mean	9.50E – 01	5.64E – 01	2.31E – 01	1.25E – 01	**8.07E – 02**
	SD	3.92E – 01	4.74E – 01	2.21E – 01	2.40E – 02	**1.63E – 02**
DTLZ4	Mean	6.08E – 02	8.80E – 02	1.15E – 01	5.65E – 02	**5.47E – 02**
	SD	4.37E – 03	2.39E – 02	4.06E – 02	3.59E – 03	**3.14E – 03**

Table 4 Results comparison on five UF problems

Problem	IGD	NSGA-II	MOEA/D	MOEA/D-DE	MOEA/D-AGR	MOEA/D-EAM
UF1	Mean	1.07E – 01	8.83E – 02	7.33E – 02	4.64E – 02	**3.62E – 02**
	SD	3.41E – 02	1.38E – 02	1.64E – 02	**8.88E – 03**	1.02E – 02
UF 2	Mean	3.41E – 02	3.29E – 02	3.00E – 02	2.79E – 02	**2.22E – 02**
	SD	1.30E – 02	6.39E – 03	6.00E – 03	5.34E – 03	**3.35E – 03**
UF 3	Mean	1.41E – 01	1.21E – 01	**1.05E – 01**	1.37E – 01	1.10E – 01
	SD	**1.26E – 02**	1.82E – 02	1.81E – 02	4.11E – 02	2.00E – 02
UF 4	Mean	8.29E – 02	6.30E – 02	7.72E – 02	8.34E – 02	**4.55E – 02**
	SD	6.02E – 03	3.58E – 03	4.79E – 03	4.97E – 03	**1.74E – 03**
UF 5	Mean	**2.84E – 01**	4.47E – 01	4.77E – 01	4.24E – 01	3.23E – 01
	SD	**5.59E – 02**	8.16E – 02	9.44E – 02	9.03E – 02	8.77E – 02

4.4.1 Experimental results on MOPs with simple PS shape: ZDT and DTLZ

As shown in Tables 2 and 3, MOEA/D-EAM performs the best on ZDT2, ZDT4 and other four DTLZ test problems, and it also acquires better results on ZDT1 and ZDT3. Figures 4, 5 and 6 show the final solutions with lowest IGD-metric values obtained by all algorithms after 20 runs on ZDT4, ZDT6 and DTLZ1, which are three nonconvex and multimodal test problems. It is obvious that the solutions found by MOEA/D-EAM are more approximate to the PF and these solutions maintain a decent diversity as well.

The reason for these results is that each solution in MOEA/D-EAM searches in the direction to its nearest archive solution, and then, each solution's convergence rate gets improved.

Besides that, in MOEA/D-EAM, to avoid solutions becoming more similar to its nearest archive solutions which discourages diversity, a perturbed learning strategy is utilized to the process of offspring generating, solutions will search in the direction to other archive solutions under a certain probability.

To verify the efficiency of external archive matching strategy and perturbed learning strategy of MOEA/D-EAM, we compare the evolution process of the mean IGD-metric values of MOEA/D-EAM with other algorithms that have got good results. As shown in Fig. 7, benefiting from the external archive matching strategy, solutions in MOEA/D-EAM approximate the PF quickly. In addition, the perturbed learning strategy supports the solutions in MOEA/D-EAM maintain a good diversity. For the reason of the mutual effect of these strategies, the mean IGD-metric values obtained by MOEA/D-EAM decrease more quickly than those obtained by MOEA/D-AGR.

Moreover, MOEA/D-AGR performs better on six out of the nine problems and shows a best performance on ZDT1. MOEA/D, MOEA/D-DE and NSGA-II show a worse ability than MOEA/D-EAM on most test problems. NSGA-II only obtains the best result on ZDT3, the reason for the other algorithms' poor performance is that the PF of ZDT3 in objective space is disconnected, and the process of updating the reference point in these decomposition-based MOEAs may slow down the evolutionary rate of solutions. MOEA/D only obtains the best result on ZDT6, and other algorithms show a similar performance to MOEA/D on ZDT6.

In summary, the proposed algorithm MOEA/D-EAM outperforms other algorithms in dealing with ZDT and DTLZ problems, since the PS shape of these problems is simple, solutions in MOEA/D-EAM can evolve quickly by searching in the direction to their most matching archive solutions. The experimental results have verified the efficiency of these proposed strategies.

4.4.2 Experimental results on MOPs with complicated PS shape: UF

In this section, we compare the performance of these five algorithms on UF test bio-objective problems, whose PS shape is complicated. Table 4 presents the mean and standard deviation values of IGD-metric. It is clear that, in terms of IGD-metric values, MOEA/D-EAM gets the best results on UF1, UF2 and UF4, and it also gets the second best results on UF3 and UF5. MOEA/D-AGR performs well on UF1 and UF2. MOEA/D-DE acquires a lowest IGD-metric value on UF3, the reason MOEA/D-DE performs better than MOEA/D-EAM may be that the offspring solutions in MOEA/D-EAM are generated based on their parent solutions' most matching archive solutions, since the PS shape of UF test problems is complicated, the offspring solutions may not always be excellent compared to their parent solutions. NSGA-II outperforms other decomposition-based MOEAs on UF5.

