Soft Comput (2017) 21:2283–2295
DOI 10.1007/s00500-015-1940-x

METHODOLOGIES AND APPLICATION

Nadir point estimation for many-objective optimization problems based on emphasized critical regions

Handing Wang[1,2] · **Shan He**[3] · **Xin Yao**[3]

Published online: 17 November 2015

Abstract Nadir points play an important role in many-objective optimization problems, which describe the ranges of their Pareto fronts. Using nadir points as references, decision makers may obtain their preference information for many-objective optimization problems. As the number of objectives increases, nadir point estimation becomes a more difficult task. In this paper, we propose a novel nadir point estimation method based on emphasized critical regions for many-objective optimization problems. It maintains the non-dominated solutions near extreme points and critical regions after an individual number assignment to different critical regions. Furthermore, it eliminates similar individuals by a novel self-adaptive ε-clearing strategy. Our approach has been shown to perform better on many-objective optimization problems (between 10 objectives and 35 objectives) than two other state-of-the-art nadir point estimation approaches.

Keywords Multi-objective evolutionary algorithm · Nadir point · Critical region · Many-objective optimization problem · Decision making

1 Introduction

In the solution set of an multi-objective optimization problem (MOP), some points are very important, such as ideal points, knee points, and nadir points. It is easy to obtain an ideal point by optimizing every single objective separately. Around a knee point (Branke et al. 2004), a small improvement of any objective leads to a large degeneration of other objectives. Knee points are interesting to decision makers if they do not know their preferences. A nadir point is derived from the extreme points in a non-dominated solution set, which describes the range of the non-dominated solution set indirectly. Therefore, nadir points play an important role in MOPs, especially in many-objective optimization problems.

Many-objective optimization problems (MaOPs) are a type of MOPs with more than three objectives (Khare et al. 2003). One of the challenges in this research area is that the non-dominated solution set becomes too large to handle effectively as the dimension of the objective space increases. Recently, many new MOEAs have been designed for MaOPs. For example, NSGA-III (Deb and Jain 2014) uses a set of reference points to maintain diversity; Two_Arch2 (Wang et al. 2015) tackles convergence and diversity in separate archives; MOEA/D-DU (Yuan et al. 2015b) tries to introduce NSGA-III's diversity maintenance scheme into MOEA/D; and θ-dominance (Yuan et al. 2015a) introduces a new dominance relation that can provide more selection pressure for MaOPs, inspired by penalty-based boundary intersection (PBI).

Usually, decision makers do not need the entire solution set. They only need the solutions in their interested regions. In

Communicated by V. Loia.

✉ Handing Wang
wanghanding@163.com

Shan He
s.he@cs.bham.ac.uk

Xin Yao
x.yao@cs.bham.ac.uk

[1] Key Lab of Intelligent Perception and Image Understanding of Ministry of Education, International Research Center of Intelligent Perception and Computation, Xidian University, Xi'an 710071, China

[2] Department of Computer Science, University of Surrey, Guildford GU2 7XH, UK

[3] CERCIA, School of Computer Science, The University of Birmingham, Birmingham B15 2TT, UK

such situations, preference-based many-objective optimization (Thiele et al. 2009) can be used to obtain those solutions. However, it is not easy to obtain a comprehensive view of the preferences of decision makers (Fan and Liu 2010; Hsu and Wang 2011; Dai et al. 2013; He et al. 2015) without knowing the range of the non-dominated solution set before running preference-based multi-objective evolutionary algorithms (MOEAs). Nadir points can provide more accurate information about preferences for decision makers (Branke et al. 2008; Deb and Kumar 2007). Nadir points can also be used for normalizing objectives and portfolio problems (Amiri et al. 2010).

Approaches to obtain nadir points can be classified into two categories, namely non-evolutionary and evolutionary approaches:

Non-evolutionary approaches The pay-off table (Benayoun et al. 1971) is the earliest nadir point estimation method. All the objective values of the optimal solutions for every single objective are listed on the pay-off table. Then, the worst objectives from the table are obtained to construct a nadir point. The pay-off table is easy to construct but does not guarantee to obtain the true nadir points (Szczepanski and Wierzbicki 2003; Klamroth and Miettinen 2008; Deb et al. 2006). Several improvements to the pay-off table were proposed (Dessouky et al. 1986; Korhonen et al. 1997). For example, the method proposed in (Ehrgott and Tenfelde-Podehl 2003) decomposes an m-objective optimization problem into C_m^2 2-objective optimization problems. However, it is impractical for MaOPs because of its high computational cost. There are some other nadir point estimation methods based on reference directions (Korhonen et al. 1997) and points (Metev and Vassilev 2003) in nadir regions. However, these methods still cannot obtain the true nadir points.

Evolutionary approaches According to (Deb and Miettinen 2009), the existing nadir point estimation approaches based on evolutionary approaches can be classified into three types—‘Surface-to-Nadir’, ‘Edge-to-Nadir’, and ‘Extreme-point-to-Nadir’. Popular MOEAs such as NSGA-II (Deb et al. 2002a), MOEA/D-IR (Li et al. 2015), MOEA-DLA (Chen et al. 2015), MOEA/DVA (Ma et al. 2015) and MOEA/D-ACO (Ke et al. 2013) can obtain extreme points from the entire Pareto front (PF) to construct a nadir point. All the MOEAs mentioned above follow the ‘Surface-to-Nadir’ approach. ‘Surface-to-Nadir’ approaches cannot be applied to MaOPs because of their unsatisfactory performance on MaOPs (Praditwong and Yao 2007). One of the ‘Edge-to-Nadir’ approaches (Szczepanski and Wierzbicki 2003) decomposes the original problem into C_m^2 2-objective subproblems to build a nadir point from the obtained solutions of those subproblems, which leads to C_m^2 edges. As the method in (Szczepanski and Wierzbicki 2003) only considers two objectives at a time, it does not guarantee to obtain the true nadir points in all cases. The ‘Extreme-point-to-Nadir’ approach focuses on extreme points rather than the entire solution set or edges to build nadir points, such as *Worst Crowded* NSGA-II (Deb et al. 2006) and *Extremized Crowded* NSGA-II (Deb et al. 2006). Since nadir point estimation can be seen as a special preference problem of MOPs (Thiele et al. 2009), the task of nadir point estimation was treated as solving a preference problem in nadir regions in (Amiri et al. 2010). Also, NSGA-III views nadir point estimation as a preference problem by employing m reference points near m extreme points (Deb and Jain 2014). However, these reference points are based on some strong assumptions, which may not hold for different PF structures.

‘Extreme-point-to-Nadir’ approaches are shown to be the most effective methods to obtain nadir points so far (Deb et al. 2010). Only extreme points are needed for the extraction of a nadir point. That is the reason why the existing approaches focus on extreme points. Based on the frame of ‘Extreme-point-to-Nadir’ approaches, some local search strategies (Deb et al. 2010, 2009; Bechikh et al. 2010; Miettinen et al. 2010) are applied to further improve their performance, but extra computational expenses are added inevitably. For example, a reference-point-based bi-level local search approach with scalarizing functions is adopted in (Deb et al. 2010), and a bi-level optimization method is needed to suit the characteristics of the specific problem. Therefore, a hybrid approach can improve performance for nadir point estimation on specific problems rather than general problems.

However, focusing on extreme points too much causes poor diversity in the population during search. The population is easily trapped into local optima. Therefore, poor extreme points may be obtained in the population with poor diversity. The performance of the existing approaches such as *Worst Crowded* and *Extremized Crowded* NSGA-II (Deb et al. 2006) on MaOPs has been shown less than satisfactory. In order to improve their performance, we propose a novel ‘Extreme-point-to-Nadir’ based estimation method (non-hybrid approach, the local search strategies in (Deb et al. 2010, 2009; Bechikh et al. 2010; Miettinen et al. 2010) can be employed to improve its performance), named emphasized critical regions based nadir point estimation (ECRNPE). Its main idea is to employ the emphasized critical regions strategy (ECR) to add more diversity in the population to obtain extreme points more effectively. The proposed method can be easily incorporated into a wide range of MOEAs to obtain nadir points by replacing their diversity maintenance methods with ECR. The detailed contributions of our paper are summarized.

