Soft Comput
DOI 10.1007/s00500-016-2314-8

METHODOLOGIES AND APPLICATION

Many-objective E-dominance dynamical evolutionary algorithm based on adaptive grid

Xuezhi Yue[1] · Zhaolu Guo[1] · Yali Yin[1] · Xiaosheng Liu[2]

© Springer-Verlag Berlin Heidelberg 2016

Abstract In evolutionary multi-objective optimization, achieving a balance between convergence speed and population diversity remains a challenging topic especially for many-objective optimization problems (MaOPs). To accelerate convergence toward the Pareto front and maintain a high degree of diversity for MaOPs, we propose a new many-objective dynamical evolutionary algorithm based on E-dominance and adaptive-grid strategies (EDAGEA). In EDAGEA, it incorporates the E_dominance and adaptive strategies to enhance the search ability. Instead of the Pareto dominance mechanism in the traditional dynamical evolutionary algorithm, EDAGEA employs the E-dominance strategy to improve the selective pressure and to accelerate the convergence speed. Moreover, EDAGEA incorporates the adaptive-grid strategy to promote the uniformity and diversity of the population. In the experiments, the proposed EDAGEA algorithm is tested on DTLZ series problems under 3–8 objectives with diverse characteristics and is compared with two excellent many-objective evolutionary algorithms. Experimental results demonstrate that the proposed EDAGEA algorithm exhibits competitive performance in terms of both convergence speed and diversity of population.

Communicated by V. Loia.

✉ Xuezhi Yue
9120000040@jxust.edu.cn

[1] School of Science, Jiangxi University of Science and Technology, Ganzhou 341000, China

[2] School of Architectural and Surveying and Mapping Engineering, Jiangxi University of Science and Technology, Ganzhou 341000, China

Keywords Many-objective · Dynamical evolutionary algorithm · Adaptive-grid · E-dominance

1 Introduction

In the real world, many optimization problems consist of several mutually dependent sub-problems that have to be optimized in parallel, which are called multi-objective optimization problems (MOPs). MOPs with at least four objectives are known as many-objective optimization problems (MaOPs) (Deb and Jain 2012; Li et al. 2015a). MaOPs have attracted great attention (Freire et al. 2014; Narukawa 2013) in the last few decades, which widely exist in many real-world applications, such as engineering design (Fleming et al. 2005), data deduplication (Li et al. 2014), air traffic control (Herrero et al. 2009), land use management (Chikumbo et al. 2012), vehicle classification (Wen et al. 2015), ordinal regression (Gu et al. 2014), image processing (Li et al. 2015b; Xia et al. 2014a, b; Zheng et al. 2015; Chen et al. 2014) and sensor networks (Xie and Wang 2014; Shen et al. 2015).

Over the past two decades, a large number of multi-objective evolutionary algorithms (MOEAs) have been developed and demonstrated to be promising performance for solving MOPs, e.g., NSGA-II (Deb et al. 2002), SPEA2 (Zitzler et al. 2001), PESA-II (Corne et al. 2002), MOEA/D (Zhang and Li 2007), and MOPSO (Coello et al. 2004). In all these MOEAs, a variety of selection strategies have been proposed to achieve fast convergence speed and good diversity of population, which play the most important role in determining the effectiveness and efficiency of the MOEA in obtaining the Pareto optimal solutions. But the efficiency of such Pareto-based MOEAs will seriously degrade for MaOPs. The main reason is largely due to the loss of selection pressure to drive the search toward the Pareto front and

Published online: 26 August 2016

the ineffective strategy to balance the convergence speed and population diversity. The proportion of non-dominated solutions in the population is increasing dramatically with the number of objectives increasing, which severely decreases the selective pressure and leads to the slow or stagnant convergence (Ishibuchi et al. 2008; Adra and Fleming 2011).

Particle dynamical multi-objective evolutionary algorithm (PDEA) is one of the classical MOEAs based on statistical mechanics theory (Li et al. 2003; Zou et al. 2002). In PDEA, the individual is regarded as a particle in phase space and the population is regarded as a particle system. PDEA simulates the particle system principle of particle phase space to conduct crossover and mutation operations, which makes the particle system from non-equilibrium to equilibrium (Li et al. 2007). Moreover, PDEA employs the Rank value mechanism to determine the dominance among the particles, which can accelerate the convergence speed. By using such mechanism, the rank of each particle in the population is unique. The experimental results indicate that PDEA can well converge to the Pareto front for 2 or 3 objectives optimization problems, but it is difficult to converge to the Pareto front and even suffering from premature due to the poor diversity of population for MaOPs. The main reason is that with the increase in the number of objectives, the probability of non-dominant between individuals is increasing, which leads to slow convergence or even stagnation.

Aiming at the deficiency of the dominance mechanism in the PDEA for MaOPs, we propose a new many-objective dynamical evolutionary algorithm based on E-dominance and adaptive grid (EDAGEA). In the proposed EDAGEA algorithm, the E-dominance and adaptive-grid strategies are implemented to improve the selective pressure and to accelerate the convergence of the population as well as to promote the diversity of population.

The rest of this paper is structured as follows. Section 2 introduces the related works mainly including E-dominance mechanism and adaptive-grid technique on improved PDEA. In Sect. 3, the framework of the proposed method is developed based on the proposed operations. The simulation results and comparisons with other algorithms are reported in Sect. 4. Finally, Sect. 5 concludes the paper.

2 Related works

2.1 E-dominance mechanism

Kang et al. (2007) proposed a new dominance mechanism, namely E-dominance mechanism, by analyzing the pros and cons of the latest dominance mechanisms. E-dominance combines the advantages of K-dominance and epsilon-dominance. It is a slack Pareto dominance and enhances the selection pressure of EMO algorithm for solving MaOPs, which accelerates the convergence of the EMO algorithm. The definitions of E-dominance are introduced as follows.