Figures 8, 9, 10, 11 and 12 plot the final solutions with lowest IGD-metric values obtained by all algorithms after

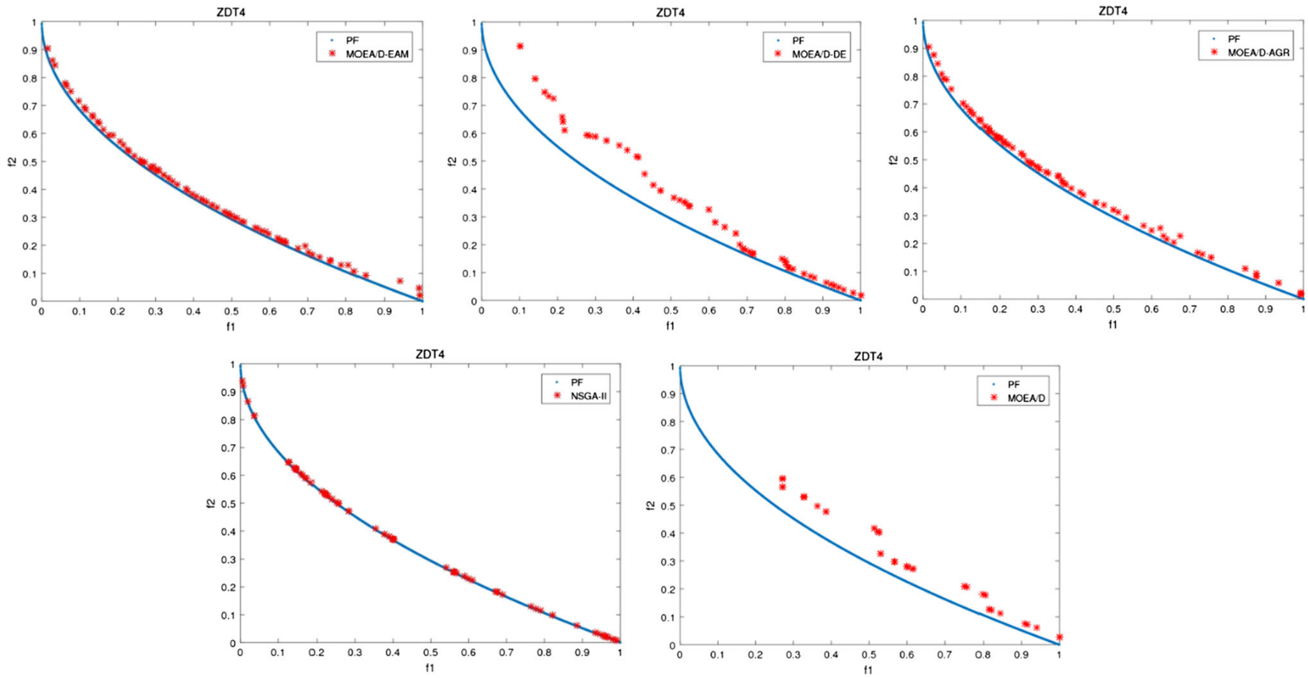

Fig. 4 The distribution of the final solutions with the lowest IGD-metric values on ZDT4

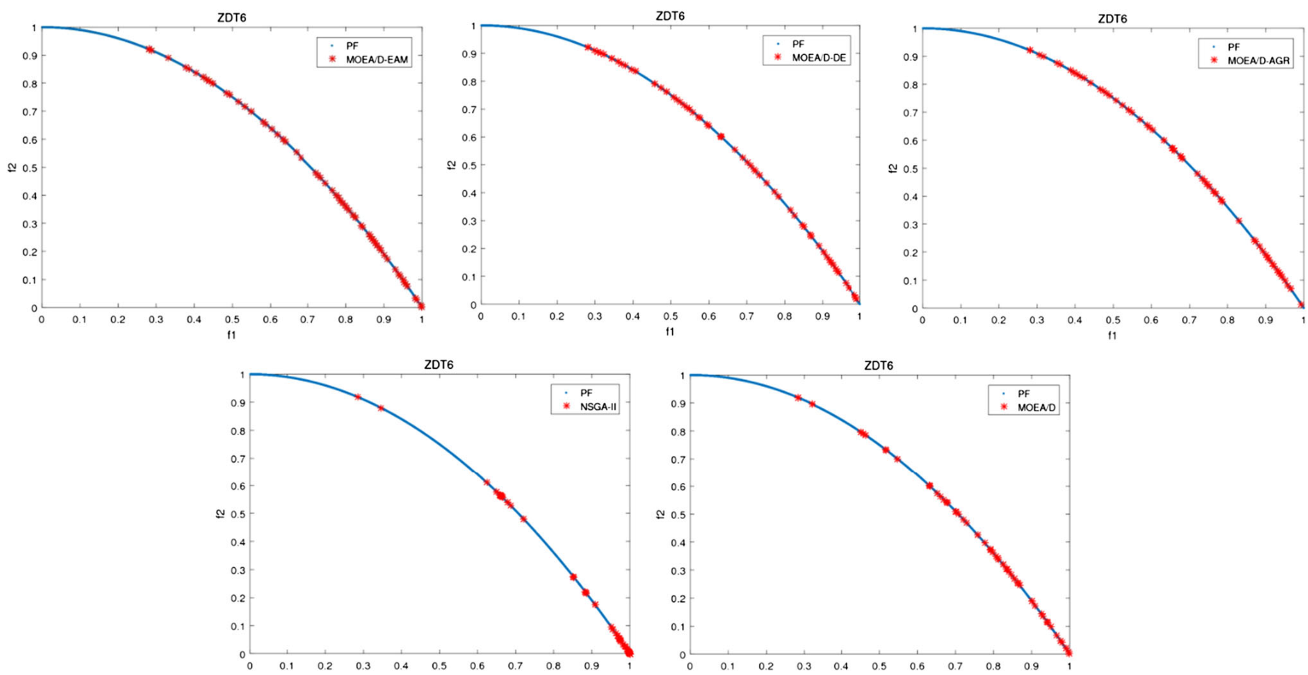

Fig. 5 The distribution of the final solutions with the lowest IGD-metric values on ZDT6

