An Improved Multiobjective Differential Evolution based on Pareto-adaptive ϵ-dominance and Orthogonal Design

Wenyin Gong [a], Zhihua Cai [a,1]

[a] *School of Computer Science, China University of Geosciences, Wuhan 430074, P. R. China*

Abstract

Evolutionary multiobjective optimization has become a very popular topic in the last few years. Since the 1980s, various evolutionary approaches that are capable of searching for multiple solutions simultaneously in a single run have been developed to solve multiobjective optimization problems (MOPs). However, to find a *uniformly distributed*, *near-complete*, and *near-optimal* Pareto front in a small number of fitness function evaluations (NFFEs) is a challenging task for any multiobjective optimization evolutionary algorithm (MOEA). In this paper, we present an improved differential evolution algorithm to MOPs that combines several features of previous evolutionary algorithms in a unique manner. It is characterized by a) employing the orthogonal design method with quantization technique to generate the initial population, b) adopting an archive to store the nondominated solutions and employing the new Pareto-adaptive ϵ-dominance method to update the archive at each generation, c) storing the extreme points and inserting them into the final archive in order to remedy one of the limitations of ϵ-dominance: the loss of the extreme points in the final archive, and d) using a hybrid selection mechanism in which a random selection and an elitist selection are alternated in order to allow using the archive solution to guide the search towards the Pareto-optimal front. Experiments have been conducted on a number of unconstrained real-valued artificial functions of two and three objectives. The results prove the efficiency of our approach with respect to the quality of the approximation of the Pareto-optimal front and the considerable reduction of NFFEs in these test problems. By examining the selected performance metrics, our approach is found to be statistically competitive with five state-of-the-art MOEAs in terms of keeping the diversity of the individuals along the tradeoff surface, finding a well-approximated Pareto-optimal front and reducing the computational effort.

Key words: Multiobjective optimization; differential evolution algorithm; Pareto-adaptive ϵ-dominance; orthogonal design method

1. Introduction

In many real-world optimization applications, it is often hard to formulate the optimization goal as a scalar function. Typically, there are several criteria or objectives, and not unusually, these objectives stay in conflict with each other. As an example, in the design of an automobile an engineer may wish to maximize crash resistance for safety and minimize weight for fuel economy. Instead of finding a single solution, the multiobjective optimization methods try to produce a set of good trade-off solutions called the non-dominated solutions or Pareto-optimal solutions from which the decision maker may select one.

The Operations Research community has developed approaches to solve MOPs since the 1950s [1]. Currently, a wide variety of mathematical programming techniques to solve

[1] Corresponding author. Tel: +86-27-62359809
E-mail address: cug11100304@yahoo.com.cn (W. Gong), zhcai@cug.edu.cn (Z. Cai)

MOPs are available in the specialized literature. However, mathematical programming techniques have certain limitations when tackling MOPs (e.g., many of them are susceptible to the shape of the Pareto front and may not work when the Pareto front is concave or disconnected). Others require differentiability of the objective functions and the constraints. In addition, there are several stochastic search algorithms used to solve MOPs in literature, such as Simulated Annealing [2], Tabu Search [3], and so on. However, most of them only generate a single solution from each run. In contrast, evolutionary algorithms (EAs) deal simultaneously with a set of possible solutions (the so-called population) which allows us to find several members of the Pareto-optimal set in a single run of the algorithm. Additionally, EAs are less susceptible to the shape or continuity of the Pareto front. For example, they can easily deal with discontinuous and concave Pareto fronts.

Since the 1980s, several multiobjective evolutionary algorithms (MOEAs) have been proposed and applied in MOPs [4] - [15]. These algorithms share the same purpose - searching for a *uniformly distributed*, *near-complete*, and *near-optimal* Pareto front in a small NFFEs for a given MOP. However, this ultimate goal is far from being accomplished by existing MOEAs described in literature. In one respect, most of the MOPs are very complicated and require the computational resources to be homogenously distributed in a high-dimensional search space. On the other hand, those better-fitted individuals generally have strong tendencies to restrict searching efforts within local areas because of the "genetic drift" phenomenon, which results in the loss of diversity due to stochastic sampling. This phenomenon is a well-known tradeoff decision pertaining to the efficiency and efficacy dilemma [9].

Differential evolution (DE) [16] algorithm is a novel evolutionary algorithm for faster optimization, which mutation operator is based on the distribution of solutions in the population. DE has won the third place at the first International Contest on Evolutionary Computation on a real-valued function test-suite [17]. DE is a simple yet powerful population based, direct search algorithm with the generation-and-test feature for globally optimizing functions using real valued parameters. Note, however, that although DE is encoded by binary code to solve binary problems recently [18], it is mainly used for continuous optimization problems, and that we consider only continuous problems in this work. Among the DE's advantages are its simple structure, ease of use, speed and robustness. Price & Storn [16] gave the working principle of DE with single scheme. Later on, they suggested ten different schemes of DE [17]. It has been successfully used in solving single-objective optimization problems. Hence, several researchers have tried to extend it to handle MOPs. Such as Pareto DE [19], Pareto DE Approach [20], Multiobjective DE [21], DE for Multiobjective Optimization (DEMO) [22], GDE3 [23], ϵ-MyDE [24], and so on. A detailed survey of multiobjective DE has been published recently [24], in which the advantages and disadvantages of the most known multiobjective DE methods are discussed. However, all of previous approaches generated the initial population randomly and most of them were tested in problems with only two objectives.