Definition 1 *(many-objective optimization problem)* The problem of m minimization functions is called as a many-objective optimization problem, which is formulated as follows (Kang et al. 2007):

$$\begin{aligned} \min F(x) &= (F_1(x), F_2(x), \ldots, F_m(x)) \\ \text{s.t.}: \ & g_i(x) \le 0, \quad i = 1, 2, \ldots, p \\ & h_j(x) = 0, \quad j = 1, 2, \ldots, q \end{aligned} \tag{1}$$

where $m \ge 4$ is the number of objectives, and F_i, g_i, h_j: $R^m \to R, x \in R^m$ as decision variables, $X = \{x | x \in R^m, g_i(x) \le 0, h_j(x) = 0, i = 1, 2, \ldots, p, j = 1, 2, \ldots, q\}$ is called the feasible domain of many-objective optimization problem.

Definition 2 *(B_t, W_s, E_q)* (Kang et al. 2007) $X_1, X_2 \in R^n$, in the objective space the objective number of $F(X_1)$ better than $F(X_2)$ in performance is denoted as $B_t(X_1, X_2)(B_t)$, where *card* is used to denote the potential of the set:

$$B_t(X_1, X_2) = \text{card}\,\{i | F_i(X_1) < F_i(X_2),\ i = 1, 2, \ldots, m\} \tag{2}$$

$X_1, X_2 \in R^n$, in the objective space the objective number of $F(X_1)$ worse than $F(X_2)$ in performance is denoted as $W_s(X_1, X_2)(W_s)$:

$$W_s(X_1, X_2) = \text{card}\,\{i | F_i(X_1) > F_i(X_2),\ i = 1, 2, \ldots, m\} \tag{3}$$

$X_1, X_2 \in R^n$, in the objective space when $F(X_1)$ and $F(X_2)$ have the same performance, the objective number is denoted as $E_q(X_1, X_2)(E_q)$:

$$E_q(X_1, X_2) = \text{card}\,\{i | F_i(X_1) = F_i(X_2),\ i = 1, 2, \ldots, m\} \tag{4}$$

Definition 3 *(many-objective vector fitness value)* (Kang et al. 2007) The 2-norm of objective vector $F(X)$ is denoted as fitness value of many-objective vector, which is defined as follows:

$$\|F(X)\| = \sqrt{\sum_{i=1}^{m} (F_i(X))^2} \tag{5}$$

Definition 4 *(E-dominance)* (Kang et al. 2007)

1. if $B_t - W_s > 0$, and $||F(X_1)|| < ||F(X_2)||$, then X_1 is said to E-dominate X_2, denoted as $X_1 \prec_E X_2$. When $W_s = 0$, E-dominance equivalent to Pareto dominance.

2. if $B_t - W_s = 0$, and $||F(X_1)|| \leq ||F(X_2)||$, then X_1 is said to weakly E-dominate X_2, denoted as $X_1 \prec=_E X_2$.
3. if $E_q = m$ (m is the number of objectives), then X_1 is said to E-equivalent of X_2, denoted as $X_1 \Leftrightarrow_E X_2$.
4. if $B_t - W_s \geq 0$, and $||F(X_1)|| > ||F(X_2)||$, then X_1 is unbiased order relation with X_2, denoted as $X_1 \sim_E X_2$.

Definition 5 *(dominance relationship between individual X and archive set M)* (Kang et al. 2007)

1. X dominates M if there is an individual $X' \in M$ such that $X \prec_E X'$;
2. X is dominated by M if there is an individual $X' \in M$ such that $X' \prec_E X$;
3. X is non-dominance relation with M if $\neg\exists X' \in M$ such that $X \prec_E X'$ or $X' \prec_E X$.

2.2 Adaptive-grid technique

In order to keep the distribution and diversity of the population, adaptive-grid method has been used in multi-objective evolutionary algorithms. In 1999, Knowles et al. proposed the areto archived evolution strategy for solving the approximating non-dominant front of multi-objective optimization problems (Knowles and Corne 2000), in which adaptive-grid strategy is firstly used to solve multi-objective optimization problems. Coello et al. (2004) adopted adaptive-grid strategy to maintain the diversity of solutions in multi-objective particle swarm optimization. In 2011, Chen et al. proposed multi-objective optimization algorithm based on quantum-behaved particle swarm and adaptive grid (Shi and Chen 2011). Yang et al. (2013) proposed a grid-based evolutionary algorithm for many-objective optimization.

Let the number of objectives be m for a multi-objective optimization problem, then the number of grid boundaries is 2 m. Grid boundaries include the lower bound lb_k and the upper bound ub_k ($k = 1, 2, \ldots, m$). A grid is identified by the two diagonal of hyper-box expressed as $(lb_1, lb_2, \ldots, lb_m)$ and $(ub_1, ub_2, \ldots, ub_m)$. A grid can be divided into several part areas which are called hyper-boxes. Mesh segmentation is determined by the population size and the number of the objectives. Whether an individual falls into a certain area is determined by the identification of the grid and each small area in grid. In the multi-objective evolutionary algorithm, grid boundaries are generally not fixed and may be different in each generation. Grid boundaries are adjusted adaptively according to the individual distribution of current generation, which is called adaptive-grid strategy. The following describes the two important contents in grid technology: grid boundaries and individual position in the grid.

2.2.1 Grid boundaries (Zheng 2007)

Denote each small region of grid as m^i, $i = (i_1, i_2, \ldots, i_m)$ and $i_k \in 1, 2, \ldots, d, d$ is a constant which represents the number of division in each dimension, range_k is the domain width of the k-th dimension, ω_k is the width of each small region and $\omega_k = \text{range}_k/d$. So corresponding to each m^i, the boundaries can be denoted as:

$$\text{rub}_{k,i} = [lb_k + (i_k/d)(ub_k - lb_k)]\,\omega_k \quad k = 1, 2, \ldots, m \tag{6}$$

$$rlb_{k,i} = [lb_k + ((i_k - 1)/d)(ub_k - lb_k)]\,\omega_k \quad k = 1, 2, \ldots, m \tag{7}$$

For adaptive-grid strategy, the domain width range_k of each dimension is denoted as:

$$\text{range}_k = \max\{z_k | z \in ARCH\} - \min\{z_k | z \in ARCH\} \tag{8}$$

where $ARCH$ is the archive set.

2.2.2 Individual position in the grid (Zheng 2007)

Set an individual $z = (z_1, z_2, \ldots, z_m)$ for the region m^i, if $rlb_{k,i} \leq z_k \leq \text{rub}_{k,i}$ for any $k = 1, 2, \ldots, m$, the individual z is deemed in the region m^i.