20 runs on UF1, UF2, UF3, UF4 and UF5. Obviously, on UF1, UF2, and UF4, MOEA/D-EAM finds a majority of non-dominated solutions along PF and these solutions maintain an excellent convergence and diversity as well. Because of the complicated PS shape, some UF problems hold a number of Pareto optimal solutions of these problems that are extremely hard to find. As shown in Fig. 10, these five algorithms cannot find a set of solutions that approximate to the PF within a limited number of iterations.

Furthermore, since UF5 problem has dispersed Pareto optimal solutions, the effect of evolution information

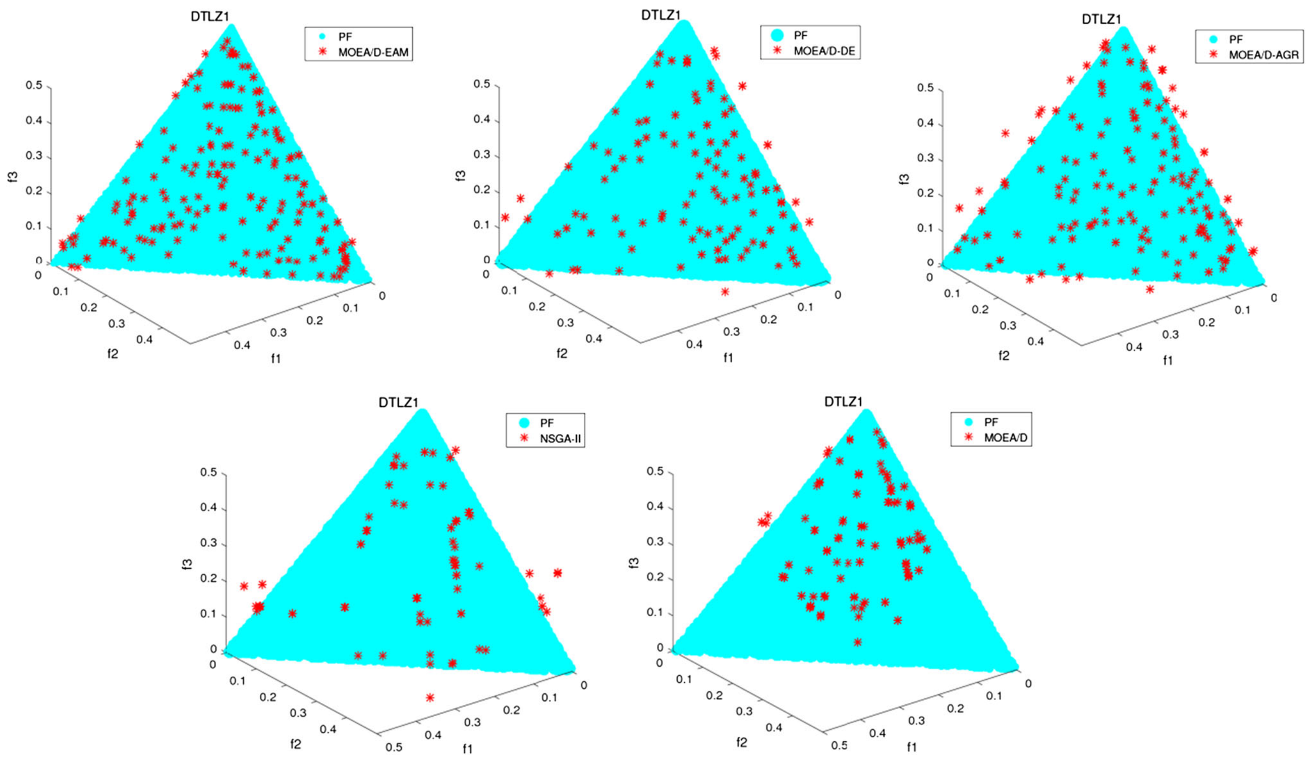

Fig. 6 The distribution of the final solutions with the lowest IGD-metric values on DTLZ1