In order to further extend DE to MOPs, in this paper, we extend our previous work [31] and present an improved multiobjective DE algorithm called $pa\epsilon$-ODEMO, which integrates established techniques in existing EA's in a single unique algorithm. The new approach uses the orthogonal design method with quantization technique to generate the initial population. Moreover, it adopts an archive to store the nondominated solutions and employs the new Pareto-adaptive ϵ-dominance method [32] to update the archive at each generation. Experiments were carried on 10 unconstrained real-valued artificial problems and compared with five state-of-the-art MOEAs, we show that $pa\epsilon$-ODEMO outperforms other algorithms in finding a *uniformly distributed*, *near-complete*, and *near-optimal* Pareto front in a small NFFEs when solving these problems. The simulation results show that $pa\epsilon$-ODEMO is competitive with the selective MOEAs measured by some performance metrics.

The rest of this paper is organized as follows. In Section 2, we give the problem formulation of MOPs; we also briefly introduce the ϵ-dominance used in MOEAs and orthogonal design method in EAs. In Section 3, we briefly describe five state-of-the-art MOEAs that are selected to compare with our approach. The DE algorithm is described in Section 4. Section 5 presents an improved $pa\epsilon$-ODEMO to deal with MOPs and describes its main components in detail. In Section 6, we test our algorithm through a number of two- and three-objective artificial test problems. In addition, the experiment results are compared with those of the seletive MOEAs. The last section, Section 7, is devoted to conclusions and future work.

2. Preliminary

2.1. *Problem formulation*

Without loss of the generality, an MOP includes a set of n decision variables, a set of k objective functions, and a set of m constraints. Objective functions and constraints are functions of the decision variables. The optimization goal is to

$$\begin{aligned} minimize: & \boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), \cdots, f_k(\boldsymbol{x})) \\ subject\ to: & \boldsymbol{e}(\boldsymbol{x}) = (e_1(\boldsymbol{x}), \cdots, e_m(\boldsymbol{x})) \geq 0 \\ where: & \boldsymbol{x} = (x_1, x_2, \cdots, x_n) \in \boldsymbol{X} \\ & \boldsymbol{y} = (y_1, y_2, \cdots, y_k) \in \boldsymbol{Y} \end{aligned} \quad (1)$$

where $\boldsymbol{x}$ is the decision vector, $\boldsymbol{y}$ is the objective vector, $\boldsymbol{X}$ denotes as the decision space, and $\boldsymbol{Y}$ represents the objective space. Generally, for each variable x_i it satisfies a constrained boundary

$$l_i \leq x_i \leq u_i, i = 1, 2, \cdots, n \quad (2)$$

The constraints $\boldsymbol{e}(\boldsymbol{x}) \geq 0$ determine the set of feasible solutions.

Definition 1 (*Pareto Dominance*) A vector $\boldsymbol{x} = (x_1, \cdots, x_k)$ is said to Pareto dominate another vector $\boldsymbol{y} = (y_1, \cdots, y_k)$, denoted as $\boldsymbol{x} \prec \boldsymbol{y}$, if and only if

$$\forall i \in 1, \cdots, k, x_i \leq y_i \quad \text{and} \quad \exists i \in 1, \cdots, k, x_i < y_i$$

Definition 2 (*Pareto Optimality*) A solution $\boldsymbol{x} \in \boldsymbol{X}$ is said to be Pareto optimal in $\boldsymbol{X}$ if and only if $\neg\exists \boldsymbol{y} \in \boldsymbol{X}, \boldsymbol{v} \prec \boldsymbol{u}$, where $\boldsymbol{u} = \boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), \cdots, f_k(\boldsymbol{x})), \boldsymbol{v} = \boldsymbol{f}(\boldsymbol{y}) = (f_1(\boldsymbol{y}), \cdots, f_k(\boldsymbol{y}))$.

Definition 3 (*ϵ-dominance*) Let $\boldsymbol{f}, \boldsymbol{g} \in \Re^k$. Then $\boldsymbol{f}$ is said to ϵ-dominance $\boldsymbol{g}$ for some $\epsilon > 0$, denoted as $\boldsymbol{f} \prec_\epsilon \boldsymbol{g}$, if and only if for all $i \in \{1, \cdots, k\}$, $(1-\epsilon) f_i \leq g_i$.

Definition 4 (*Pareto-optimal set*) The Pareto-optimal set *POS* is defined as the set of all Pareto-optimal solutions, i.e., $POS = \{\boldsymbol{x} \in \boldsymbol{X} | \neg\exists \boldsymbol{y} \in \boldsymbol{X}, \boldsymbol{f}(\boldsymbol{y}) \prec \boldsymbol{f}(\boldsymbol{x})\}$.

Definition 5 (*Pareto-optimal front*) The Pareto-optimal front *POF* is defined as the set of all objective functions values corresponding to the solutions in *POS*, i.e., $POF = \{\boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), \cdots, f_k(\boldsymbol{x})) | \boldsymbol{x} \in \boldsymbol{X}\}$.