For multi-objective optimization problem, the individual is called as pole which makes a certain objective value minimally, denoted as z^{ext}. The pole needs to be distinguished with other individuals because its existence can make the distribution better of the evolution population. The pole is always distributed on the end point and the pole z^{ext} is denoted as:

$$z^{\text{ext}} = \{y \in ARCH \,|\, \exists k \in 1, 2, \ldots, r, \neg\exists z \in ARCH, z \neq y, z_k \leq y_k\} \tag{9}$$

To keep the distribution of evolution population, the individuals of large density aggregation in the grid are usually selected to remove. Sometimes, even the individuals in the archive set distribute relatively uniform, but due to the restriction of the archive set size, we should select a certain number of individuals to be deleted instead of pole.

3 Proposed EDAGEA

Based on the above detailed descriptions of the E-dominance mechanism and adaptive-grid technique, the proposed EDAGEA contains two key components: Rank value calculation algorithm and adaptive-grid region division algorithm. The detailed procedure of EDAGEA is described as follows.

3.1 Rank value calculation algorithm

The rule of Rank value is constructed by E-dominance, entropy and free energy. Each individual is assigned to a unique Rank value. In the selection operation, the individuals are sorted based on the Rank value. The Rank value calculation steps are described in Algorithm 1.

Algorithm 1: Rank value calculation

Step 1, Set the t-th generation population as $P_t=\{x_1(t), x_2(t), \ldots, x_N(t)\}$;

Step 2, Calculate the objective function value $Y(t)=\{y_1(t), y_2(t), \ldots, y_N(t)\}$ of all particles in the P_t, and set Rank($x_i(t)$)=0, i=1, 2, …, N;

Step 3, For the i–th particle x_i, i=1, 2,…,N; take another particle x_j, $1 \leqslant j \leqslant N \ \wedge \ j \neq i$, calculate the Rank value of x_i, rules are as follows:

if

$$((x_i \prec_E x_j) \vee (x_i \Leftarrow_E x_j) \vee (x_i \Leftrightarrow_E x_j \wedge s_i > s_j) \vee (x_i \Leftrightarrow_E x_j \wedge s_i = s_j \wedge p_i(t,f(t)) <= p_j(t,f(t))) \vee (x_i \sim_E x_j))$$

then

Rank($x_i(t)$)= Rank($x_i(t)$)+1, ,

else

Rank($x_i(t)$)= Rank($x_i(t)$);

endif

Step 4, Sort all particles in P_t from large to small according to the Rank value;

Step 5, Output Rank value.

3.2 Adaptive-grid region division algorithm

In adaptive-grid region division algorithm, the current non-dominant set is adaptively meshed according to the size of current generation of population and the number of objectives. For a new generated individual, whether it is recorded into the archive set is determined by its dominance relation with the individuals in the archive set. An individual whether be deleted is determined by the density of the region. Adaptive-grid region division algorithm steps are described in Algorithm 2.

Algorithm 2: Adaptive-grid region division

Step 1, Initialization:

Step 1.1 Set the non-dominant set as PF_t, the archive set as M_t in the t-th generation evolution.

Step 1.2 The number of divisions d in each dimension is a constant, set as 15.

Step 1.3 For the k–th objective(k=1,2,…,m), the domain width $range_{k,t}$, lower bound $lb_{k,t}$ and upper bound $ub_{k,t}$ are calculated in terms of the following formulas:

$$range_{k,t}=max\{F_k|F_k \in PF_t\}-min\{F_k|F_k \in PF_t\}$$

$$lb_{k,t}=min\{F_k|F_k \in PF_t\}-(1/2d)range_{k,t}$$

$$ub_{k,t}=max\{F_k|F_k \in PF_t\}+(1/2d)range_{k,t}\ ;$$

Step 2, For the new particles x_i'（i=1,2,…,l）:

if x_i' is dominated by $M_{t-1;}$ **then**

$M_t = M_{t-1}$;

elseif x_i' dominates $M_{t-1;}$ **then**

$M_t = M_{t-1} \cup \{ x_i' \}$ and delete all individuals dominated by x_i';

elseif x_i' is non dominance relation with M_{t-1}; **then**

if the archive set M_{t-1} is full; **then**

Replace the individual which has the maximum density in M_{t-1} by x_i';

else

$M_t = M_{t-1} \cup \{ x_i' \}$;

endif

endif

Step 3, Output the t-th archive set M_t, the algorithm is end.

3.3 EDAGEA

By incorporating the E-dominance and adaptive-grid strategies, the steps of EDAGEA are described in Algorithm 3.

Algorithm 3: Many-objective E-dominance dynamical evolutionary algorithm based on Adaptive-grid

Step 1, Set t=0, the random initialization population as $P_0=\{x_1(t), x_2(t), \ldots, x_N(t)\}$ and the maximum capacity of P_t as P;

Step 2, Calculate the Rank value of population P_0 according to **Algorithm 1**;

Step 3, **While** (stopping criterion is not met) **do**

Step 3.1, Conduct crossover and mutation operations by GuoTao Algorithm (Zheng 2007) in mating pool $\tilde{P}_t=\{\tilde{\mathbf{x}}_1(t),\tilde{\mathbf{x}}_2(t),\cdots,\tilde{\mathbf{x}}_N(t)\}$, produce l new particles as follows:

for i=1: l **do**

$$x_i'=\sum_{j=1}^{M} a_j\tilde{\mathbf{x}}_j(t)$$

$$-0.5\le a_j\le 1.5,\sum_{j=1}^{M} a_j=1$$

endfor

Step 3.2, For the new particles x_i' (i=1,2,…,l)

if $|P_t|<P$; **then**

$P_t'=P_t\cup\{x_i'\}$, turn to **Step 3.3;**

else

Generate the archive set M_t of the next generation by **Algorithm 2**;

endif

Step 3.3, Calculate the Rank value of the population P_t' by **Algorithm 1**, sort it as next generation P_{t+1}, Set $M_t=P_{t+1}$.

Step 3.4, Set t=t+1 , N=N+l

endwhile

Step 4, Output P_t, Y_t , the algorithm is end.