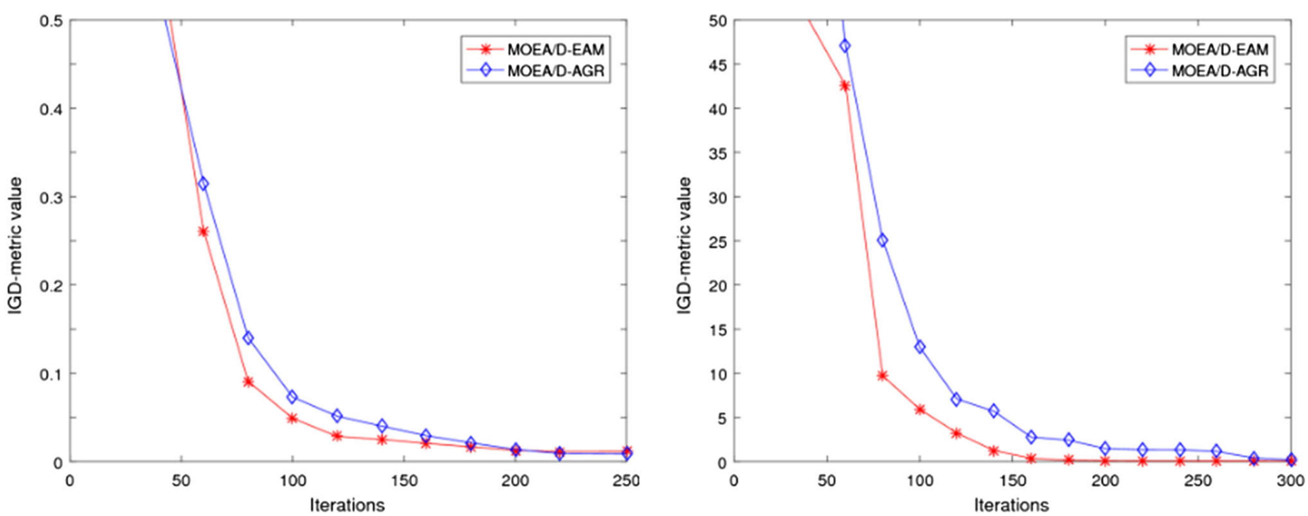

Fig. 7 Evolutionary process of mean IGD-metric values obtained by MOEA/D-EAM and MOEA/D-AGR; the left plot is the result on ZDT1, and the right plot is the result on DTLZ3

exchanged between neighbors in decomposition-based MOEAs is reduced.

However, benefiting from the external archive matching strategy, solutions found by MOEA/D-EAM maintain a good convergence compared with others. Besides that, offspring solutions in MOEA/D-DE and MOEA/D-AGR are generated using their neighbor solutions or the global solutions under a certain probability, so the solutions obtained by these two algorithms maintain a good diversity similar to MOEA/D-EAM. Since the solutions in MOEA/D only utilize their neighbor solutions as parent solutions, the evolutionary efficiency is not as good as MOEA/D-EAM.

Overall, the proposed MOEA/D-EAM shows a competitive performance in solving UF problems. It outperforms other algorithms on three of the five UF problems. The capability of MOEA/D-EAM has been validated.

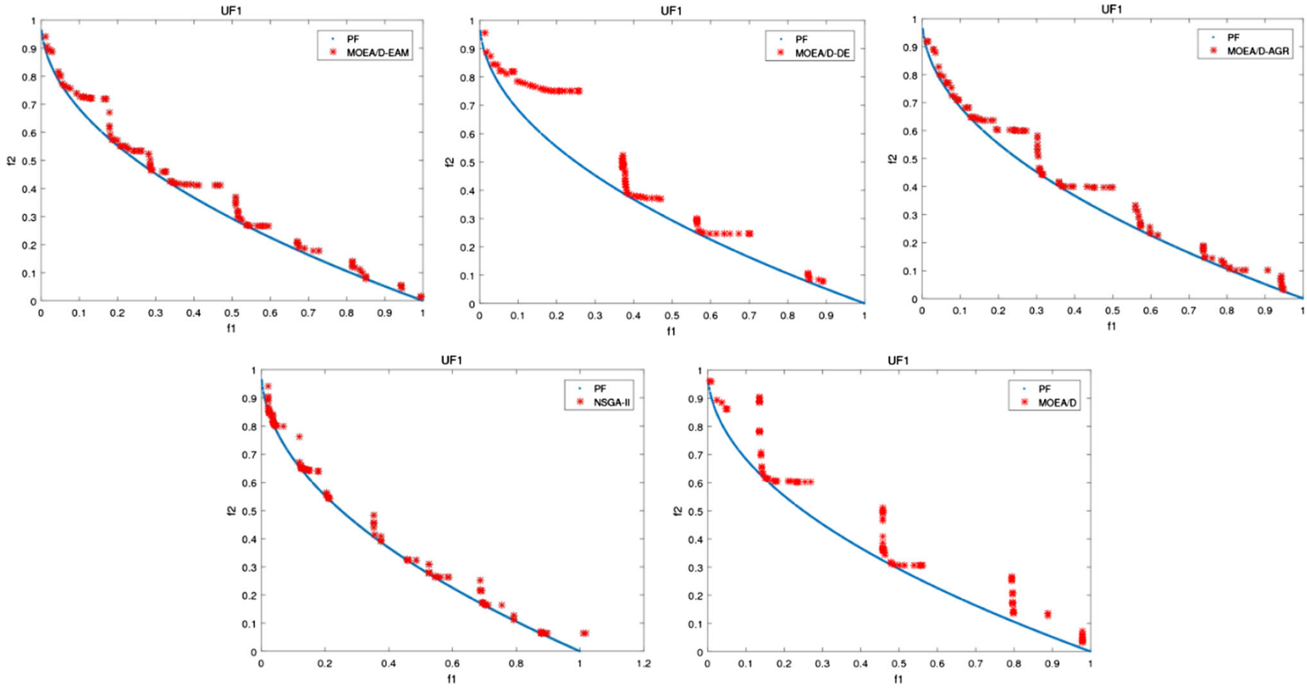

Fig. 8 The distribution of the final solutions with the lowest IGD-metric values on UF1