2.2. *ϵ-domination based MOEAs*

To achieve a better diversity in multiobjective optimization, one of the relaxed forms of Pareto dominance, ϵ-dominance introduced in [30], has become popular in the last few years. The ϵ-dominance acts as an archiving strategy to ensure both properties of convergence towards the Pareto-optimal set and properties of diversity among the solutions found. Deb *et al.* [28,29] proposed a steady-state MOEA, ϵ-MOEA, based on the ϵ-dominance concept and efficient parent and archive update strategies to MOPs. The simulation results indicated that ϵ-MOEA is a good compromise in terms of convergence near to the Pareto-optimal front, diversity of solutions, and computational time. The authors concluded that the use of ϵ-dominance criterion has been found to have two advantages: (i) it helps in reducing the cardinality of Pareto-optimal region and (ii) it ensures that no two obtained solutions are within an ϵ_i from each other in the i-th objective. Later, Santana-Quintero *et al.* [24] and Cai *et al.* [31] incorporated the ϵ-dominance concept into DE algorithm to solve MOPs. However, the above-mentioned ϵ-domination based MOEAs do not overcome the main limitation of ϵ-dominance: the loss of several nondominated solutions from the hypergrid adopted in the archive because of the way in which solutions are selected within each box [32]. In order to remedy the limitation, Hernández-Díaz *et al.* [32] proposed a new Pareto-adaptive ϵ-dominance method, called $pa\epsilon$-dominance, where different ϵ-dominance regions depending on the geometrical characteristics of the Pareto-optimal front is used. This method remedies some limitations of the original ϵ-dominance and can finds a higher number of efficient points. However, so as to use the $pa\epsilon$-dominance method, an initial Pareto front approximation, denoted by F, must be generated [32]. The number of efficient points in F can be critical for the final performance. If F is not generated efficiently, the final performance may be very poor.

2.3. *Orthogonal design method in EAs*

In a discrete single objective optimization problem, when there are N factors (variables) and each factor has Q levels, the search space consists of Q^N combinations of levels. When N and Q are large, it may not be possible to do all Q^N experiments to obtain optimal solutions. Therefore, it is desirable to sample a small, but representative set of combinations for experimentation, and based on the sample, the optimal may be estimated. The orthogonal design was developed for the purpose [25]. The selected combinations are scattered uniformly over the space of all possible combinations Q^N.

Recently, some researchers applied the orthogonal design method incorporated with EAs to solve optimization problems. Leung *et al.* [26] incorporated orthogonal design in GA for numerical optimization problems and found such method was more robust and statistically sound than the classical GAs. OMOEA [27] presented by Zeng *et al.* adopted the orthogonal design method to solve MOPs. In OMOEA, it uses the orthogonal design method to generate a group of sub-niches, every subniche evolves at each generation. Because the orthogonal arrays (OAs) must be generated at each generation, OMOEA is very time-consuming. Cai *et al.* [31] proposed a novel multiobjective DE, ϵ-ODEMO, which uses the orthogonal design method to generate the initial population and adopts ϵ-dominance to update the archive. Experimental results indicate that ϵ-ODEMO is very efficient in terms of convergence near to the Pareto-optimal front, diversity of solutions, and computational time. However, there are three limitations of ϵ-ODEMO, (i) it may lose some nondominated solutions in the final archive; (ii) the choice of the ϵ-vector is difficult of this approach; and (iii) the extreme points are lost in the final archive.

3. Multiobjective Optimization Evolutionary Algorithms

Generally, the approximation of the Pareto-optimal set involves two conflicting objectives: the distance to the true Pareto-optimal front is to be minimized, while the diversity of the generated solutions is to be maximized [7]. To address the first objective, a Pareto-based fitness assignment method is usually designed in many existing MOEAs in order to guide the search toward the true Pareto-optimal front. For the second objective, some successful MOEAs provide density estimation methods to preserve the population diversity. Recently, a relaxed form of dominance for MOEAs, named ϵ-dominance, has been used [28] - [32]. This mechanism acts as an archiving strategy to ensure both properties of convergence towards the Pareto-optimal set and properties of diversity among the solutions found. These methods and techniques can be found in five state-of-the-art MOEAs - NSGA-II, SPEA2, DEMO, ϵ-MOEA, and $pa\epsilon$-MyDE - which are briefly reviewed in the following.

(i) **Nondominated Sorting Genetic Algorithm II (NSGA-II)**. NSGA-II [5] was advanced from its origin, NSGA [6]. In NSGA-II, a nondominated sorting approach is used for each individual to create Pareto rank, and a crowding distance assignment method is applied to implement density estimation. In a fitness assignment between two individuals, NSGA-II prefers the point with a lower rank value, or the point located in a region with fewer number of points if both of the points belong to the same front. Therefore, by combining a fast nondominated sorting approach, an elitist scheme and a parameterless sharing method with the original NSGA, NSGA-II claims to produce a better spread of solutions in some testing problems [5].

(ii) **Strength Pareto Evolutionary Algorithm 2 (SPEA2)**. SPEA2 [7] has three main differences with respect to its predecessor, SPEA [4]: (i) it incorporates a fine-grained fitness assignment strategy which takes into account for each individual the number of individuals that dominate it and the number of individuals by which it is dominated; (ii) it uses a nearest neighbor density estimation technique which guides the search more efficiently; and (iii) it has an enhanced archive truncation method that guarantees the preservation of boundary solutions. In the experimental results, SPEA2 shows better performance than SPEA over all the test functions considered therein.

(iii) **Differential Evolution for Multi-Objective Optimization (DEMO)**. DEMO was proposed in [22]. This algorithm combines the advantages of DE with the mechanisms of Pareto-based ranking and crowding distance sorting. DEMO only maintains one population and it is extended when newly created candidates take part immediately in the creation of the subsequent candidates. This enables a fast convergence towards the true Pareto-optimal front, while the use of nondominated sorting and crowding distance (derived from the NSGA-II [5]) of the extended population promotes the uniform spread of solutions. DEMO is compared in five ZDT problems. It outperforms in some problems to some MOEAs.