Emphasized critical regions strategy In nadir point estimation approaches, only focusing on extreme points during the search does not help the convergence towards extreme points. Meanwhile, it is unnecessary either to keep the entire PF for nadir point estimation, which increases the computational cost significantly. In order to find the trade-off between these two extreme cases, ECR is employed in this paper. Critical regions are the regions of the best objective values, which approximate the outline of a PF. Nadir regions are in critical regions. A PF can be simply expressed by critical regions. Therefore, the individuals near extreme points as well as critical regions are emphasized by ECR.

Individual number assignment in critical regions ECR selects the individuals in different critical regions independently. Thus, it is very important to assign a number of individuals in different regions for diversity maintenance and balanced computational cost. The number of individuals in each critical region is assigned self-adaptively in ECR.

Self-adaptive ε-clearing strategy The ε-clearing strategy (Ben Said et al. 2010) has been widely used for diversity maintenance. For nadir point estimation, the approach in (Amiri et al. 2010) adopts the ε-clearing strategy, where a pre-defined parameter ε is required. However, ε is sensitive to the distribution of the population. Therefore, a fixed ε might not help accurate nadir point estimation. In this paper, a self-adaptive ε-clearing strategy is proposed, which adjusts the parameter ε according to the distribution of the current population.

The rest of this paper is organized as follows: Problem definitions are introduced in Sect. 2. Section 3 gives the details of ECRNPE. Section 4 describes the simulation experiment to analyze the behavior of ECRNPE. The Sect. 5 presents the concluding remarks.

2 Problem definitions

2.1 Multi-objective and many-objective optimization

An MOP (Miettinen 1999) with m objectives can be described by

$$\begin{aligned}\text{Minimize } \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots f_m(\mathbf{x}))^T \\ \text{Subject to } \mathbf{x} \in X,\end{aligned} \tag{1}$$

where $X \subset \mathbb{R}^n$ is the feasible space, $\mathbf{x} = \{x_1, x_2, \ldots, x_n\}^T$ is the decision variable vector, and $\mathbf{F}(\mathbf{x})$ is the objective vector. The final result of an MOP is a set of non-dominated solutions (or Pareto optima). The set of all Pareto optimal objective values is called the Pareto front (PF). An MaOP (Khare et al. 2003; Hughes 2005) is an MOP with more than 3 objectives. For MaOPs, it is hard to select good solutions by the Pareto dominance relation, because most solutions cannot be dominated by others (Ishibuchi et al. 2008). The PFs of MaOPs are hard to be described by a limited number of solutions, which makes MaOPs very challenging (Praditwong and Yao 2007; Purshouse and Fleming 2003).

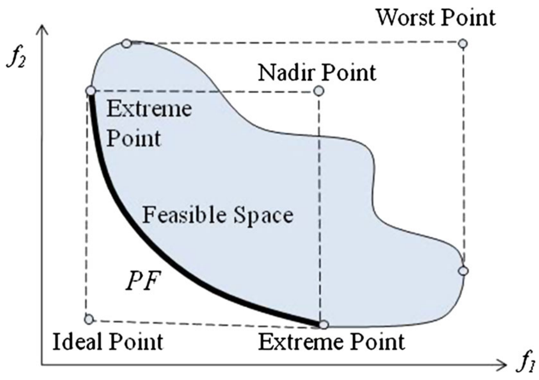

Fig. 1 Illustration of important points for MOPs

2.2 Important points for MOPs

As illustrated in Fig. 1, there exist several important points which can be used to characterize the solution set of an MOP.

An ideal point is constructed by all the optimal objective values as shown in Eq. (2), where m is the number of objectives. An ideal point describes the ideal situation of an MOP. Ideal points cannot be achieved as solutions, because the objectives of MOPs are usually in conflict with each other (Deb et al. 2010).

$$z^{\text{ideal}} = (\min(f_1), \ldots, \min(f_i), \ldots, \min(f_m)), 1 \leq i \leq m \tag{2}$$

A worst point is constructed by all the worst objective values in the feasible area as shown in Eq. (3). It describes the worst situation to an MOP (Deb et al. 2010).

$$z^{\text{worst}} = (\max(f_1), \ldots, \max(f_i), \ldots, \max(f_m)), 1 \leq i \leq m \tag{3}$$

A nadir point is constructed by all the worst objective values on the PF as shown in Eq. (4). It describes the range of the PF (Deb et al. 2010).

$$\begin{aligned}z^{\text{nadir}} = (\max(f_1), \ldots, \max(f_i), \ldots, \max(f_m)), \\ f_i \in PF, 1 \leq i \leq m\end{aligned} \tag{4}$$

There are m extreme points for an MOP with m objectives. The definition of the ith extreme point is given in Eq. (5). Obtaining these m extreme points is equivalent to obtaining a nadir point. The extraction method of a nadir point from extreme points is given in Sect. 3.3.4.

$$\begin{aligned}z_i^{\text{extreme}} = (f_1(\mathbf{x}), \ldots, f_i(\mathbf{x}), \ldots f_m(\mathbf{x})), \\ \mathbf{x} = \arg\min(f_i), \\ f_i \in PF, 1 \leq i \leq m\end{aligned} \tag{5}$$

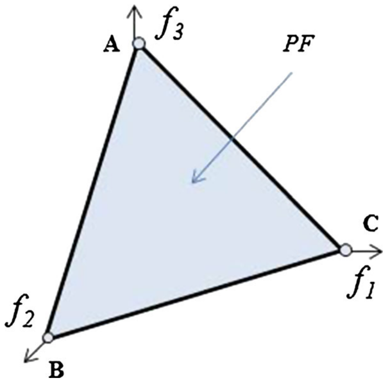

Fig. 2 Illustration of critical regions

3 Nadir point estimation based on emphasized critical regions

3.1 Critical region

The critical region of an MOP is defined as the region on the PF with the best value of one objective. Theoretically, there are m critical regions for an MOP with m objectives. The critical regions can be seen as the boundaries of the solution set in the objective space. Taking the 3-objective optimization problem in Fig. 2 as an example, points A, B, and C are extreme points, and edges BC, CA, and AB representing the best objective values of f_3, f_2, f_1, respectively, are from critical regions. The dimension of the critical region depends on the structure of the PF.

3.2 Main idea of ECRNPE

ECR is a diversity maintenance method for nadir point estimation. As shown in Fig. 3, the selection of conventional MOEAs aims at convergence and diversity. By simply replacing their diversity maintenance methods with ECR, any conventional MOEAs can be used to estimate nadir points. By adding nadir point extraction at the end of ECR, it becomes ECRNPE, a method to obtain nadir points. In other words, ECRNPE is used (instead of common diversity maintenance methods) in every generation of MOEAs to obtain nadir points.

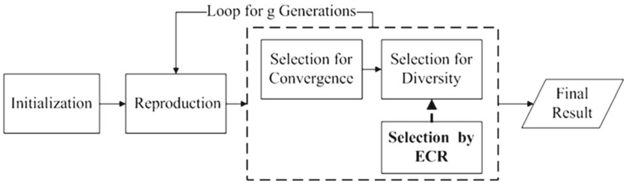

Fig. 3 Embed ECR in conventional MOEAs

As a nadir point is extracted from extreme points, searching extreme points is essential to nadir point estimation approaches. Extreme points are a small part of the entire PF. Only focusing on extreme points during search does not help the convergence towards extreme points. Therefore, the main idea of ECR is to maintain diversity in both the areas near extreme points and critical regions. In ECR, the individuals near extreme points and critical regions are ranked higher for selection. As shown in Fig. 4, ECR has three main parts, i.e., the individual number assignment, the critical region emphasizing strategy, and the self-adaptive ε-clearing strategy. Every objective has a critical region, in which the individuals have the best value of this objective. ECR first assigns the number of the individuals which should be maintained in those different critical regions. Then, it is repeated for m times to select a fixed number of individuals by the critical region emphasizing strategy and the ε-clearing strategy. The outputs from ECR will be the selected individuals, which are then used for the final step of ECRNPE (nadir point extraction). The details of ECRNPE are explained in the following sections.