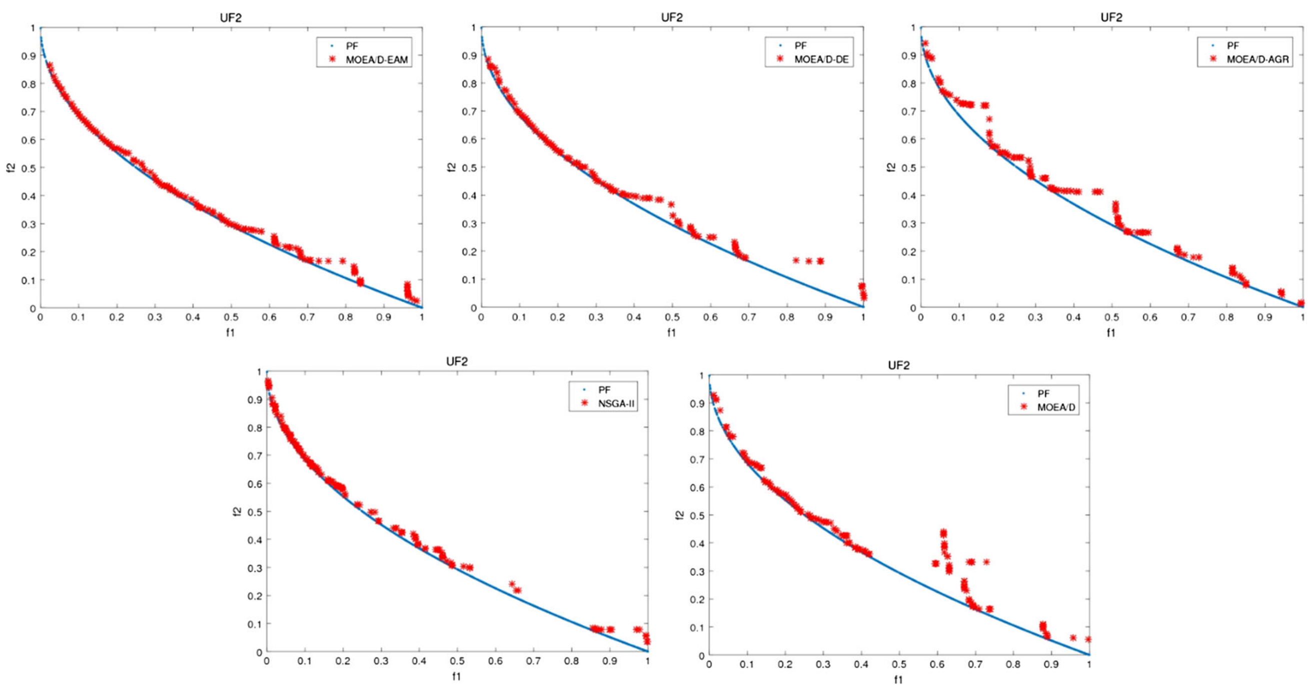

Fig. 9 The distribution of the final solutions with the lowest IGD-metric values on UF2

5 Conclusions

In this paper, we present a new reproduction approach for MOEA/D, namely MOEA/D-EAM. As mentioned above, the basic reproduction process utilized in original MOEA/D may slow down the convergence rate. To improve the performance of MOEA/D, we maintain an external archive to record non-dominated solutions and take these solutions as parent solutions to generate excellent offspring solutions. To ensure the generated offspring solutions' quality, we propose an external archive matching strategy, where the solutions are only allowed to mate with their most matching archive

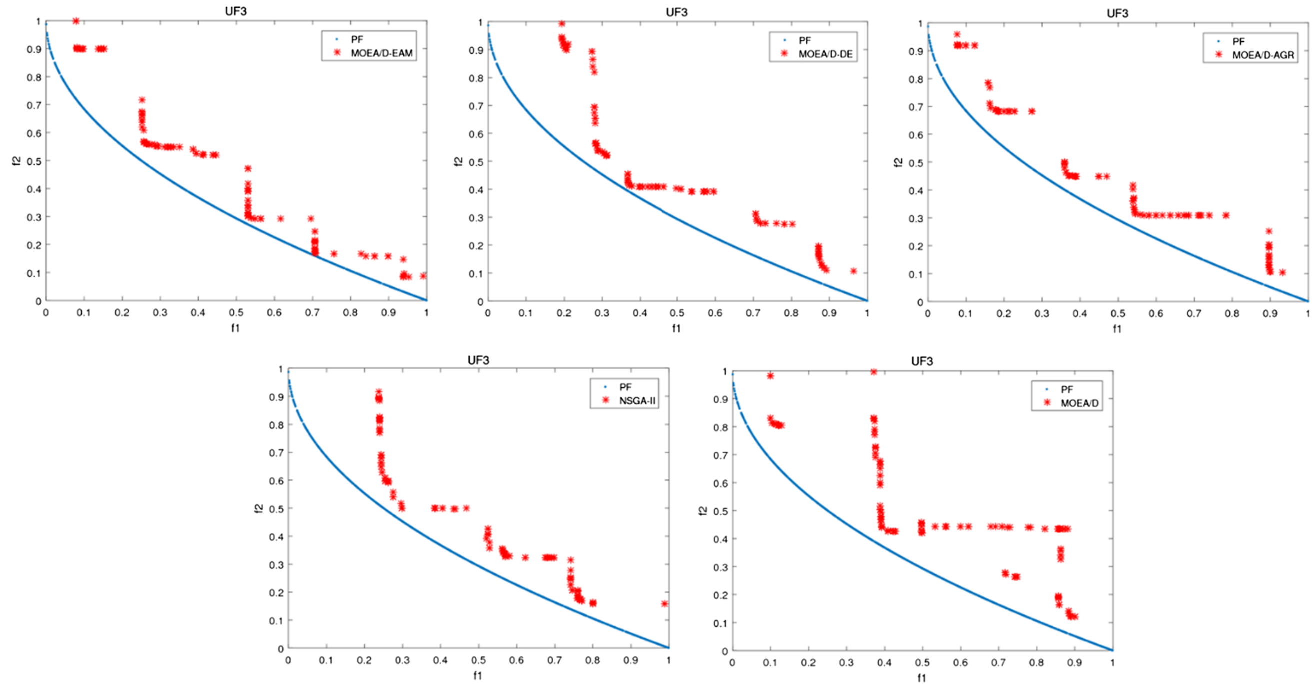

Fig. 10 The distribution of the final solutions with the lowest IGD-metric values on UF3