In EDAGEA, the E-dominance strategy is a slack Pareto dominance which can enhance the selective pressure and accelerate the convergence speed when solving many-objective optimization problems. Moreover, the adaptive-grid mechanism in EDAGEA can maintain the distribution and diversity of the population.

4 Analysis and simulation

4.1 Test problems

In order to verify the performance of the proposed algorithm, 7 test problems are used in the experiments, which is constructed by Deb et al. (2002, 2005). The main features of these test problems are scalable, namely the number of objectives can be specified. So far, DTLZ series test problems are widely used to test the performance of the many-objective optimization algorithms. These test problems have been widely used in many literatures and have become standard test problems for many-objective optimization problems.

4.2 Performance evaluation criteria

In order to evaluate the convergence, distribution and diversity of the approximate Pareto optimal solution, three criteria including generational distance (Veldhuizen and Lamont 2000), hyperarea (Zitzler and Thiele 1999) and spacing (Schott 1995) are used. They are described in detail as follows.

1. Generational Distance (GD)
Generational Distance (GD) was proposed by Veldhuizen and Lamont (2000) for evaluating the approach degree between the two important parameters PF_{known} and PF_{true}, the calculation formula as follows:

$$GD=\left(\frac{1}{n_{PF}}\sum_{i=1}^{n_{PF}} d_i^2\right)^{\frac{1}{2}} \tag{10}$$

where n_{PF} is the number of solutions in PF_{known}, d_i is the Euclidean distance between the i-th solution in objective space and the nearest solution in PF_{true}.

If $GD=0$, that is $PF_{\text{known}}=PF_{\text{true}}$; if $GD\neq 0$, that is PF_{known} deviating from PF_{true}; GD is smaller, PF_{known} is closer to PF_{true}. The main advantage of GD is simple in calculation, practical and suitable for comparing between multi algorithms.

2. Hyperarea (H)
Hyperarea (H) was proposed by Zitzler and Thiele (1999). It refers to a part of the objective space (or called curve space) which is covered by PF_{known}. It is defined as follows:

$$H=\left\{\bigcup_i a_i|v_i\in PF_{\text{known}}\right\} \tag{11}$$

where v_i is a non-dominant vector in PF_{known}, a_i is a high-dimensional space formed by source point (usually chosen the minimum value of the objective function) and v_i. When the GD of two solution sets is equal, the value of H is bigger; the diversity of solution set is better.

3. Spacing (S)
Spacing (S) is used to measure the uniformity of solution distribution (Schott 1995), it is defined as follows:

$$S=\left[\frac{1}{n_{PF}-1}\sum_{i=1}^{n_{PF}}\left(d_i-\bar{d}\right)^2\right]^{\frac{1}{2}} \tag{12}$$

where d_i is the Euclidean distance between the i-th solution in the objective space and the nearest solution in the PF_{true}. $\bar{d}$ is the average of all d_i, namely $\bar{d} = \frac{1}{n_{PF}} \sum_{i=1}^{n_{PF}} d_i$. All individuals in the PF_{known} distribute uniform when $S = 0$. The value of S is larger; the distribution of solution set is more uneven.

4.3 Simulating environment

Simulating environment of EDAGEA is set as follows:

- System: Windows XP professional edition;
- CPU: Intel ® Core ™ Duo CPU E8500 (3.16 GHz);
- Memory: 4G;
- Programming Language: C++;
- Development Environment: Microsoft Visual C++ 6.0.

4.4 Parameter settings

For many-objective optimization problems, the population size is very important. The population cannot converge to the Pareto optimal front if the population size is too small. To eliminate the influence of population size on evolutionary optimization results, the population size increases with the increasing in the number of objectives and variables. The main parameter settings are described in Table 1.

To validate the effectiveness of the proposed EDAGEA, we compare EDAGEA with two excellent algorithms, i.e., HN (Deb and Jain 2012), MOPSO (Coello et al. 2004). To have a fair comparison, we set the common parameters of HN, MOPSO and EDAGEA as shown in Table 1. We take the same values as their original papers for the other parameters of HN, MOPSO.

4.5 Experimental results and analysis

For DTLZ1-DTLZ7, each test problem is selected 3–8 objectives and is conducted 10 times, calculating the mean and variance of *GD*, *H* and *S*. However, it is worthy to notice that EDAGEA and HN are unstable for DTLZ3, DTLZ5 and DTLZ6 because that the variance is bigger than the corresponding mean in most cases in the experiments. So the results are shown only for DTLZ1, DTLZ2, DTLZ4 and DTLZ7 as Tables 2, 3, 4 and 5.

The results are listed from Tables 2, 3, 4 and 5 achieved by EDAGEA, HN and MOPSO on test problems of DTLZ1, DTLZ2, DTLZ4, DTLZ7 with 3–8 objectives. From the GD-mean, we can know that EDAGEA and HN can converge

Table 1 Parameter settings of EDAGEA in DTLZ1-7 problem for 3–8 objectives

Objective number (argument number)	3(6)	4(6)	5(8)	6(8)	7(10)	8(10)
Initial population size	100	100	200	200	300	300
Population size	400	400	500	500	700	700
Adaptive set size	100	100	100	100	100	100
Parents number (Guo Tao)	10	10	15	15	20	20
Offspring number (Guo Tao)	30	30	40	40	50	50
Iteration number	200	200	300	300	400	400

Table 2 Results of EDAGEA, HN and MOPSO to solve 3–8 objectives DTLZ1 test problems