3.3 Proposed approach

3.3.1 Individual number assignment to critical regions

According to the definition in Sect. 3.1, an MOP with m objectives and N solutions in its population has m extreme points and m critical regions. ECR selects the individuals in different critical regions separately. It is necessary to assign the same number of individuals in m critical regions to balance the computational cost of the later individual selection. The number of the maintained individuals in each region is n_1 ($\lfloor N/m \rfloor$). Because of the round-off operation in n_1, the remaining n_2 ($N - mn_1$) individuals are randomly selected from different regions. As n_2 is very small in comparison to the population size N, those n_2 individuals have very little influence on the uniformity of the assignment. The uniformity of different critical regions will be shown through the PF results in Sect. 4.3.2.

3.3.2 Emphasized critical regions strategy

The emphasized critical regions strategy is the first step of individual selection in the critical region of one objective, which essentially ranks the unselected individuals in the population. It first assigns the highest rank to extreme points. Then, it assigns ranks to the remaining individuals according to their distances to critical regions. Taking a minimization objective f_i as an example, the individual with the worst f_i which can be seen as an extreme point in the current population is ranked as “1”; the remaining individuals are ranked by sorting the values of objective f_i in an ascending order.

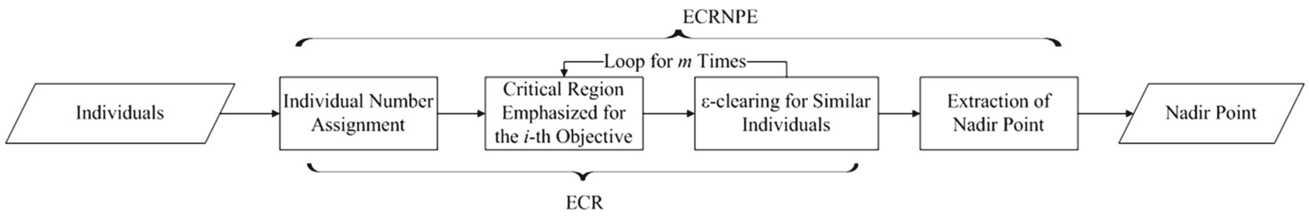
ECRNPE
Loop for m Times
Individuals
Individual Number Assignment
Critical Region Emphasized for the i-th Objective
ε-clearing for Similar Individuals
Extraction of Nadir Point
Nadir Point
ECR

Fig. 4 Flow-chart of ECRNPE

By applying this emphasized critical regions strategy, the individuals close to extreme points and critical regions are emphasized with higher ranks.

As shown in Fig. 4, the emphasized critical regions strategy and the ε-clearing strategy work together in m iterations to select the individuals close to the critical region of objective f_i ($i = 1, \dots, m$). After each iteration, n_1 individuals are selected. They will not be considered in the later iterations for other critical regions. An individual with better values on both objectives f_i and f_j will be selected for either objective. Therefore, the order of the individual selection of different regions has no influence on the final selected individuals. In this paper, ECR uses the sequential order from f_1 to f_m.

3.3.3 Self-adaptive ε-clearing strategy

In order to avoid selecting similar individuals, ECR applies the ε-clearing strategy (Ben Said et al. 2010) to eliminate similar individuals from the sorted population obtained from the emphasized critical regions strategy. The pseudo-code of the selection is shown in Algorithm 1:

Algorithm 1 Pseudo code of the ε-clearing strategy (Ben Said et al. 2010) for the selection in the critical region of f_i.

```
Input: $POP$-population of non-dominated individuals ranked by
  the emphasized critical regions strategy on objective $f_i$ as in Sect.
  3.3.2, $n_1$-the number of individuals in this critical region, $N$-
  population size, $\varepsilon$-neighborhood parameter, $S$-boolean list of $POP$
  to indicate whether individuals are selected, $counter$-the counter of
  selected individuals in critical region of $f_i$.
$counter = 0$, % initialize the counter,
$S[1] = 1$, $counter = counter + 1$,% extreme point $POP[1]$ for
  objective $f_i$ is selected firstly,
For $k = 2 : N$
  If $S[k] == 0$ and $POP[k]$ is not in the $\varepsilon$-neighborhoods of any
  selected individual,
      $S[k] = 1$, $counter = counter + 1$, % it is deleted from
  emphasized critical regions for other objectives,
      If $counter == n_1$, % $n_1$ individuals for critical region of $f_i$
  have been selected,
        Break,
      End
  End
End
Output: the updated list $S$.
```

The ε-clearing strategy is an effective technique to maintain diversity (Amiri et al. 2010; Ben Said et al. 2010). In previous studies, parameter ε was fixed. However, it is very difficult to choose an optimal parameter ε in advance. Furthermore, a fixed ε may not be suitable for different distributions of individuals in a population in the process of optimization. Therefore, we propose a self-adaptive ε-clearing strategy. Parameter ε is self-adaptive according to the distribution in the current population and calculated by Eq. (6), in which n_1 is the number of the maintained individuals in one critical region, f_i is the ith objective value in the current population.

$$\varepsilon = \sqrt{\sum\nolimits_{i=1}^{m} (\max(f_i) - \min(f_i))^2 / n_1} \qquad (6)$$

3.3.4 Extraction of the nadir point

As the individuals obtained are approximate extreme points, the last step of ECR is to extract the nadir point from the population. The details are described in Algorithm 2.

Algorithm 2 Pseudo code for extracting the nadir point.

```
Input: $POP$-population of non-dominated individuals, $m$-the num-
  ber of objectives, $N$-the number of individuals in $POP$.
For $i$=1:$m$
  Find the worst individual on objective $f_i$.
  $Nadir_i$=max($POP[1:N].f_i$)
End
$Nadir = (Nadir_1, \dots Nadir_i, \dots Nadir_m)$
Output:$Nadir$.
```

3.3.5 Discussion

ECR appears to divide a population into m sub-populations as in an island model (Branke et al. 2000). Island model is well known for its separate sub-populations, which independently optimize and occasionally interact each other. In fact, ECR is not the case, only the selected individuals are divided into m parts for those m different selections to different regions. Neither the population nor the final selected population is divided. As we know, the interaction between different sub-populations is the main characteristic of an island model.

Table 1 True ideal and nadir points of the DTLZ and WFG problems

	DTLZ1-4	WFG1-9
Ideal point	(0,0,…,0)	(0,0,…,0)
Nadir point	(1,1,…,1)	(2,4,…,2m)

Different from the island model, ECR does not apply such interactions as crossover among multiple populations.

4 Simulation experiments

As ECR can be embedded in any MOEA, we embed ECR in both Pareto-based MOEAs [NSGA-II (Deb et al. 2002a) and Two-Archive Algorithm (TwoArch) (Praditwong and Yao 2006)] and non-Pareto-based MOEAs (IBEA Zitzler and Künzli 2004) in our experiments to evaluate ECR's effectiveness in finding nadir points. Through the experiments below, the performance of ECR is analyzed from different aspects.

4.1 Metric, test problems, and parameters

As the nadir point is not a solution to an MOP, most metrics cannot be used in our experiments, such as *Hypervolume* (Zitzler and Thiele 1999; Jiang et al. 2014). *Error metric E* (Deb et al. 2002b) has been used for evaluating the accuracy of obtaining nadir points, which describes the distance from the estimated nadir point to the true nadir point. The formula is shown in Eq. (7), where z^{ideal} and z^{nadir} stand for the true ideal point and nadir point, and z is the estimated nadir point. It is clear that E is a normalized distance. The scales of different objectives have no influence on E.

$$E = \sqrt{\sum_{i=1}^{m} \left((z_i^{\text{nadir}} - z_i)/(z_i^{\text{nadir}} - z_i^{\text{ideal}})\right)^2} \tag{7}$$

Recently, many benchmark MOPs were proposed (Ishibuchi et al. 2014), such as the ZDT problems (Zitzler et al. 2000), the DTLZ problems (Deb et al. 2002b), the WFG problems (Huband et al. 2006), and some other problems (Szczepanski and Wierzbicki 2003; Klamroth and Miettinen 2008). The DTLZ and WFG problems are selected as the test problems in this paper because they can be extended to MaOPs and their nadir points are known as listed in Table 1. We use the same parameter setting as that used in (Deb et al. 2002a) [SBX crossover ($\eta = 15$) and polynomial mutation ($\eta = 15$)]. All the experiments were repeated independently for 29 times. The population size is set to 100 for the problems with 2–10 objectives, and 200 for the problems with more than 10 objectives.