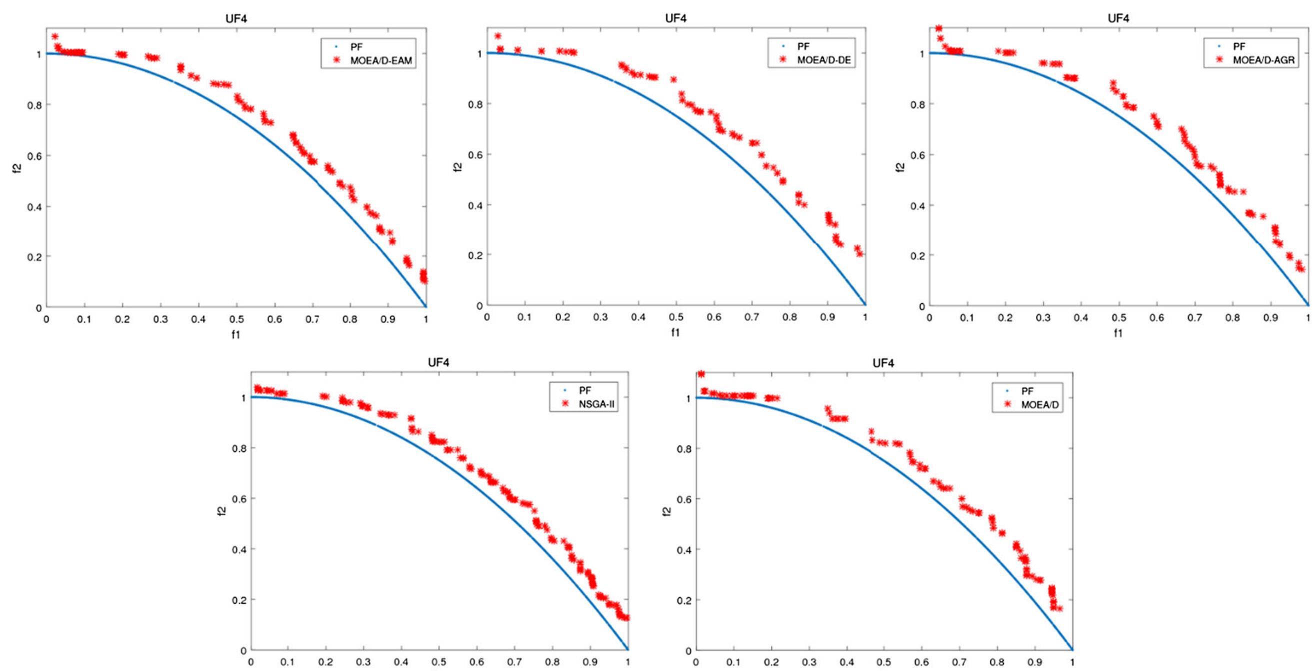

Fig. 11 The distribution of the final solutions with the lowest IGD-metric values on UF4

solutions. As a result of this, the search space of solutions is around archive solutions, the ability of exploitation gets improved. To balance convergence and diversity, a perturbed learning scheme is utilized in MOEA/D-EAM, offspring solutions would be generated by some randomly selected archive solutions to extend the search space of solutions. To verify the performance of MOEA/D-EAM, we compare MOEA/D-EAM with other state-of-the-art algorithms on three groups of test problems. The experimental results show that MOEA/D-EAM can get solutions that are more approximate to PF. Furthermore, the mean IGD-metric values obtained by MOEA/D-EAM are lower than others on the