(iv) **ϵ-MOEA**. This approach was proposed in [29], it consists of a steady-state GA which maintains an archive of nondominated individuals. Note however, that this algorithm does not use the Pareto dominance relation when updating the archive. Instead, it uses the ϵ-dominance to update the archive at each generation. One parent is selected from the main population and other from the archive. Then, an offspring is produced and it is allowed to enter into the archive if ϵ-dominates at least one element of the archive, and if no archive member ϵ-dominates it. It has been found to be a very competitive MOEA.

(v) **Pareto-adaptive ϵ-dominance ($pa\epsilon$-MyDE)**. This algorithm was proposed by Hernández-Díaz *et al.* [32], which is a revised version of ϵ-MyDE [24]. In ϵ-MyDE, it consists of an extension of the DE algorithm [16] used to solve MOPs. The operators typically adopted in DE are incorporated into this approach, but the algorithm is extended with an archive (or secondary population) which is used to retain the nondominated solutions obtained during the evolutionary process. Also, ϵ-dominance is incorporated in order to get a well-distributed set of solutions along the Pareto front. In $pa\epsilon$-MyDE, the Pareto-adaptive ϵ-dominance is adopted instead of the original ϵ-dominance used in ϵ-MyDE. This method remedies some limitations of the original ϵ-dominance and can finds a higher number of efficient points. Moreover, the solutions obtained by $pa\epsilon$-MyDE are better uniformly distributed along the Pareto front than those of ϵ-MyDE.

4. Differential Evolution Algorithm

DE algorithm [16] is a simple evolutionary algorithm which creates new candidate solutions by combining the parent individual and several other individuals of the same population. A candidate replaces the parent only if it has better fitness. This is a rather greedy selection scheme which often outperforms traditional EAs. DE is a simple yet powerful population based, direct search algorithm with the generation-and-test feature for globally optimizing functions using real valued parameters. It has been successfully used in solving single-objective optimization problems. Among the DE's advantages are its simple structure, ease of use, speed and robustness.

The DE algorithm in pseudo-code is shown in Algorithm 1. Where NP is size of the evolutionary population. n is the number of the decision variables. CR is the probability of crossover operator. F is the scaling factor. rndint$(1, n)$ is a randomly chosen index $\in 1, 2, \cdots, n$ which ensures that U^i gets at least one parameter from the mutant vector. $\text{rnd}_j[0, 1)$ is the j-th evaluation of a uniform random number generator from $[0, 1)$.

Many variants of the classic DE have been proposed, which use different learning strategies and/or recombination operations in the reproduction stage [17]. In order to distinguish among its variants, the notation $DE/a/b/c$ is used, where "a" specifies the vector to be mutated (which can be random or the best vector); "b" is the number of difference vectors used; and "c" denotes the crossover scheme, *binomial* or *exponential*. The binomial crossover scheme is represented in Algorithm 1 and in case of exponential crossover, the crossover probability CR regulates how many consecutive mutated genes are copied to the trial individual U^i. Using this notation, the DE strategy described in Algorithm 1 above can be denoted as DE/rand/1/bin. Other well-known variants are DE/best/1/bin, DE/rand/2/bin, and DE/best/2/bin which can be implemented by (3) - (5), respectively. Again, each of the above algorithms can be configured to use the exponential crossover

Algorithm 1 DE algorithm with DE/rand/1/bin scheme

```
Generate the initial population of NP individuals P(0)
Evaluate the fitness for each individual in P(0)
while The halting criterion is not satisfied do
  for i = 1 to NP do
    Select uniform randomly r1 ≠ r2 ≠ r3 ≠ i
    j_rand = rndint(1, n)
    for j = 1 to n do
      if rnd_j[0, 1) < CR or j = j_rand then
        U_j^i = X_j^r1 + F × (X_j^r2 − X_j^r3)
      else
        U_j^i = X_j^i
      end if
    end for
    Evaluate the offspring U^i
    if U^i is better than X^i then
      X^i = U^i
    end if
  end for
end while
```

$$U^i = X^{best} + F \times (X^j - X^h) \quad (3)$$

$$U^i = X^j + F \times (X^h - X^l) + F \times (X^s - X^t) \quad (4)$$

$$U^i = X^{best} + F \times (X^j - X^h) + F \times (X^l - X^s) \quad (5)$$

where X^{best} represents the best individual in the current generation, i, j, h, l, s and $t \in \{1, \cdots, n\}$, and $i \neq j \neq h \neq l \neq s \neq t$.

5. An Improved Multiobjective DE Approach: $pa\epsilon$-ODEMO

As stated in the literature review, the main deficiency in the existing MOEAs lies on designing a suitable fitness assignment strategy in order to search for a *uniformly distributed*, *near-complete*, and *near-optimal* approximated Pareto front for the given optimization problem. Unfortunately, these objectives are contradictory. Inspired by the ideas from the orthogonal design method successfully used in EAs ([26], [27], and [31]) and $pa\epsilon$-dominance proposed in [32], in this work, we extend our previous work ϵ-ODEMO [31] and present an improved DE algorithm to solve MOPs, which integrates established techniques in existing EA's in a single unique algorithm. Our proposed DE algorithm is named $pa\epsilon$-ODEMO. Compared with ϵ-ODEMO, there are three improvements in $pa\epsilon$-ODEMO.

- The ϵ-dominance method used in ϵ-ODEMO is replaced by $pa\epsilon$-dominance method in $pa\epsilon$-ODEMO based on two considerations: (i) $pa\epsilon$-dominance can overcome the main limitation of ϵ-dominance, hence it can obtain a better diversity; and (ii) $pa\epsilon$-dominance is able to avoid tuning the ϵ vector required by ϵ-dominance method.
- In $pa\epsilon$-ODEMO, an improved hybrid selection mechanism is proposed in order to use the archive members to guide the search. However, in ϵ-ODEMO, at each generation every archive member is used to generate its offspring and to update the archive and evolutionary population, so it may result in misleading the search and high selection pressure.
- The extreme points are retained in $pa\epsilon$-ODEMO, and hence it can improve the diversity performance.