N	*GD*-mean (*GD*-variance)			*S*-mean (*S*-variance)			*H*-mean (*H*-variance)		
	EDAGEA	HN	MOPSO	EDAGEA	HN	MOPSO	EDAGEA	HN	MOPSO
3	1.35×10^{-3}	3.16×10^{-2}	2.99×10^{-2}	1.48×10^{-2}	1.33×10^{-2}	1.7×10^{-2}	2.23×10^{-2}	1.2×10^{-2}	1.52×10^{-2}
	(9.91×10^{-4})	(6.21×10^{-3})	(2.71×10^{-3})	(3.15×10^{-3})	(2.1×10^{-3})	(1.4×10^{-2})	(2.3×10^{-3})	(2.3×10^{-3})	(1.56×10^{-3})
4	6.58×10^{-3}	5.16×10^{-2}	1.68×10^{-2}	2.64×10^{-2}	2.58×10^{-2}	2.89×10^{-2}	3.48×10^{-2}	3.29×10^{-2}	1.3×10^{-2}
	(5.09×10^{-4})	(3.3×10^{-3})	(1.71)	(3.52×10^{-3})	(4.5×10^{-3})	(2.27×10^{-2})	(2.13×10^{-3})	(5.31×10^{-3})	(2.2×10^{-3})
5	3.24×10^{-3}	3.16×10^{-2}	2.98	2.58×10^{-2}	2.54×10^{-2}	2.7×10^{-2}	2.99×10^{-2}	7.29×10^{-3}	1.1×10^{-3}
	(8.91×10^{-4})	(6.21×10^{-3})	(3.82)	(5.5×10^{-3})	(8.15×10^{-3})	(2.18×10^{-2})	(6.14×10^{-3})	(5.31×10^{-3})	(1.5×10^{-3})
6	6.25×10^{-3}	3.57×10^{-2}	3.78	2.78×10^{-2}	3.62×10^{-2}	3.57×10^{-2}	1.65×10^{-2}	6.3×10^{-3}	1.97×10^{-3}
	(5.9×10^{-4})	(3.18×10^{-3})	(1.60)	(5.80×10^{-3})	(6.5×10^{-3})	(3.18×10^{-2})	(6.5×10^{-3})	(7.1×10^{-3})	(8.27×10^{-3})
7	9.49×10^{-3}	6.27×10^{-2}	2.52	2.39×10^{-2}	2.57×10^{-2}	3.33×10^{-2}	8.22×10^{-2}	7.8×10^{-2}	2.82×10^{-2}
	(9.2×10^{-3})	(8.34×10^{-2})	(3.03)	(5.4×10^{-3})	(7.70×10^{-3})	(3.23×10^{-2})	(3.4×10^{-3})	(9.1×10^{-3})	(2.1×10^{-3})
8	1.41×10^{-2}	3.1×10^{-2}	5.84	8.52×10^{-3}	7.74×10^{-2}	8.1×10^{-2}	7.52×10^{-2}	7.43×10^{-2}	8.62×10^{-2}
	(1.4×10^{-3})	(2.93×10^{-2})	(6.52)	(1.37×10^{-3})	(9.21×10^{-3})	(6.93×10^{-2})	(6.7×10^{-3})	(7.7×10^{-3})	(8.7×10^{-3})

Table 3 Results of EDAGEA, HN and MOPSO to solve 3–8 objectives DTLZ2 test problems

N	*GD*-mean (*GD*-variance)			*S*-mean (*S*-variance)			*H*-mean (*H*-variance)		
	EDAGEA	HN	MOPSO	EDAGEA	HN	MOPSO	EDAGEA	HN	MOPSO
3	1.1×10^{-3}	2.8×10^{-3}	6.1×10^{-4}	4.3×10^{-2}	4.6×10^{-2}	1.3×10^{-1}	5.37×10^{-1}	4.32×10^{-1}	4.9×10^{-1}
	(4.1×10^{-4})	(1.85×10^{-3})	(1.8×10^{-4})	(5.6×10^{-4})	(6.6×10^{-4})	(5.1×10^{-3})	(3.7×10^{-3})	(5.5×10^{-3})	(1.46×10^{-3})
4	1.1×10^{-3}	2.9×10^{-3}	6.07×10^{-2}	8.8×10^{-3}	8.14×10^{-2}	2.6×10^{-1}	5.6×10^{-1}	3.47×10^{-1}	2.5×10^{-1}
	(4.1×10^{-4})	(2.05×10^{-3})	(3.4×10^{-2})	(4.3×10^{-3})	(2.3×10^{-3})	(2.7×10^{-2})	(1.56×10^{-3})	(3.6×10^{-3})	(6.28×10^{-3})
5	1.78×10^{-3}	3.76×10^{-3}	1.85	9.53×10^{-3}	9.5×10^{-2}	4.8×10^{-1}	5.68×10^{-2}	6.78×10^{-2}	3.24×10^{-2}
	(8.22×10^{-4})	(2.23×10^{-3})	(1.9×10^{-1})	(5.3×10^{-3})	(1.2×10^{-3})	(2.1×10^{-2})	(1.6×10^{-2})	(1.1×10^{-2})	(9.1×10^{-2})
6	1.65×10^{-3}	4.57×10^{-3}	4.5	1.0×10^{-2}	1.6×10^{-1}	6.1×10^{-1}	5.23×10^{-2}	3.01×10^{-2}	1.34×10^{-2}
	(3.9×10^{-4})	(4.18×10^{-3})	(8.8×10^{-1})	(4.5×10^{-3})	(6.5×10^{-3})	(6.2×10^{-2})	(6.7×10^{-3})	(8.7×10^{-3})	(7.34×10^{-2})
7	2.1×10^{-3}	5.33×10^{-3}	4.99	1.47×10^{-2}	1.57×10^{-1}	7.34×10^{-1}	1.09×10^{-2}	2.33×10^{-3}	6.77×10^{-4}
	(1×10^{-3})	(3.23×10^{-3})	(3.4×10^{-1})	(4.4×10^{-3})	(5.4×10^{-3})	(3.1×10^{-2})	(1.5×10^{-3})	(2.1×10^{-3})	(5.06×10^{-2})
8	1.56×10^{-3}	7.51×10^{-3}	5.45	1.35×10^{-2}	1.84×10^{-1}	9.5×10^{-1}	5.34×10^{-3}	6.56×10^{-4}	4.64×10^{-4}
	(6.4×10^{-4})	(4.87×10^{-3})	(1.4×10^{-1})	(6.8×10^{-3})	(9.1×10^{-3})	(2.9×10^{-2})	(3.5×10^{-4})	(3.67×10^{-4})	(2.34×10^{-3})

Table 4 Results of EDAGEA, HN and MOPSO to solve 3–8 objectives DTLZ4 test problems