4.2 Experiments on self-adaptive ε-clearing strategy

The self-adaptive ε-clearing strategy is one of our main contributions in this paper. In the following experiment, we compare *ECR* NSGA-IIs with and without the self-adaptive ε-clearing strategy on DTLZ2 with 10 objectives. Both algorithms repeated for 29 independent times with a population of 100 individuals and for 1000 generations. The metric E is employed to evaluate the performance of these two algorithms. The boxplot of E of the final result is shown in Fig. 5. According to Fig. 5, the *ECR* NSGA-II without the self-adaptive ε-clearing strategy cannot obtain the true nadir point within 1000 generations, while the *ECR* NSGA-II with the self-adaptive ε-clearing strategy can. As the ε-clearing strategy eliminates similar individuals, more individuals in critical regions can be maintained for better diversity. Thus, the self-adaptive ε-clearing strategy improves the convergence to the true nadir point.

4.3 Experimental study on computational time and scalability

4.3.1 Comparative experiments

ECR NSGA-II, *Extremized Crowded* NSGA-II (Deb et al. 2006), NSGA-II, *ECR* TwoArch, TwoArch, *ECR* IBEA, and IBEA are included in our comparisons in this section. We test the DTLZ problems with 2–20 objectives. Because of the different difficulties of different problems, the true nadir points of the DTLZ1 with 2–15 objectives, the DTLZ2 with 2–20 objectives, the DTLZ3 with 2–10 objectives, and the DTLZ4

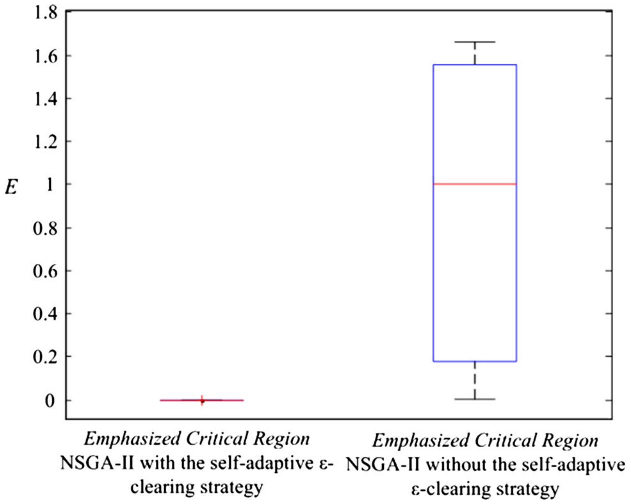

Fig. 5 Boxplot of E for *ECR* NSGA-IIs with and without the self-adaptive ε-clearing strategy on DTLZ2 with 10 objectives. Obviously, the *ECR* NSGA-II with the self-adaptive ε-clearing strategy found the true nadir point while the *ECR* NSGA-II without the self-adaptive ε-clearing strategy did not find the true nadir point

with 2–20 objectives can be obtained within a limited computational cost. Normally, these approaches run for a fixed number of generations. In order to fairly evaluate the convergence speed to nadir points, the stopping criterion is changed to $E < 0.01$, which indicates very close approximation to the true nadir points. Under this criterion, an algorithm with the smallest number of function evaluations has the highest speed. The metric E is calculated in every generation. If $E < 0.01$, the algorithm stops. It is worth noting that the true nadir point is only used for calculating E. The true nadir point has no influence on the behaviors of the compared algorithms. It is only used for performance evaluation. The parameter settings were given in Sect. 4.1. The numbers of function evaluations of the seven algorithms under the same stopping criterion ($E < 0.01$) are listed in Fig. 6.

It is clear that the *Original* NSGA-II had difficulties in converging to the true nadir point when the number of objectives increases. With the increasing number of objectives, the required function evaluations of the *Original* NSGA-II increase more significantly in comparison with the other two NSGA-IIs. *Extremized Crowded* NSGA-II performs better than *ECR* NSGA-II on problems with a smaller number of objectives. With more than 7 objectives for DTLZ1, 6 objectives for DTLZ2, and 10 objectives for DTLZ4, *ECR* NSGA-II uses fewer function evaluations than *Extremized Crowded* NSGA-II. For DTLZ3 with 2 to 10 objectives, *Extremized Crowded* NSGA-II uses fewer function evaluations than *ECR* NSGA-II. As DTLZ3 is hard to converge to its true PF, we speculate that the diversity in the critical regions cannot improve convergence.

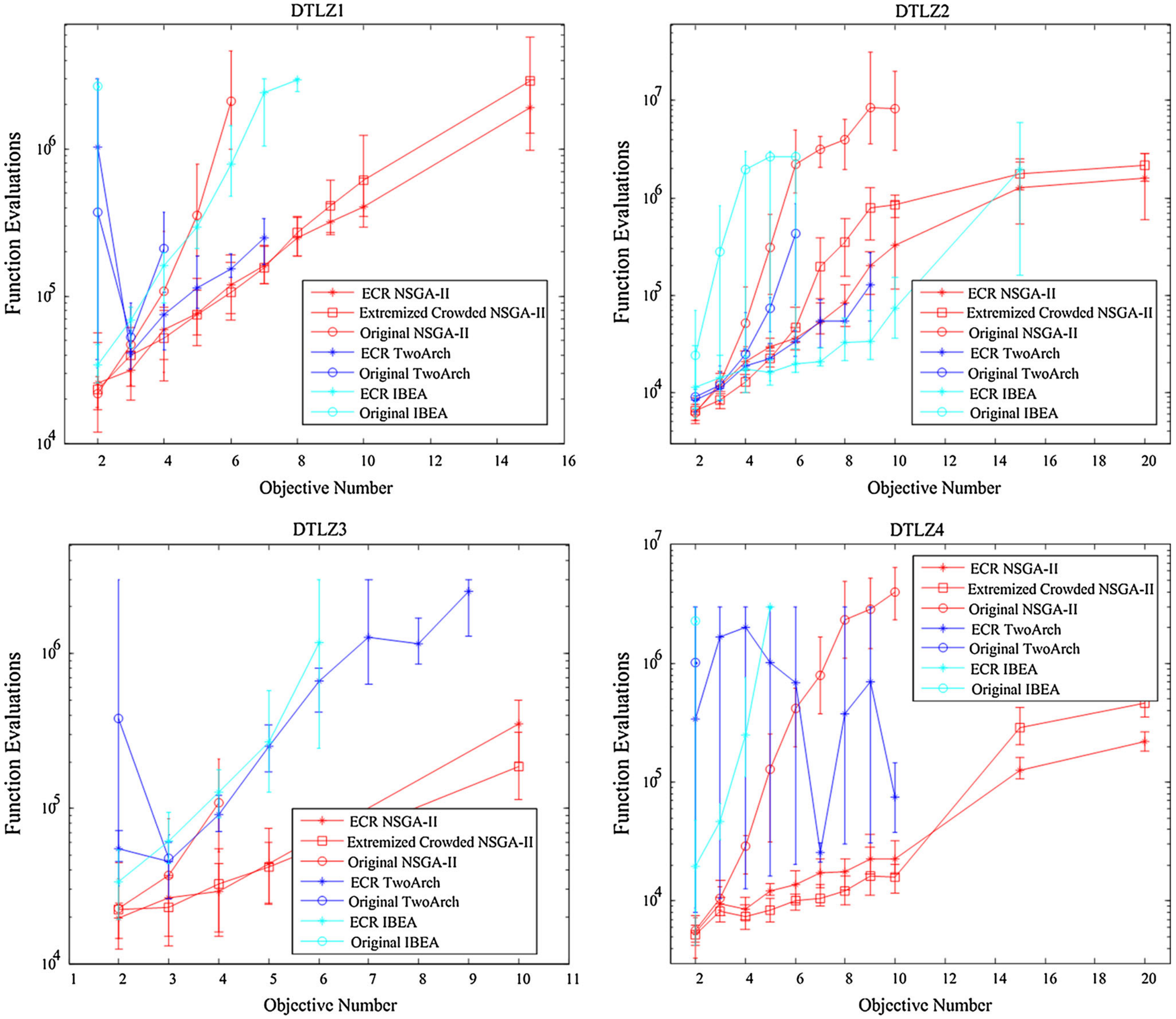

Fig. 6 Function evaluations with a variable number of objectives on the DTLZ problems within limited computational time