Our approach is similar to $pa\epsilon$-MyDE [32]. However, there are three main differences compared with $pa\epsilon$-MyDE.

- In $pa\epsilon$-ODEMO, the orthogonal population initialization is used in order to generate an initial Pareto front approximation with the higher number of efficient points as soon as possible, and hence, to make the $pa\epsilon$-dominance more efficient.
- In our approach, an improved hybrid selection mechanism is proposed. Compared with that of $pa\epsilon$-MyDE, the improved method has lower selection pressure. In addition, the evolutionary population, which contains useful information of evolution, is still valid in the elitist selection.
- The extreme points are retained in $pa\epsilon$-ODEMO, and hence it can improve the diversity performance.

5.1. *Orthogonal Initial Population*

As mentioned above, in order to use the $pa\epsilon$-dominance method, an initial Pareto front approximation F must be generated [32]. The number of efficient points in F can be critical for the final performance. Obviously, the higher the number of the efficient points in F the better performance of the grid generated. However, before solving an optimization problem, we usually have no information about the location of the global minimum. It is desirable that an algorithm starts to explore those points that are scattered evenly in the solution space. In our presented manner, the algorithm can scan the solution space once to locate good points for further exploration in subsequent iterations. As the algorithm iterates and improves the population of points, some points may move closer to the global minimum. Based on these considerations, in order to obtain an efficient F to generate the the first grid as soon as possible, we apply the quantization technique and the orthogonal design method to generate this initial archive and evolutionary population (EP).

5.1.1. *Design of the orthogonal array*

To design an orthogonal array (OA), in this research, we use $L_R(Q^C)$ to denote the OA with different level Q, where Q is odd and use $R = Q^J$ to indicate the number of the rows of OA, where J is a positive integer fulfilling

$$C = \frac{Q^J - 1}{Q - 1} \tag{6}$$

C denotes the number of the columns of OA in the above equation. The OA needs to find a proper J and Q to satisfy

$$\begin{aligned} minimize &: R = Q^J \\ subject\ to &: C = \frac{Q^J - 1}{Q - 1} \geq n \\ & R \geq NP \end{aligned} \tag{7}$$

where n is the number of the variables, NP is the size of the evolutionary population. In this study, we adopt the algorithm described in [26] to construct an OA. In particular, we use $L(R, C)$ to indicate the OA; $a_{i,j}$ to denote the level of the jth factor in the ith combination in $L(R, C)$. If $C > n$, we delete the last $C - n$ columns to get an OA with n factors. The algorithm to generate the OA is described in Algorithm 2.

5.1.2. *Quantization*

For one decision variable X_j with the boundary $[l_j, u_j]$, we quantize the domain into Q levels $\alpha_1^j, \alpha_2^j, \cdots, \alpha_Q^j$, where the design parameter Q is odd and α_i is given by

$$\alpha_k^j = l_j + (k-1)(\frac{u_j - l_j}{Q - 1}), 1 \leq k \leq Q \tag{8}$$

In other words, the domain $[l_j, u_j]$ is quantized $Q-1$ fractions, and any two successive levels are same as each other.

5.1.3. *Generation of Initial Population*

After constructing a proper OA and quantizing the domain of each decision variable, we can generate the orthogonal popula-

Algorithm 2 Construction of Orthogonal Array

/* Construct the basic columns */
for $k = 1$ to J **do**
  $j = \frac{Q^{k-1}-1}{Q-1} + 1$
  for $i = 1$ to R **do**
    $a_{i,j} = \lfloor \frac{i-1}{Q^{J-k}} \rfloor \bmod Q$
  end for
end for
/* Construct the nonbasic columns */
for $k = 2$ to J **do**
  $j = \frac{Q^{k-1}-1}{Q-1} + 1$
  for $s = 1$ to $j - 1$ **do**
    for $t = 1$ to $Q - 1$ **do**
      for $i = 1$ to R **do**
        $a_{i,(j+(s-1)(Q-1)+t)} = (a_{i,s} \times t + a_{i,j}) \bmod Q$
      end for
    end for
  end for
end for
Increment $a_{i,j}$ by one for all $i \in [1, R]$ and $j \in [1, C]$

tion (OP) which can scatter uniformly over the solution space. The algorithm for generating the OP is described in Algorithm 3, where $OP_{i,j}$ is the j-th variable of the i-th individual of OP, R is the number of rows of OA, n is the number of decision variables, and $eval$ is the current NFFEs. Generally, the number of the rows R of the OA is larger than the population size NP, so we create the initial archive with the nondominated solutions from OP first. Then we generate the initial EP from the initial archive and OP with the following procedure (more details are described in Algorithm 3). If $ar\_size > NP$, we select NP solutions from the initial archive randomly; or all of the ar_size solutions in the initial archive are inserted into EP, and the remainder $NP - ar\_size$ solutions are selected from OP randomly. Compared with OGA/Q [26], our approach and OGA/Q differ in two aspects with respect to the generation of initial population: (i) our approach integrates the orthogonal design method within MOEAs, and (ii) in our approach there are two population, archive and EP, we first generate the initial archive with the nondominated solutions from OP, and then the initial EP is generated from the initial archive and OP. It is worth to point out that for the given Q and J, the OA generated by Algorithm 2 is fixed, so we can generate the OA offline. Furthermore, for a given MOP, we only need to generate a minimal OA which satisfies Eqn. (7), so that the parameters Q and J are very easy to be selected for different problems.