N	*GD*-mean (*GD*-variance)			*S*-mean (*S*-variance)			*H*-mean (*H*-variance)		
	EDAGEA	HN	MOPSO	EDAGEA	HN	MOPSO	EDAGEA	HN	MOPSO
3	4.2×10^{-3}	3.8×10^{-2}	3.31×10^{-2}	6.34×10^{-3}	5.8×10^{-3}	8.8×10^{-3}	1.65×10^{-2}	8.6×10^{-2}	9.5×10^{-2}
	(6.1×10^{-3})	(2.05×10^{-2})	(0.24)	(2.9×10^{-3})	(6.3×10^{-3})	(2.6×10^{-3})	(2.87×10^{-2})	(8.7×10^{-3})	(7.7×10^{-3})
4	8.3×10^{-3}	2.1×10^{-2}	4.43×10^{-2}	2.23×10^{-3}	2.1×10^{-3}	3.14×10^{-3}	7.1×10^{-2}	3.4×10^{-2}	2.7×10^{-2}
	(8.1×10^{-3})	(2.05×10^{-2})	(4.54)	(1.18×10^{-3})	(2.3×10^{-3})	(3.4×10^{-3})	(2.3×10^{-2})	(6.2×10^{-2})	(8.3×10^{-2})
5	1.09×10^{-2}	5.32×10^{-2}	7.97	8.9×10^{-3}	9.5×10^{-3}	7.8×10^{-2}	1.35×10^{-2}	7.3×10^{-3}	9.33×10^{-3}
	(2.22×10^{-3})	(5.67×10^{-2})	(8.64)	(6.66×10^{-3})	(6.2×10^{-3})	(7.31×10^{-3})	(3.1×10^{-2})	(7.1×10^{-3})	(8.6×10^{-3})
6	8.05×10^{-3}	3.57×10^{-2}	1.09×10^{1}	1.27×10^{-2}	1.5×10^{-2}	2.15×10^{-2}	5.82×10^{-3}	9.8×10^{-3}	9.3×10^{-3}
	(7.9×10^{-4})	(3.18×10^{-2})	(1.57×10^{1})	(1.18×10^{-2})	(5.5×10^{-3})	$(6.3 \times 10^{-3}$	(6.7×10^{-2})	(9.1×10^{-3})	(9.4×10^{-3})
7	9.2×10^{-3}	1.9×10^{-2}	1.36×10^{1}	1.98×10^{-3}	9.57×10^{-3}	7.83×10^{-3}	1.84×10^{-3}	2.8×10^{-3}	2.3×10^{-3}
	(1.1×10^{-3})	(1.7×10^{-2})	(1.77×10^{1})	(2.3×10^{-3})	(9.4×10^{-3})	(8.23×10^{-3})	(6.9×10^{-2})	(3.1×10^{-3})	(2.4×10^{-3})
8	1.62×10^{-2}	8.1×10^{-2}	1.54×10^{1}	2.34×10^{-3}	6.84×10^{-3}	5.1×10^{-2}	1.0×10^{-3}	1.8×10^{-3}	1.3×10^{-3}
	(1.4×10^{-3})	(7.9×10^{-2})	(2.54×10^{1})	(3.5×10^{-2})	(7.1×10^{-3})	(5.93×10^{-3})	(7.8×10^{-3})	(2.1×10^{-3})	(1.4×10^{-3})

Table 5 Results of EDAGEA, HN and MOPSO to solve 3–8 objectives DTLZ7 test problems

N	*GD*-mean (*GD*-variance)			*S*-mean (*S*-variance)			*H*-mean (*H*-variance)		
	EDAGEA	HN	MOPSO	EDAGEA	HN	MOPSO	EDAGEA	HN	MOPSO
3	1.2×10^{-2}	4.8×10^{-2}	8.6×10^{-1}	3.01×10^{-2}	6.17×10^{-3}	7.3×10^{-3}	2.46×10^{-2}	2.9×10^{-3}	1.2×10^{-3}
	(4.1×10^{-3})	(3.05×10^{-2})	(1.21×10^{-2})	(3.6×10^{-3})	(6.3×10^{-3})	(5.32×10^{-3})	(2.78×10^{-3})	(3.7×10^{-3})	(2.7×10^{-3})
4	2.3×10^{-2}	3.1×10^{-2}	5.22	3.77×10^{-3}	2.08×10^{-3}	1.03×10^{-3}	2.92×10^{-2}	2.78×10^{-2}	2.16×10^{-2}
	(8.1×10^{-3})	(4.01×10^{-2})	(4.68)	(3.6×10^{-3})	(3.3×10^{-3})	(1.7×10^{-3})	(1.4×10^{-2})	(4.2×10^{-2})	(6.3×10^{-2})
5	1.09×10^{-2}	6.32×10^{-2}	8.91	8.73×10^{-3}	8.5×10^{-3}	8.9×10^{-2}	2.26×10^{-2}	6.45×10^{-3}	7.35×10^{-3}
	(2.3×10^{-3})	(4.67×10^{-2})	(9.64)	(9.31×10^{-3})	(7.2×10^{-3})	(6.66×10^{-2})	(2.2×10^{-2})	(7.51×10^{-3})	(7.54×10^{-3})
6	2.05×10^{-2}	5.6×10^{-2}	12.5	3.13×10^{-3}	3.3×10^{-2}	9.7×10^{-3}	6.71×10^{-2}	8.8×10^{-3}	8.97×10^{-3}
	(8.1×10^{-3})	(4.3×10^{-2})	(18.71)	(9.3×10^{-3})	(5.1×10^{-3})	(6.18×10^{-3})	(5.7×10^{-2})	(8.1×10^{-3})	(8.8×10^{-3})
7	3.2×10^{-2}	2.7×10^{-2}	15.87	7.2×10^{-4}	8.69×10^{-3}	7.16×10^{-3}	2.73×10^{-2}	2.17×10^{-3}	1.87×10^{-3}
	(8.15×10^{-3})	(2.6×10^{-2})	(15.09)	(2.1×10^{-3})	(3.4×10^{-3})	(4.23×10^{-3})	(5.9×10^{-2})	(2.7×10^{-3})	(6.4×10^{-3})
8	4.06×10^{-2}	8.93×10^{-2}	20.5	3.78×10^{-4}	5.5×10^{-3}	4.3×10^{-2}	1.9×10^{-2}	1.23×10^{-3}	1.1×10^{-3}
	(7.4×10^{-3})	(5.1×10^{-2})	(19.42)	(3.5×10^{-3})	(3.1×10^{-3})	(3.67×10^{-3})	(2.8×10^{-3})	(4.1×10^{-3})	(5.4×10^{-3})