For TwoArch, we can find that ECR improves the nadir point estimation performance of the *Original* TwoArch on most of the DTLZ problems. When the number of objectives increases, the *Original* TwoArch does not obtain the true nadir point, while *ECR* TwoArch performs well on these problems. Since the *Original* TwoArch is specifically designed for MaOPs, both algorithms perform unsatisfacorily on the 2-objective problems. For DTLZ4, we find that the required number of function evaluations of *ECR* TwoArch does not increase with the number of objectives. Especially, the number of function evaluations for the DTLZ4 with 7 objectives is the smallest. DTLZ4 is the easiest problem to obtain the solutions near its extreme points among all of the DTLZ problems. Furthermore, there are only three kinds of solutions (the non-dominated solutions with domination, the non-dominated solutions without domination, and the dominated solutions) in the *Original* TwoArch, which increases the convergence speed on DTLZ4. However, this characteristic of the *Original* TwoArch leads to varied performance on the same problem in different runs. That is the reason why the curve of DTLZ4 becomes such irregular.

For IBEA, we find that ECR can improve its nadir point estimation ability significantly. For DTLZ1, DTLZ3, and DTLZ4, the *Original* IBEA did not obtain any points close to extreme points. Even for DTLZ3 with two objectives, the *Original* IBEA does not obtain the nadir point within the limited computational time. After adding ECR, IBEA improves the performance of nadir point estimation on MaOPs.

Comparing the above results from different algorithms, we find that the results of *ECR* IBEA are not better than that of *ECR* NSGA-II. Theoretically, IBEA is an excellent MOEA on MaOPs. We expect that it should perform better than NSGA-II, especially in terms of convergence. However, because IBEA has almost no diversity maintenance mechanisms, extreme points are harder to be maintained in IBEA than in NSGA-II, which might explain the worse results from *ECR* IBEA (its maximum number of objectives is much smaller than 20). Similarly, comparing the results of IBEA to those of TwoArch, we find that the results of *ECR* IBEA are not better than those of *ECR* TwoArch, either. However, it is worth noting that the performance of estimating nadir

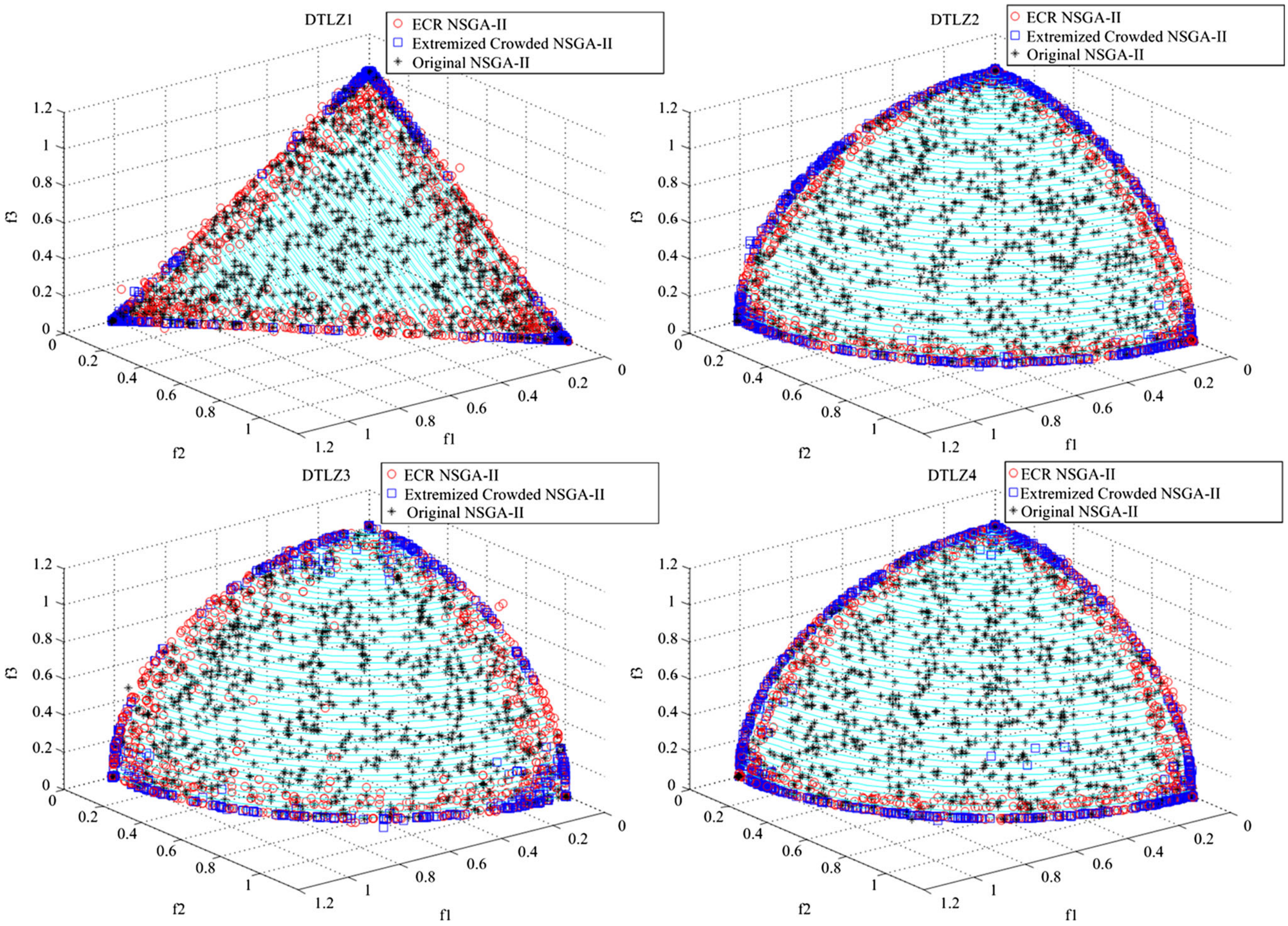

Fig. 7 Final populations of all the runs of *ECR* NSGA-II, *Extremized Crowded* NSGA-II, and *Original* NSGA-II on the 3-objective DTLZ problems

points is not the same thing as the performance for solving the MOP itself.

4.3.2 Discussion

In order to analyze the mechanism of ECR, we show the final populations of *ECR* NSGA-II, *Extremized Crowded* NSGA-II, and *Original* NSGA-II on the DTLZ problems with 3 objectives in Fig. 7. According to Fig. 7, the final population of the *Original* NSGA-II is distributed on the entire PF, that from *Extremized Crowded* NSGA-II is distributed in the regions of extreme points, and that from *ECR* NSGA-II is distributed in the critical regions most uniformly as expected in Sect. 3.3.1.

The results show different behaviors caused by different diversity maintenance strategies on different problems. The solutions of *Original* NSGA-II are distributed on the entire PF. *Original* NSGA-II are not suitable for the nadir point estimation of MaOPs. *Extremized Crowded* NSGA-II improves the above disadvantages by concentrating on extreme points. In order to maintain the diversity of the current population, *Extremized Crowded* NSGA-II maintains some individuals in critical regions. It can solve some MOPs with 15–20 objectives. However, its diversity is unsatisfactory. The individuals are distributed in some areas non-uniformly. To overcome this weakness, *ECR* NSGA-II applies the self-adaptive ε-clearing strategy to delete similar individuals for better diversity. Larger diversity in critical regions helps to improve convergence to the true nadir point. *ECR* NSGA-II performs better than *Extremized Crowded* NSGA-II on most MaOPs.

However, the individuals in the critical regions of DTLZ3 cannot be maintained as well as we expected, which was shown in Fig. 7. The critical regions of DTLZ3 sre hard to obtain. In ECR, we only keep the individuals of critical regions, which is of insufficient diversity for DTLZ3. Because we cannot obtain the expected individuals, the behavior of *ECR* NSGA-II on DTLZ3 was less satisfactory.