Algorithm 3 Construction of Initial Archive and Evolutionary Population

/* Construction of orthogonal population (OP) */
$eval = 0, ar\_size = 0$
for $i = 1$ to R **do**
  for $j = 1$ to n **do**
    $k = a_{i,j}$
    $OP_{i,j} = \alpha_k^j$
  end for
  Evaluate OP_i and $eval$++
end for
/* Construction of initial archive (AR) */
Find all of the nondominated solutions of OP
Insert them into the initial archive, now $ar\_size > 0$
/* Construction of initial evolutionary population (EP) */
if $ar\_size \geq NP$ **then**
  Randomly select NP solutions from AR to generate EP
else
  Insert all of the ar_size solutions of AR into EP
  Select the remainder $NP - ar\_size$ solutions from EP randomly
  Insert them into EP
end if

5.2. *Archiving the Candidate Solutions*

In the study of Zitzler *et al.* [33], it was clearly shown that elitism helps in achieving better convergence in MOEAs. In $pa\epsilon$-ODEMO, the elitist scheme is also adopted through maintaining an external archive of nondominated solutions found in evolutionary process. In order to achieve a better diversity, in this work, the Pareto-adaptive ϵ-dominance, the so-called $pa\epsilon$-dominance proposed by Hernández-Díaz *et al.* [32], is used to update the archive. At each generation, so as to include a solution into this archive, it is compared with each member already contained in the archive using $pa\epsilon$-dominance after the grid is generated. The procedure is described as follows.

Every solution in the archive is assigned an identification array ($\boldsymbol{B} = (B_1, B_2, \cdots, B_k)^T$, where k is the total number of objectives) as follows:

$$B_i(\boldsymbol{f}) = \left\lfloor \frac{\log(\frac{\epsilon_1^i p^{v_i} - (p^{v_i}-1) f_i}{\epsilon_1^i})}{\log(\frac{1}{p^{v_i}})} + 1 \right\rfloor \quad (9)$$

where, p controls the shape of the curve (or surface), T is the number of points desired by the decision maker, ϵ_1^i is the size of the first box for each dimension, and v_i controls the speed of variation. These parameters satisfy the following equations:

$$\begin{cases} \epsilon_1^i = \dfrac{(p^{v_i}-1)p^{(T-1)v_i}}{p^{Tv_i-1}} \\ (1-2^{1/p})p^{Tv_i} + 2^{1/p}p^{Tv_i/2} - 1 = 0 \end{cases} \quad (10)$$

The identification array divides the whole objective space into hyper-boxes. With the identification arrays calculated for the offspring c and each archive member a, the offspring c updates the archive as Algorithm 4 described, where $\boldsymbol{B}_a$ indicates the identification array of solution a (more details of this procedure can be found in [24], [29]).

Algorithm 4 Updating the Archive with $pa\epsilon$-dominance

```
if B_c of the offspring dominates B_a of any archive member a then
    Delete all of the dominated archive members
    Accept the offspring c
else if B_c is dominated by B_a of any archive member a then
    Reject c
else
    if c shares the same grid with an archive member a then
        if c dominates a or c is closer to the grid than a then
            Delete a from the archive and accept c
        else
            Reject c
        end if
    else
        Insert c into the archive
    end if
end if
```

Using the $pa\epsilon$-dominance method, we can maintain the good properties of the original ϵ-dominance, such as ensuring both properties of convergence towards the Pareto-optimal set and properties of diversity among the solutions found in a small computation time, while overcoming the main limitation of ϵ-dominance: the loss of several nondominated solutions from the hypergrid adopted in the archive.

5.3. *Hybrid Selection Mechanism*

In [30], Laumanns *et al.* concluded that the archived members are really guaranteed to be the best solutions found. As the archive is used to maintain the nondominated solutions in $pa\epsilon$-ODEMO, a key issue must be addressed is that how to allow the archive members to take part in the generating process. In ϵ-MOEA [28], [29], Deb *et al.* randomly picked a solution from the archive for mating starting from the evolutionary process. In ϵ-MyDE [24], Santana-Quintero *et al.* proposed two selection mechanisms, where the random selection and an elitist selection are alternated. At the beginning, all of the parents for mating are randomly selected from the EP to generate the offspring. When the current generations is larger than a pre-defined number, the elitist selection is used and all of the parents are randomly selected from the archive to generate the offspring. However, these two techniques have a limitation: the selection pressure is very high. Especially, in ϵ-MyDE, the evolutionary population is not used at all when the elitist selection is adopted.

In our proposed approach, in order to regulate the selection pressure we propose a hybrid selection mechanism, which is similar to the method used in ϵ-MyDE, in which a random selection and an elitist selection are alternated. The difference between ϵ-MyDE and our approach is that in elitist selection we only randomly choose one solution from the archive as the base parent, X^{r_1}, in DE/rand/1/bin scheme described in Algorithm 1. And the other two parents, X^{r_2} and X^{r_3}, are selected from EP randomly. We use a *selection parameter* $\lambda \in [0.1, 1.0]$ to regulate the selection pressure.

$$\text{selection} = \begin{cases} \text{random selection}, & eval < (\lambda \times Max\_eval) \\ \text{elitist selection}, & \text{otherwise} \end{cases} \quad (11)$$

where, $eval$ is the current NFFEs, and Max_eval is the maximal NFFEs pre-defined by the user. In our proposed hybrid selection mechanism, one archive solution, which is disturbed by the two randomly selected solutions from EP, is selected to take part in the generating process when the elitist selection is used. In this manner, our approach has three advantages: (i) it can guarantee to be the best solutions found; (ii) because we do not use the archive solution to guide the search at the beginning of evolutionary process, it can avoid being misled by the inefficient archive solutions; and (iii) the solutions in EP can also guide the search in the elitist scheme.