Table 6 Average Rankings of EDAGEA, HN and MOPSO for DTLZ1, DTLZ2, DTLZ4 and DTLZ7 test problems by the Friedman test

Rankings	DTLZ1	DTLZ2	DTLZ4	DTLZ7
GD				
EDAGEA	1	1.17	1	1.17
HN	2.33	2.17	2.17	1.83
MOPSO	2.67	2.67	2.83	3
S				
EDAGEA	1.50	1	1.33	1.83
HN	1.67	2	1.83	2
MOPSO	2.83	3	2.83	2.17
H				
EDAGEA	2.67	2.83	1.67	3
HN	1.83	2	2.33	1.67
MOPSO	1.5	1.17	2	1.33

to the Pareto optimal front for the test problems DTLZ1, DTLZ2, DTLZ4, DTLZ7 with 3–8 objectives. MOPSO can close to the Pareto optimal front for these test problems with 3–4 objectives, whereas MOPSO cannot converge to the Pareto optimal front for these test problems with more than four objectives. From the GD-variance, the convergence and stability of EDAGEA are better than HN, and MOPSO is almost unstable. Because increasing in the number of non-dominant solutions leads to the selection pressure of MOPSO relatively small or even stagnant. MOPSO becomes a random research algorithm, and its performance is reduced sharply.

For the test problems of DTLZ1, DTLZ2, DTLZ4, DTLZ7 with 3–5 objectives, the S-mean of EDAGEA is slightly bigger than MOPSO; the S-mean of HN and MOPSO is close to each other. For the test problems of DTLZ1, DTLZ2, DTLZ4, DTLZ7 with 6–8 objectives, the S-mean of EDAGEA is smaller than that of HN and MOPSO. It Indicates that the test problem is more objectives, the results of EDAGEA are more uniform.

For the test problems of DTLZ1, DTLZ2, DTLZ4, DTLZ7 with 3–8 objectives, H-mean of EDAGEA is slightly bigger than that of HN and MOPSO for most of objectives. From H-variance, EDAGEA is relatively better than HN and MOPSO, but H-variance of EDAGEA is still less than H-mean. It shows that EDAGEA can maintain a good diversity and has relatively stable search ability during the evolutionary process.

According to the observations and discussions above, it can be concluded EDAGEA shows better performance in improving the convergence speed and maintaining the uniformity and diversity of the obtained solutions. The main reasons are as follows:

1. E-dominance mechanism enhances the selection pressure of the population;
2. The rule of Rank value constructed by E-dominance, entropy and free energy accelerates the convergence of the population;
3. Adaptive-grid technique maintains the uniformity and diversity of the population.

In order to further compare the total performance of the three algorithms on DTLZ1, DTLZ2, DTLZ4 and DTLZ7 test problems, we carry out the rankings of Friedman test on the experimental results following the suggestions in Garcia and Herrera (2008), Garc'ıa et al. (2009) and Wang et al. (2013). Table 6 presents the average rankings of the three algorithms on DTLZ1, DTLZ2, DTLZ4 and DTLZ7 test problems. We can sort these three algorithms by the average rankings into the following order: EDAGEA, HN and MOPSO. Therefore, EDAGEA obtains the best average ranking, and its total performance is better than that of the other two algorithms on DTLZ1, DTLZ2, DTLZ4 and DTLZ7 test problems.

5 Conclusions

PDEA is a classic MOEA, which encounters great challenges in solving MaOPs due to the ineffectiveness of the Pareto dominance strategy and the diversity maintenance mechanism. To address the issue, this paper proposes a new MOEA for MaOPs, called EDAGEA. The main idea is to use the E-dominance mechanism to enhance the search performance. Moreover, to keep the distribution and diversity of the population, the current non-dominant set is adaptively meshed according to the size of current population and the number of objectives.

To verify the performance, EDAGEA is compared with two excellent MOEAs including HN and MOPSO on DTLZ testing problems. Three performance indicators GD, H and S are used to evaluate the efficiency of the comparison algorithms. The experimental results indicate that the proposed EDAGEA is better than or equal to the two excellent competitors on the most of the test problems. Specifically, the proposed EDAGEA is better than HN and MOPSO in finding a uniformly distributed and well-approximated population. The experimental results demonstrate that using E-dominance in EDAGEA can increase the selection pressure and therefore accelerate the convergence speed, while incorporating the adaptive-grid strategy into EDAGEA can improve the distribution of the population for MaOPs.

In future work, we intend to further investigate the performance of EDAGEA for a wider range of problems, especially for the real-world practical problems with a high number of objectives and complicated Pareto front.

Acknowledgments This work was supported in part by the National Natural Science Foundation of China (Nos. 41561091, 41261093, 61364015 and 61462036), by the Natural Science Foundation of Jiangxi, China (Nos. 20151BAB217010 and 20151BAB201015).

Compliance with ethical standards

Conflict of interest The authors declare that there is no conflict of interests regarding the publication of this paper.

References

Adra S, Fleming P (2011) Diversity management in evolutionary many-objective optimization. IEEE Trans Evolut Comput 15(2):183–195

Chen BJ, Shu HZ, Coatrieux G, Chen G, Sun XM, Coatrieux JL (2014) Color image analysis by quaternion-type moments. J Math Imaging Vis 51(1):124–144

Chikumbo O, Goodman ED, Deb K (2012) Approximating a multi-dimensional Pareto front for a land use management problem: a modified MOEA with an epigenetic silencing metaphor. In: IEEE Congress on evolutionary computation (CEC), vol 22, no 10, pp 1–9

Coello CAC, Pulido GT, Lechuga MS (2004) Handing multiple objectives with particle swarm optimization. IEEE Trans Evol Comput 8(3):256–279

Corne DW, Knowles JD, Oates MJ, Jerram NR (2002) PESA-II: region-based selection in evolutionary multi-objective optimization. In: Proceedings of the genetic and evolutionary computation conference, pp 283–290

Deb K, Jain H (2012) Handling many-objective problems using an improved NSGA-II procedure. In: IEEE Congress on evolutionary computation(CEC), pp 1–8

Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multi-objective genetic algorithm: NSGA-II. IEEE Trans Evol Comput 6(2):182–197

Deb K, Thiele L, Laumanns M, Zitzler E (2002) Scalable multi-objective optimization test problems. In: Proceedings of IEEE Congress on evolutionary computation, pp 825–830

Deb K, Thiele L, Laumanns M, Zitzler E (2005) Scalable test problems for evolutionary multi-objective optimization. In: Theoretical advances and applications, Evolutionary multi-objective optimization, Springer, London, pp 105–145

Fleming PJ, Purshouse RC, Lygoe RJ (2005) Many-objective optimization: an engineering design perspective. In: Evolutionary multi criterion optimization of lecture notes in computer science, pp 14–32

Freire H, Oliveira PBDM, Pires EJS, Bessa M (2014) Corner based many-objective optimization. Nat Inspired Coop Strateg Optim 512:125–139

Garc'ıa S, Molina D, Lozano M, Herrera F (2009) A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behavior: a case study on the CEC'2005 special session on real parameter optimization. J Heuristics 15(6):617–644

Garcia S, Herrera F (2008) An extension on statistical comparisons of classifiers over multiple data sets for all pairwise comparisons. J Mach Learn Res 9:2677–2694

Gu B, Sheng VS, Tay KY, Romano W, Li S (2014) Incremental support vector learning for ordinal regression. IEEE Trans Neural Netw Learn Syst 26(7):1403–1416

Guo T, Kang LS (1999) A new evolutionary algorithm for function optimization. Wuhan Univ J Nat Sci 4(4):409–414

Herrero JG, Berlanga A, L'opez JMM (2009) Effective evolutionary algorithms for many specifications attainment: application to air traffic control tracking filters. IEEE Trans Evol Comput 13(1):151–168

Ishibuchi H, Tsukamoto N, Nojima Y (2008) Evolutionary many-objective optimization: a short review. In: IEEE Congress on evolutionary computation (CEC), pp 2419–2426

Kang Z, Kang LS, Zou XF et al (2007) A new evolutionary decision theory for many-objective optimization problems. In: Kang L, Liu Y, Zeng S (eds) Advances in computation and intelligence. Springer, Berlin, pp 1–11

Knowles JD, Corne DW (2000) Approximating the non-dominated front using the Pareto archived evolution strategy. Evol Comput 8(2):149–172

Li YX, Zou XF, Kang LS (2003) A new dynamical evolutionary algorithm from statistical mechanics. Chin J Comput Sci Technol 18(3):361–368

Li KS, Li YX, Kang LS, Wu ZJ (2007) A MOP evolutionary algorithm based on transportation theory. Chin J Comput 5(30):796–805

Li J, Chen XF, Li MQ, Li JW, Lee P, Lou WJ (2014) Secure deduplication with efficient and reliable convergent key management. IEEE Trans Parallel Distrib Syst 25(6):1615–1625

Li BD, Li JL, Tang K, Yao X (2015a) Many-objective evolutionary algorithms: a survey. ACM Comput Surv 48(1):1–35

Li J, Li XL, Yang B, Sun XM (2015b) Segmentation-based image copy-move forgery detection scheme. IEEE Trans Inf Forensics Secur 10(3):507–518

Narukawa K (2013) Effect of dominance balance in many-objective optimization. In: Evolutionary multi-criterion optimization of lecture notes in computer science, vol 7811, pp 276–290

Schott JR (1995) Fault tolerant design using single and multicriteria genetic algorithm optimization. Massachusetts Institute of Technology, Cambridge

Shen J, Tan HW, Wang J, Wang JW, Lee SY (2015) A novel routing protocol providing good transmission reliability in underwater sensor networks. J Internet Technol 16(1):171–178

Shi Z, Chen QW (2011) Multi-objective optimization algorithm based on quantum-behaved particle swarm and adaptive grid. Inf Control 40(2):214–220

Veldhuizen DAV, Lamont GB (2000) On measuring multi-objective evolutionary algorithm performance. In: Proceedings of IEEE Congress on evolutionary computation, vol 1, pp 204–211

Wang H, Sun H, Li C, Rahnamayan S, Pan JS (2013) Diversity enhanced particle swarm optimization with neighborhood search. Inf Sci 223(2):119–135

Wen XZ, Shao L, Xue Y, Fang W (2015) A rapid learning algorithm for vehicle classification. Inf Sci 295:395–406

Xia ZH, Wang XH, Sun XM, Wang BW (2014a) Steganalysis of least significant bit matching using multi-order differences. Secur Commun Netw 7(8):1283–1291

Xia ZH, Wang XH, Sun XM, Liu QS, Xiong NX (2014b) Steganalysis of LSB matching using differences between nonadjacent pixels. Multimedia Tools Appl 75(4):1–16

Xie SD, Wang YX (2014) Construction of tree network with limited delivery latency in homogeneous wireless sensor networks. Wirel Pers Commun 78(1):231–246

Yang SX, Li MQ, Liu XH, Zheng JH (2013) A grid-based evolutionary algorithm for many-objective optimization. IEEE Trans Evol Comput 17(5):721–736

Zhang QF, Li H (2007) MOEA/D: a multi-objective evolutionary algorithm based on decomposition. IEEE Trans Evol Comput 11(6):712–731

Zheng JH (2007) Multi-objective evolutionary algorithms and applications. Science Press, Beijing

Zheng YH, Byeungwoo J, Xu DH, Wu QMJ, Zhang H (2015) Image segmentation by generalized hierarchical fuzzy C-means algorithm. J Intell Fuzzy Syst 28(2):4024–4028

Zitzler E, Laumanns M, Thiele L (2001) SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization. In: Proceedings of the 4th international conference on evolutionary methods for design optimization and control with applications to industrial problems, Athens, Greece, 19–21 September 2001, pp 95–100

Zitzler E, Thiele L (1999) Multi-objective evolutionary algorithms: a comparative case study and the strength Pareto approach. IEEE Trans Evol Comput 3(4):257–271

Zou XF, Li YX, Kang LS (2002) Finding global minima with a new dynamical evolutionary algorithm. Wuhan Univ J Nat Sci 7(2):157–160