4.3.3 Scalability and extension

We find that *ECR* NSGA-II can solve DTLZ2 with 20 objectives easily. DTLZ2 with 25–35 objectives are chosen to test the scalability of ECR. In our experiments, the stopping criterion is set as $E < 0.01$. All the experiments are repeated with 29 independent runs. The number of objectives increases until ECR is not able to find satisfactory nadir points. The results in Table 2 show that ECR can find the nadir points for MOPs with 25–35 objectives within a reasonable number of function evaluations.

Table 2 Number of function evaluations of *ECR* NSGA-II (the stopping criterion is $E < 0.01$) on DTLZ2

Obj #	Best	Median	Worst
25	1440000	1965600	4258800
30	2650200	5426400	10290000
35	4320000	8736000	21110400

Table 3 Number of function evaluations of *ECR* NSGA-II (the stopping criterion is $E < 0.01$) on DTLZ1 with α from 0 to 20

α	Best	Median	Worst
0	71100	84900	108600
5	77400	115200	121800
10	76800	100200	178800
15	74700	87900	111600
20	54900	76800	109200

DTLZ2 has no local optima, which might be the reason why the scalability of proposed method can be up to 35 objectives. Fairly, we take another group of experiments to analyze the sensitivity from local optima by a series of modified DTLZ1 problems with 5 objectives. $g(\mathbf{x}_M) = 100[|\mathbf{x}_M| + \sum_{x_i \in \mathbf{x}_M} ((x_i - 0.5)^2 - \cos(\alpha\pi(x_i - 0.5)))]$ in DTLZ1 controls the local optima, and α controls the frequency of local optima. In the following experiment, we run *ECR* NSGA-II (the stopping criterion is set as $E < 0.01$) on DTLZ1 with α from 0 to 20 (no local optima to large amount of local optima). From the result in Table 3, we find that the amount of local optima weakly affects the performance of our method.

The existing test problems all have special PFs, on which a single Pareto optimal solution has the best values of all objectives. NSGA-III (Deb and Jain 2014) is based on such a strong assumption to assign reference points, which is the reason why it can perform well on nadir point estimation. However, the PFs of MOPs may have very different structures in practice. In order to thoroughly evaluate the performance of our method, we use a modified DTLZ2 by changing its PF structure as shown in Fig. 8. The modified DTLZ2 is given in Eq. (8). Thus, only one single Pareto optimal solution has the best objective value of a single objective on the PF of the new DTLZ2. Its true nadir point is (1,1,…,1).

$$
\begin{aligned}
&f_1(\mathbf{x}) = (1 + g(\mathbf{x}_M))(1 - \cos(x_1\pi/2)\cdots\cos(x_{m-1}\pi/2))\\
&f_2(\mathbf{x}) = (1 + g(\mathbf{x}_M))(1 - \cos(x_1\pi/2)\cdots\sin(x_{m-1}\pi/2))\\
&\vdots\\
&f_m(\mathbf{x}) = (1 + g(\mathbf{x}_M))(1 - \sin(x_1\pi/2))\\
&\mathbf{x}_M = \{x_i | i = m, \ldots n\}, n > m\\
&0 \le x_i \le 1 (i = 1, 2, \ldots n)\\
&\text{where} \;\; g(\mathbf{x}_M) = \textstyle\sum_{x_i \in \mathbf{x}_M} (x_i - 0.5)^2
\end{aligned}
\tag{8}
$$

ECR NSGA-II, *Extremized Crowded* NSGA-II and NSGA-III are selected for the comparison. The constraint $g(\mathbf{x})$ of the

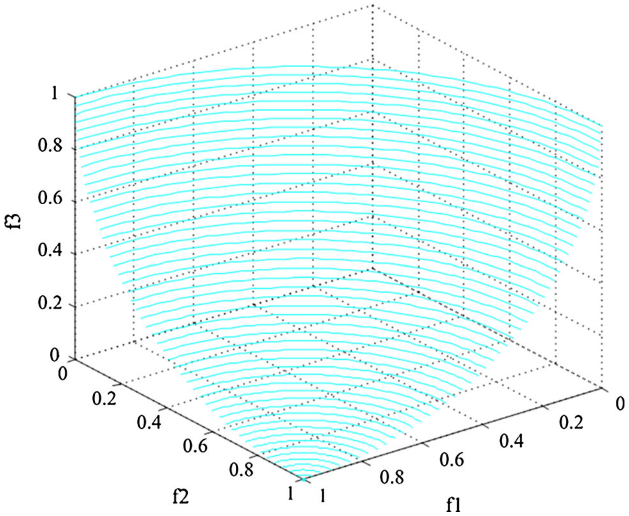

Fig. 8 PF of the modified DTLZ2 with 3 objectives

new DTLZ2 has 10 dimensions, which is the same as the original DTLZ2 problem. The stopping criterion is $E < 0.01$. Other parameters are shown in Sect. 4.1.

From Table 4, we find that *ECR* NSGA-II performed better than *Extremized Crowded* NSGA-II on this modified DTLZ2. Comparing to the results in Sect. 4.3, the nadir point of the modified DTLZ2 is harder to obtain than that of the original DTLZ2. *Extremized Crowded* NSGA-II cannot obtain the nadir point of the new DTLZ2 with more than 4 objectives. *ECR* NSGA-II cannot work well on the new DTLZ2 with more than 7 objectives. Such results show that the structure of PF influences significantly the behaviors of different nadir point estimation methods. Additionally, NSGA-III fails to solve the modified DTLZ2 with more than 2 objectives, because it assumes the extreme point as (1,0,…,0) rather than the true situation (0,1,…,1).

4.4 Experimental study on accuracy

From the results in Sect. 4.3, we now understand the general performance of these compared algorithms. However, the true nadir point is usually unknown in practice. Involving the true nadir point is only a method to evaluate the performance of different algorithms. We carry on the comparison of their accuracy in this section. The stopping criterion of the experiments is set as a fixed number of function evaluations.

A comparative experiment between *ECR* NSGA-II, *Extremized Crowded* NSGA-II, *ECR* TwoArch, and *ECR* IBEA on the WFG problems with 10 objectives for 900000 function evaluations is conducted. The metric E is calculated in every generation to show the results. The experiment runs for 29 independent times, and the average and standard deviation values of the metric E are given in Table 5. The results are analyzed by Kruskal–Wallis test with Tukey test (Derrac et al. 2011) in Table 6.

From the results on the E values and the nonparametric statistical test, TwoArch and IBEA are not suitable for estimating nadir points even after embedding ECR. The reason is that both algorithms do not have strategies to keep extreme points. NSGA-II has advantages in this aspect. Therefore, the E values of *ECR* NSGA-II and *Extremized Crowded* NSGA-II are much better than that of *ECR* TwoArch and *ECR* IBEA on all WFG problems.

The nadir points of the WFG problems are harder to be obtained than that of the DTLZ problems, especially WFG1, WFG3, WFG5, WFG6, WFG8, and WFG9. Although *ECR* NSGA-II performs better than *Extremized Crowded* NSGA-II on WFG1, WFG6, and WFG8, the advantages are not as significant as that on the DTLZ problems. The performance of *ECR* NSGA-II and *Extremized Crowded* NSGA-II are similar in the remaining WFG problems.

Figure 9 is the average E values of *ECR* NSGA-II, *Extremized Crowded* NSGA-II, *ECR* TwoArch, and *ECR* IBEA on WFG1 with 10 objectives in 29 independent runs. TwoArch and IBEA embedded with ECR are not suitable to estimate nadir points of the WFG problems. NSGA-II has significant advantages on nadir point estimation. As ECR keeps solutions in the critical regions, the diversity of *ECR* NSGA-II is better than *Extremized Crowded* NSGA-II. That is the reason why *ECR* NSGA-II has a faster convergence speed to the true nadir point than *Extremized Crowded* NSGA-II.