5.4. *Storage of the Extreme Points*

Although the $pa\epsilon$-dominance can remedy some limitations of ϵ-dominance, it may also lose the extreme points for some MOPs. In $pa\epsilon$-ODEMO, we propose a simple strategy to maintain the extreme points in the final archive. First, we find all of the initial extreme points in the initial archive. Second, the dominated extreme points are updated by the offspring. Finally, when the maximal NFFEs is arrived, insert the extreme points into the final archive if they are not in the archive and non-dominated by any archive member. There are three independent procedures described in Algorithm 5, where k is the number of the objectives, AR indicates the archive, $AR_i.f_j$ is the value of the j-th objective of the i-th archive solution, and E_j is the j-th extreme point in array E.

Algorithm 5 Storage of the Extreme Points

```
/* find_extreme(): find the initial extreme points in the initial archive AR */
for i = 1 to ar_size do
    for j = 1 to k do
        Find the solution A with the minimal value of AR_i.f_j
        E_j = A
    end for
end for

/* update_extreme(): update the extreme points with the offspring */
for j = 1 to k do
    if The offspring c dominates E_j then
        E_j = c
    end if
end for

/* store_extreme(): store the extreme points in the final archive */
for j = 1 to k do
    if E_j is not in the final archive then
        if E_j is not dominated by any solution in the final archive then
            Insert E_j into the final archive
        end if
    end if
end for
```

5.5. *Handling the Constraint of the Variables*

After using the DE/rand/1/bin scheme to generate a new solution, if one or more of the variables in the new solution are outside their boundaries, i.e. $x_i \notin [l_i, u_i]$, the following repair rule is applied:

$$x_i = \begin{cases} l_i + \mathrm{rnd}_i[0,1] \times (u_i - l_i) & \text{if } x_i < l_i \\ u_i - \mathrm{rnd}_i[0,1] \times (u_i - l_i) & \text{if } x_i > u_i \end{cases} \tag{12}$$

where $\mathrm{rnd}_i[0,1]$ is the uniform random variable from [0,1] in each dimension i.

Algorithm 6 Main procedure of the proposed $pa\epsilon$-ODEMO

```
Construct a proper OA and generate the orthogonal population OP
Create the initial archive AR_1 with the nondominated solutions of OP
Create the initial evolutionary population EP_1 from AR_1 and OP
Find the extreme points from AR_1 // find_extreme()
t = 1, flag = 0
while eval < Max_eval do
    child_size = 0
    for i = 1 to NP do
        if eval < λ × Max_eval then
            Random selection
            Produce the offspring c with DE/rand/1/bin scheme
        else
            Elitist selection
            Produce the offspring c with DE/rand/1/bin scheme
        end if
        Evaluate the offspring c and eval++
        if the offspring c dominates the target parent EP_t^i then
            EP_t^i = c
        else if c is nondominated by EP_t^i then
            Add c to the child population CP
            child_size++
        else
            Discard c
        end if
        if flag == 0 then
            Update the archive with the usual dominance
        else
            Update the archive with paϵ-dominance
        end if
        Update the extreme points // update_extreme()
    end for
    if child_size ≠ 0 then
        Combine CP and EP_t
        Prune the mixed population using nondominated ranking method only
        Get the next evolutionary population EP_{t+1}
    end if
    if ar_size ≥ N_F and flag == 0 then
        Generate the paϵ-dominance grid
        flag = 1
    end if
    t++
end while
Add the extreme points in the final archive // store_extreme()
```

5.6. *Main Procedure of $pa\epsilon$-ODEMO*

For an MOP, the proposed $pa\epsilon$-ODEMO algorithm works as Algorithm 6. First, the temporary orthogonal population (OP) is created using Algorithm 3, all of the nondominated solutions of OP are inserted into the initial archive (AR). Then the initial evolutionary population (EP) is created from AR and OP. Also, the initial extreme points are found from AR. At each generation, an offspring is generated using DE/rand/1/bin scheme. The offspring replaces the parent *immediately* if the parent is dominated by the offspring. If the parent dominates the offspring, the offspring is discarded. Otherwise (when the offspring and parent are nondominated with regard to each other), the offspring is added to a temporary child population (CP). Meanwhile, the extreme points are updated by the offspring. If the first grid is not generated, i.e. $ar\_size < N\_F$, update the archive with *usual* dominance concept. Otherwise, if the first grid has been generated, update the archive with $pa\epsilon$-dominance concept as described in Algorithm 4. This step is repeated until NP number of offspring are created. After that, combine the temporary child population CP with EP, and we get a population of the size between NP and $2\times NP$. If the population has enlarged, we truncate it to prepare it for the next step of the algorithm.

The truncation sorts the individuals with nondominated sorting and if the individuals belong to the same front we only select them randomly. This is different from NSGA-II [5] and DEMO [22] to evaluate the individuals of the same front with the crowding distance metric. The reason is that in our approach the diversity is maintained in the archive with $pa\epsilon$-dominance. The truncation procedure keeps in EP only the best NP individuals (with regard to the nondominated sorting metric).

If the initial Pareto front approximation F is created, i.e. ar_size is larger than N_F that is the size of F, and the first grid is not generated, then we use the $pa\epsilon$-dominance concept to generate the grid (the algorithm is omitted here, interest readers can refer [32] for more details).