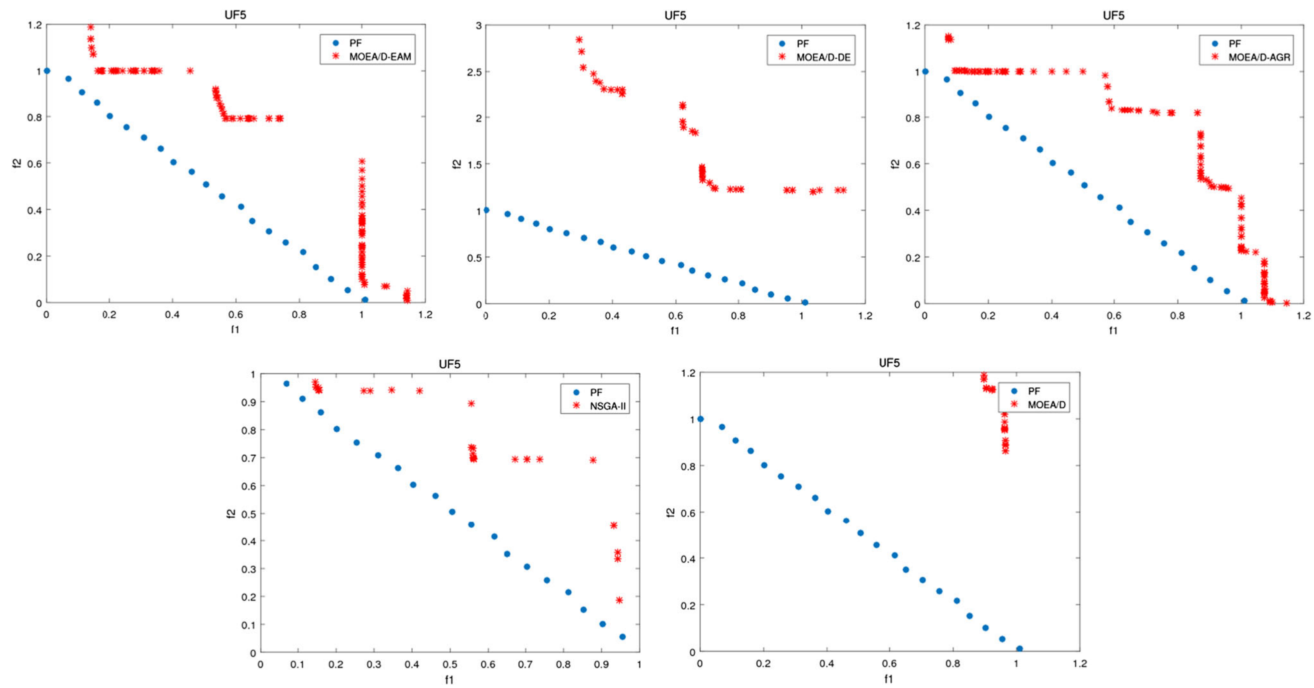

Fig. 12 The distribution of the final solutions with the lowest IGD-metric values on UF5

whole, which means the solutions obtained by MOEA/D-EAM maintain a better convergence and diversity as well.

In the future, we will consider other components of the OEA/D algorithm and then study how to ameliorate them. Another related future work is to investigate the applications of this MOEA/D-EAM into solving various multiobjective problems in reality.

Acknowledgements This work is supported by National Nature Science Foundation of China (Grant No. 61773296); the 111 Programme of Introducing Talents of Discipline to Universities (Grant No. B07037); the Fundamental Research Funds for the Central Universities (Grant No. 2042018kf0224); and Research Fund for Academic Team of Young Scholars at Wuhan University (Grant No. Whu2016013).

Compliance with ethical standards

Conflict of interest All authors declare that they have no conflict of interest.

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

Informed consent Informed consent was obtained from all individual participants included in the study.

References

1. Alsmirat MA, Jararweh Y, Obaidat I, Gupta BB (2017) Automated wireless video surveillance: an evaluation framework. J Real Time Image Process 13(3):527–546
2. Alzain MA, Li AS, Soh B, Pardede E (2015) Multi-cloud data management using shamir's secret sharing and quantum byzantine agreement schemes. Int J Cloud Appl Comput 5(3):35–52
3. Beng KCT, Beng (2004) Multiobjective evolutionary algorithms and applications, vol 45, no 11. Springer, Berlin, pp 2281–2293
4. Beume N, Naujoks B, Emmerich M (2007) Sms-emoa: multiobjective selection based on dominated hypervolume. Eur J Oper Res 181(3):1653–1669
5. Cao Y, Zhou Z, Sun X, Gao C (2018) Coverless information hiding based on the molecular structure images of material. Comput Mater Contin 54(2):197–207
6. Coello CAC, Corts NC (2005) Solving multiobjective optimization problems using an artificial immune system. Genet Program Evolv Mach 6(2):163–190
7. Deb K, Beyer HG (2001) Self-adaptive genetic algorithms with simulated binary crossover. Evol Comput 9(2):197–221
8. Deb K, Kalyanmoy D (2001) Multi-objective optimization using evolutionary algorithms. Wiley, Hoboken
9. Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: Nsga-ii. IEEE Trans Evol Comput 6(2):182–197
10. Deb K, Thiele L, Laumanns M, Zitzler E (2005) Scalable test problems for evolutionary multiobjective optimization. In: Evolutionary multiobjective optimization, pp 105–145
11. Gupta BB, Agrawal DP, Yamaguchi S (2016) Handbook of research on modern cryptographic solutions for computer and cyber security. IGI Global, Hershey
12. He P, Deng Z, Wang H, Liu Z (2016) Model approach to grammatical evolution: theory and case study. Soft Comput 20(9):3537–3548
13. Hillermeier C (2001) Nonlinear multiobjective optimization. Birkhäser Verlag, Basel
14. Huang Y, Li W, Liang Z, Xue Y, Wang X (2016) Efficient business process consolidation: combining topic features with structure matching. Soft Comput 22(2):645–657
15. Ishibuchi H, Sakane Y, Tsukamoto N, Nojima Y (2009) Effects of using two neighborhood structures on the performance of cel-

lular evolutionary algorithms for many-objective optimization. In: Eleventh conference on congress on evolutionary computation, pp 2508–2515