Table 4 Number of function evaluations of *ECR* NSGA-II, *Extremized Crowded* NSGA-II, and NSGA-III (the stopping criterion is $E < 0.01$) on the new DTLZ2 problem

Obj #	*ECR* NSGA-II			*Extremized crowded* NSGA-II			NSGA-III		
	Best	Median	Worst	Best	Median	Worst	Best	Median	Worst
2	5100	7500	8400	5100	6000	9000	**4200**	**4800**	**8700**
3	**11100**	**12000**	**15300**	82200	455700	900000	–	–	–
4	**20100**	**34500**	**55200**	900000	900000	900000	–	–	–
5	**70500**	**117600**	**243600**	–	–	–	–	–	–
6	**79800**	**188400**	**257100**	–	–	–	–	–	–
7	**153000**	**361200**	**487500**	–	–	–	–	–	–

The values in bold face means the best result
"–" means that the algorithm cannot obtain the nadir point within limited computational time

Table 5 E values of *ECR* NSGA-II, *Extremized Crowded* NSGA-II, *ECR* TwoArch, and *ECR* IBEA (the stopping criterion is 900000 function evaluations) on the WFG problems with 10 objectives

	ECR NSGA-II	*Extremized crowded* NSGA-II	*ECR* TwoArch	*ECR* IBEA
WFG1	0.1101 ± 0.0426	0.1583 ± 0.1016	0.8133 ± 0.0292	0.6187 ± 0.1112
WFG2	0.0006 ± 0.0014	0.0002 ± 0.0003	0.8537 ± 0.0102	0.6028 ± 0.1196
WFG3	0.1961 ± 0.0002	0.1962 ± 0.0000	0.9125 ± 0.0182	0.1962 ± 0.0000
WFG4	0.0000 ± 0.0000	0.0000 ± 0.0000	0.7771 ± 0.0258	0.1948 ± 0.0012
WFG5	0.0095 ± 0.0006	0.0095 ± 0.0004	0.4662 ± 0.2710	0.1576 ± 0.0461
WFG6	0.0121 ± 0.0080	0.0205 ± 0.0119	0.7763 ± 0.1091	0.1843 ± 0.0260
WFG7	0.0000 ± 0.0000	0.0001 ± 0.0001	0.7961 ± 0.0698	0.1918 ± 0.0027
WFG8	0.1472 ± 0.0250	0.1678 ± 0.0215	0.6486 ± 0.0658	0.1952 ± 0.0006
WFG9	0.1583 ± 0.0347	0.1526 ± 0.0595	0.8387 ± 0.0335	0.1865 ± 0.0068

Table 6 p vales of Kruskal–Wallis test with Tukey test on the results in Table 5, where A–D stands for *ECR* NSGA-II, *Extremized Crowded* NSGA-II, *ECR* TwoArch, and *ECR* IBEA

	A	B	C	D
WFG1				
A	0.00	0.98	0.00	0.02
B	0.98	0.00	0.00	0.06
C	0.00	0.00	0.00	0.26
D	0.02	0.06	0.26	0.00
WFG2				
A	0.00	1.00	0.00	0.04
B	1.00	0.00	0.00	0.02
C	0.00	0.00	0.00	0.27
D	0.04	0.02	0.27	0.00
WFG3				
A	0.00	0.41	0.00	0.03
B	0.41	0.00	0.00	0.62
C	0.00	0.00	0.00	0.10
D	0.03	0.62	0.10	0.00
WFG4				
A	0.00	0.39	0.00	0.00
B	0.39	0.00	0.00	0.22
C	0.00	0.00	0.00	0.27
D	0.00	0.22	0.27	0.00
WFG5				
A	0.00	1.00	0.00	0.02
B	1.00	0.00	0.00	0.03
C	0.00	0.00	0.00	0.41
D	0.02	0.03	0.41	0.00
WFG6				
A	0.00	0.88	0.00	0.01
B	0.88	0.00	0.00	0.09
C	0.00	0.00	0.00	0.27
D	0.01	0.09	0.27	0.00
WFG7				
A	0.00	0.26	0.00	0.00
B	0.26	0.00	0.00	0.26
C	0.00	0.00	0.00	0.26
D	0.00	0.26	0.26	0.00
WFG8				
A	0.00	0.72	0.00	0.01
B	0.72	0.00	0.00	0.13
C	0.00	0.00	0.00	0.27
D	0.01	0.13	0.27	0.00
WFG9				
A	0.00	0.65	0.00	0.31
B	0.65	0.00	0.00	0.94
C	0.00	0.00	0.00	0.02
D	0.31	0.94	0.02	0.00

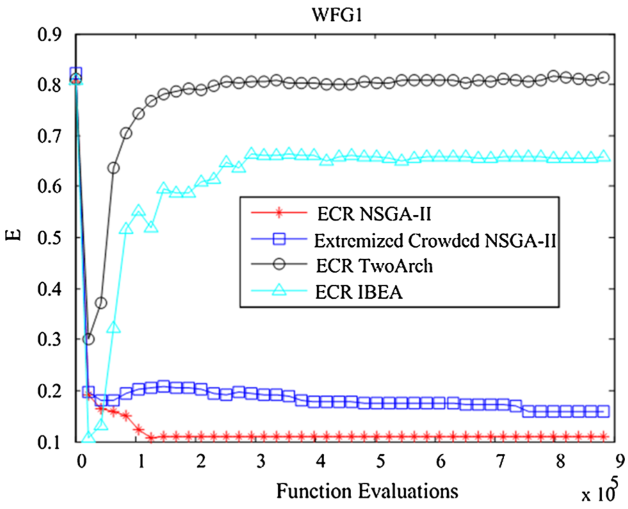

Fig. 9 The evolution of the average E values of *ECR* NSGA-II, *Extremized Crowded* NSGA-II, *ECR* TwoArch, and *ECR* IBEA on WFG1 with 10 objectives in 29 independent runs

5 Concluding remarks

Nadir point estimation based on emphasized critical regions is proposed in this paper, which can estimate nadir points of MaOPs with better performance than the other existing methods. It has been shown that the proposed method is capable of estimating the nadir point for the DTLZ2 with up to 35 objectives. In this paper, three major contributions have been made, as follows:

5.1 Emphasized critical regions

ECR emphasizes not only extreme points but also critical regions, which increases the diversity of the current population. For a MaOP, the objective space is large, and it is hard to select individuals only by the Pareto dominance relation. Therefore, the emphasized critical regions strategy is important for improving both diversity and convergence. Although the diversity in critical regions seems to be useless for finding nadir points directly, it improves the convergence to extreme points, from which better estimation of nadir points can be obtained.

5.2 Individual number assignment to critical regions

In ECR, the computational cost is assigned uniformly among all critical regions. Individual number assignment in critical regions in ECR maintains the balance of searching different critical regions.

5.3 Self-adaptive ε-clearing strategy

In ECR, the ε-clearing strategy is adopted self-adaptively, which deletes similar individuals to enhance the diversity of the population. Parameter ε can be set automatically according to the distribution of the current population.

Although ECR achieves good performance on the MaOPs tested here, there is room for improvement. For example, it has been found that our ECR method cannot solve the MOPs with more than 35 objectives; the performance on the WFG problems is less than satisfactory. Our future work includes: 1) Some local search strategies such as (Deb et al. 2010) could be added to ECR. 2) More studies should focus on the impact of the structures of PFs on the performance of ECR.

Acknowledgements This work was supported by the National Basic Research Program (973 Program) of China (No.2013CB329402), an EPSRC Grant (No. EP/J017515/1) on “DAASE: Dynamic Adaptive Automated Software Engineering”, the Program for Cheung Kong Scholars and Innovative Research Team in University (No. IRT1170), the National Natural Science Foundation of China (No. 61329302), National Science Foundation of China (Nos. 91438103 and 91438201), and the Fund for Foreign Scholars in University Research and Teaching Programs (the 111 Project) (No. B07048). Xin Yao was supported by a Royal Society Wolfson Research Merit Award.

Compliance with ethical standards

Conflict of interest The authors declared that they have no conflicts of interest to this work.

References

Amiri M, Ekhtiari M, Yazdani M (2010) Nadir compromise programming: a model for optimization of multi-objective portfolio problem. Expert Syst Appl 36(6):7222–7226

Bechikh S, Ben Said L, Ghedira K (2010) Estimating nadir point in multi-objective optimization using mobile reference points. In: Evolutionary computation (CEC), 2010 IEEE congress on, IEEE Press, pp 1–9

Ben Said L, Bechikh S, Ghédira K (2010) The r-dominance: a new dominance relation for interactive evolutionary multicriteria decision making. IEEE Trans Evol Comput 14(5):801–818

Benayoun R, De Montgolfier J, Tergny J, Laritchev O (1971) Linear programming with multiple objective functions: step method (STEM). Math Program 1(1):366–375

Branke J, Kaußler T, Smidt C, Schmeck H (2000) A multi-population approach to dynamic optimization problems. In: Evolutionary design and manufacture. Springer, pp 299–307

Branke J, Deb K, Dierolf H, Osswald M (2004) Finding knees in multi-objective optimization. In: Parallel problem solving from nature-PPSN VIII. Springer, pp 722–731

Branke J, Deb K, Miettinen K (2008) Multiobjective optimization: interactive and evolutionary approaches. Springer, New York

Chen N, Chen WN, Gong YJ, Zhan ZH, Zhang J, Li Y, Tan YS (2015) An evolutionary algorithm with double-level archives for multiobjective optimization. IEEE Trans Cybern 45(9):1851–1863

Dai J, Wang W, Xu Q (2013) An uncertainty measure for incomplete decision tables and its applications. IEEE Trans Cybern 43(4):1277–1289

Deb K, Jain H (2014) An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: solving problems with box constraints. IEEE Trans Evol Comput 18(4):577–601

Deb K, Kumar A (2007) Interactive evolutionary multi-objective optimization and decision-making using reference direction method. In: Proceedings of the genetic and evolutionary computation conference (GECCO 2007). ACM Press, pp 781–788

Deb K, Miettinen K (2009) A review of nadir point estimation procedures using evolutionary approaches: a tale of dimensionality reduction. In: Proceedings of the multiple criterion decision making (MCDM-2008) conference. Springer, pp 1–14

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

Deb K, Thiele L, Laumanns M, Zitzler E (2002b) Scalable multi-objective optimization test problems. In: Evolutionary computation (CEC), 2002 IEEE congress on, vol 1. IEEE Press, pp 825–830

Deb K, Chaudhuri S, Miettinen K (2006) Towards estimating nadir objective vector using evolutionary approaches. In: Proceedings of the genetic and evolutionary computation conference (GECCO 2006). ACM Press, pp 643–650

Deb K, Miettinen K, Sharma D (2009) A hybrid integrated multi-objective optimization procedure for estimating nadir point. In: Evolutionary multi-criterion optimization. Springer, pp 569–583

Deb K, Miettinen K, Chaudhuri S (2010) Toward an estimation of nadir objective vector using a hybrid of evolutionary and local search approaches. IEEE Trans Evol Comput 14(6):821–841

Derrac J, García S, Molina D, Herrera F (2011) A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. Swarm Evol Comput 1(1):3–18

Dessouky M, Ghiassi M, Davis W (1986) Estimates of the minimum nondominated criterion values in multiple-criteria decision-making. Eng Costs Prod Econ 10(1):95–104

Ehrgott M, Tenfelde-Podehl D (2003) Computation of ideal and nadir values and implications for their use in MCDM methods. Eur J Oper Res 151(1):119–139

Fan Z, Liu Y (2010) A method for group decision-making based on multi-granularity uncertain linguistic information. Expert Syst Appl 37(5):4000–4008

He Y, He Z, Chen H (2015) Intuitionistic fuzzy interaction bonferroni means and its application to multiple attribute decision making. IEEE Trans Cybern 45(1):116–128

Hsu S, Wang T (2011) Solving multi-criteria decision making with incomplete linguistic preference relations. Expert Syst Appl 38(9):10882–10888

Huband S, Hingston P, Barone L, While L (2006) A review of multi-objective test problems and a scalable test problem toolkit. IEEE Trans Evol Comput 10(5):477–506

Hughes E (2005) Evolutionary many-objective optimisation: many once or one many? In: Evolutionary computation (CEC), 2005 IEEE congress on, vol 1. IEEE Press, pp 222–227

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

Ishibuchi H, Masuda H, Tanigaki Y, Nojima Y (2014) Review of coevolutionary developments of evolutionary multi-objective and many-objective algorithms and test problems. In: Computational intelligence in multi-criteria decision-making (MCDM), 2014 IEEE symposium on. IEEE, pp 178–184

Jiang S, Ong YS, Zhang J, Feng L (2014) Consistencies and contradictions of performance metrics in multiobjective optimization. IEEE Trans Cybern 44(12):2391–2404

Ke L, Zhang Q, Battiti R (2013) MOEA/D-ACO: a multiobjective evolutionary algorithm using decomposition and antcolony. IEEE Trans Cybern 43(6):1845–1859

Khare V, Yao X, Deb K (2003) Performance scaling of multi-objective evolutionary algorithms. In: Fonseca C, Fleming P, Zitzler E, Thiele L, Deb K (eds) Evolutionary multi-criterion optimization. Lecture notes in computer science, vol 2632. Springer, Berlin, pp 376–390

Klamroth K, Miettinen K (2008) Integrating approximation and interactive decision making in multicriteria optimization. Oper Res 56(1):222–234

Korhonen P, Salo S, Steuer R (1997) A heuristic for estimating nadir criterion values in multiple objective linear programming. Oper Res 45(5):751–757

Li K, Kwong S, Zhang Q, Deb K (2015) Interrelationship-based selection for decomposition multiobjective optimization. IEEE Trans Cybern 45(10):2076–2088

Ma X, liu f, Qi Y, Wang X, Li L, Jiao L, Yin M, Gong M (2015) A multiobjective evolutionary algorithm based on decision variable analyses for multi-objective optimization problems with large scale variables. IEEE Trans Evolut Comput PP(99):1–1. doi:10.1109/TEVC.2015.2455812

Metev B, Vassilev V (2003) A method for nadir point estimation in MOLP problems. Cybern Inf Technol 3(2):15–24

Miettinen K (1999) Nonlinear multiobjective optimization. Springer, New York

Miettinen K, Eskelinen P, Ruiz F, Luque M (2010) NAUTILUS method: an interactive technique in multiobjective optimization based on the nadir point. Eur J Oper Res 206(2):426–434

Praditwong K, Yao X (2006) A new multi-objective evolutionary optimisation algorithm: the two-archive algorithm. In: Computational intelligence and security, 2006 international conference on, vol 1. IEEE Press, pp 286–291

Praditwong K, Yao X (2007) How well do multi-objective evolutionary algorithms scale to large problems. In: Evolutionary computation (CEC), 2007 IEEE congress on. IEEE Press, pp 3959–3966

Purshouse R, Fleming P (2003) Evolutionary many-objective optimisation: an exploratory analysis. In: Evolutionary computation (CEC), 2003 IEEE congress on, vol 3. IEEE Press, pp 2066–2073

Szczepanski M, Wierzbicki A (2003) Application of multiple criteria evolutionary algorithms to vector optimisation, decision support and reference point approaches. J Telecommun Inf Technol 3:16–33

Thiele L, Miettinen K, Korhonen P, Molina J (2009) A preference-based evolutionary algorithm for multi-objective optimization. Evol Comput 17(3):411–436

Wang H, Jiao L, Yao X (2015) Two_Arch2: an improved two-archive algorithm for many-objective optimization. IEEE Trans Evol Comput 19(4):524–541

Yuan Y, Xu H, Wang B, Yao X (2015a) A new dominance relation based evolutionary algorithm for many-objective optimization. doi:10.1109/TEVC.2015.2420112

Yuan Y, Xu H, Wang B, Zhang B, Yao X (2015b) Balancing convergence and diversity in decomposition-based many-objective optimizers. doi:10.1109/TEVC.2015.2443001

Zitzler E, Künzli S (2004) Indicator-based selection in multiobjective search. In: Parallel problem solving from nature-PPSN VIII. Springer, pp 832–842

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

Zitzler E, Deb K, Thiele L (2000) Comparison of multiobjective evolutionary algorithms: empirical results. Evol Comput 8(2):173–195