16. Ishibuchi H, Akedo N, Nojima Y (2013) Relation between neighborhood size and moea/d performance on manyobjective problems. In: International conference on evolutionary multi-criterion optimization, pp 459–474
17. Lai X, Zou W, Xie D, Li X, Fan L (2017) DF relaying networks with randomly distributed interferers. IEEE Access 5:18909–18917
18. Li H, Zhang Q (2009) Multiobjective optimization problems with complicated pareto sets, MOEA/D and NSGA-II. IEEE Trans Evol Comput 13(2):284–302
19. Li H, Ding M, Deng J, Zhang Q (2015) On the use of random weights in MOEA/D. In: IEEE Congress on Evolutionary Computation (CEC), pp 978–985
20. Li Y, Peng Z, Liang D, Chang H, Cai Z (2016) Facial age estimation by using stacked feature composition and selection. Vis Comput Int J Comput Graph 32(12):1525–1536
21. Li P, Li J, Huang Z, Li T, Gao Cz, Chen WB, Chen K (2017a) Privacy-preserving outsourced classification in cloud computing. Cluster Comput 21(1):1–10
22. Li P, Li J, Huang Z, Li T, Cz Gao, Yiu SM, Chen K (2017b) Multi-key privacy-preserving deep learning in cloud computing. Future Gener Comput Syst 74:76–85
23. Li W, Li K, Huang Y, Yang S, Yang L (2017c) A eaand aca-based qos multicast routing algorithm with multiple constraints for ad hoc networks. Soft Comput 21(19):5717–5727
24. Li W, Li S, Chen Z, Zhong L, Ouyang C (2017d) Self feedback differential evolution adapting to fitness landscape characteristics. Soft Comput. https://doi.org/10.1007/s00500-017-2833-y
25. Li W, Li K, Guo L, Huang Y, Xue Y (2018a) A new validity index adapted to fuzzy clustering algorithm. Multimed Tools Appl 77(3):1–23
26. Li Y, Wang G, Nie L, Wang Q, Tan W (2018b) Distance metric optimization driven convolutional neural network for age invariant face recognition. Pattern Recognit 75:51–62
27. Lin W, Xu S, He L, Li J (2017a) Multi-resource scheduling and power simulation for cloud computing. Inf Sci 397(C):168–186
28. Lin W, Xu SY, Li J, Xu L, Peng Z (2017b) Design and theoretical analysis of virtual machine placement algorithm based on peak workload characteristics. Soft Comput 21(5):1301–1314
29. Liu HL, Gu F, Zhang Q (2014) Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems. IEEE Trans Evol Comput 18(3):450–455
30. Qi Y, Ma X, Liu F, Jiao L, Sun J, Wu J (2014) MOEA/D with adaptive weight adjustment. Evol Comput 22(2):231–264
31. Sato H (2015) Analysis of inverted pbi and comparison with other scalarizing functions in decomposition based moeas. J Heuristics 21(6):819–849
32. Schutze O, Esquivel X, Lara A, Coello CAC (2012) Using the averaged hausdorff distance as a performance measure in evolutionary multiobjective optimization. IEEE Trans Evol Comput 16(4):504–522
33. Srinivas N, Deb K (2014) Muiltiobjective optimization using nondominated sorting in genetic algorithms. Evol Comput 2(3):221–248
34. Wang F, Zhang Y, Rao Q, Li K, Zhang H (2017a) Exploring mutual information-based sentimental analysis with kernel-based extreme learning machine for stock prediction. Soft Comput 21(12):3193–3205
35. Wang Y, Li K, Li K (2017b) Partition scheduling on heterogeneous multicore processors for multi-dimensional loops applications. Int J Parallel Program 45:1–26
36. Wang Z, Zhang Q, Zhou A, Gong M, Jiao L (2017c) Adaptive replacement strategies for MOEA/D. IEEE Trans Cybern 46(2):474–486
37. Wang F, Zhang H, Li K, Lin Z, Yang J, Shen XL (2018a) A hybrid particle swarm optimization algorithm using adaptive learning strategy. Inf Sci 436–437:162–177
38. Wang H, Wang W, Cui Z, Zhou X, Zhao J, Li Y (2018b) A new dynamic firefly algorithm for demand estimation of water resources. Inf Sci 438:95–106
39. Wu H, Kuang L, Wang F, Rao Q, Gong M, Li Y (2017) A multiobjective box-covering algorithm for fractal modularity on complex networks. Appl Soft Comput 61:294–313
40. Xie D, Lai X, Lei X, Fan L (2018) Cognitive multiuser energy harvesting decode-and-forward relaying system with direct links. IEEE Access 6:5596–5606
41. Yuan C, Li X, Wu QMJ, Li J, Sun X (2017) Fingerprint liveness detection from different fingerprint materials using convolutional neural network and principal component analysis. Comput Mater Contin 53(3):357–371
42. Zhang Q, Li H (2007) MOEA/D: a multiobjective evolutionary algorithm based on decomposition. IEEE Trans Evol Comput 11(6):712–731
43. Zhang Q, Zhou A, Zhao S, Suganthan PN, Liu W, Tiwari S (2008) Multiobjective optimization test instances for the CEC 2009 special session and competition. University of Essex, Colchester
44. Zhang H, Zhou A, Song S, Zhang Q, Gao XZ, Zhang J (2016) A self-organizing multiobjective evolutionary algorithm. IEEE Trans Evol Comput 20(5):792–806
45. Zhang S, Yang Z, Xing X, Gao Y, Xie D, Wong HS (2017) Generalized pair-counting similarity measures for clustering and cluster ensembles. IEEE Access 5:16904–16918
46. Zhou J, Wang F, Xu J, Yan Y, Zhu H (2018) A novel character segmentation method for serial number on banknotes with complex background. J Ambient Intell Humaniz Comput. https://doi.org/10.1007/s12652-018-0707-5
47. Zitzler E, Deb K, Thiele L (2000) Comparison of multiobjective evolutionary algorithms: empirical results. Evol Comput 8(2):173–195