Finally, when the loop is terminated, $pa\epsilon$-ODEMO combines the extreme points with the final archive using the routine store_extreme() as described in Algorithm 5.

6. Simulation Results

In this section, we select five bi-objective and five tri-objective artificial benchmark problems to compare the performance of our proposed $pa\epsilon$-ODEMO with five state-of-the-art MOEAs - NSGA-II (only the real code NSGA-II is considered in this study), ϵ-MOEA, DEMO, $pa\epsilon$-MyDE, and SPEA2. For NSGA-II, ϵ-MOEA, DEMO, $pa\epsilon$-MyDE, and SPEA2, we have identical parameter settings as suggested in the original studies. For $pa\epsilon$-ODEMO, we have chosen a reasonable set of value and have not made any effort in finding the best parameter setting. We leave this task for a future study.

6.1. *Parameter Settings*

All approaches are only run for a maximum of 5,000 NFFEs on all test problems. For different approaches, the parameter settings are as follows:

- For real code NSGA-II [5] and ϵ-MOEA [29], the simulated binary crossover (SBX) and polynomial mutation are used. The crossover probability of $p_c = 0.9$ and a mutation probability of $p_m = 1/n$ (where n is the number of decision variables). The distribution indexes for crossover and mutation operators set as $\eta_c = 20$ and $\eta_m = 20$, respectively. The population size of $NP = 100$.
- For DEMO [22], there are three variants proposed in [22]. In this study, we only select the DEMO/parent for comparison, because the performance of this variant is not worse than the other two variants. The crossover probability of $p_c = 0.3$ and the scaling factor of $F = 0.5$. NP is set to 100.
- For $pa\epsilon$-MyDE [32], the crossover probability of $p_c = 0.95$ and the scaling factor of $F = 0.5$. A mutation probability of $p_m = 1/n$. $NP = 100$. The size of initial Pareto front approximation F is $N\_F = 100$.
- For SPEA2 [7], we use a population of size 80 and an external population of size 20 (this $4:1$ ration is suggested by the developer of SPEA2 to maintain an adequate selection pressure for the elite solutions). The SBX and polynomial mutation operators are used. And the crossover probability of $p_c = 0.9$ and a mutation probability of $p_m = 1/n$. The distribution indexes for crossover and mutation operators set as $\eta_c = 20$ and $\eta_m = 20$, respectively.
- For $pa\epsilon$-ODEMO, the crossover probability of $p_c = 0.1$ and the scaling factor of $F = 0.5$. $NP = 100$, $N\_F = 100$. The number of points desired by the decision maker of $T = 100$. The selection parameter of $\lambda = 0.1$. To generate a minimal OA, we use $J = 2$, and $Q = n - 1$ (except for ZDT4, ZDT6 and DTLZ1, we set $Q = 11$ to satisfy Eqn. (7) abovementioned).

6.2. *Performance Metrics*

Unlike in single-objective optimization, there are two goals in a multiobjective optimization: (i) convergence to the Pareto-optimal front, and (ii) maintenance of diversity in solutions of the Pareto-optimal set. These two tasks cannot be measured adequately with one performance metric. An analysis and review of existing performance metrics can be found in [36].

In order to make a fair comparison with other MOEAs, we first use two unary performance metrics proposed by Deb *et al.* [5] to assess the performance. The two metrics are derived from the final generation of 50 independent runs with different random seeds to benchmark the comparison results via statistical box plots. For these metrics, we need to know the true Pareto-optimal front for a problem. Since we are dealing with artificial test problems, the true Pareto-optimal front is not difficult to be obtained. In our experiments we use uniformly spaced Pareto-optimal solutions as the approximation of the true Pareto-optimal front. For all test problems (described in the following sections) used in this study, the true Pareto-optimal fronts are made available online at http://mallba10.lcc.uma.es/wiki/index.php/Problems.

The first metric is the *Convergence metric* γ. It measures the distance the obtained nondominated front Q and the set POF of Pareto-optimal front:

$$\gamma = \frac{\sum_{i=1}^{|Q|} d_i}{|Q|} \tag{13}$$

where d_i is the Euclidean distance (in the objective space, hereinafter) between the solution $i \in Q$ and the nearest member of POF. The lower the γ value, the better the convergence of solutions. A result of $\gamma = 0$ indicates the convergence $Q = POF$; any other value indicates Q deviates from POF.

The second metric is *Diversity metric* Δ. This metric measures the extent of spread achieved among the obtained nondominated front Q. It is desirable to get a set of solutions that spans the entire Pareto-optimal region. Δ is defined as follows:

$$\Delta = \frac{\sum_{i=1}^{k} d_i^e + \sum_{i=1}^{|Q|} |d_i - \overline{d}|}{\sum_{i=1}^{k} d_i^e + |Q|\overline{d}} \tag{14}$$

where d_i^e denotes the Euclidean distance between the i-th coordinate for both extreme points in Q and POF, and d_i measures the Euclidean distance of each point in Q to its closer point in Q. The lower the Δ value, the better the distribution of solutions. A perfect distribution, that is $\Delta = 0$, means that the extreme points of POF have been found and d_i is constant for all i.

Zitzler et al. [36] suggested that the power of unary metrics was restricted. So, a binary performance metric, the coverage of two sets (C value) [4], is selected to overcome the limitations of the unary metrics. This metric is measured to show how the final population of one algorithm dominates the final population of another algorithm. The C value can be calculated as follows:

$$C(X', X'') = \frac{|a'' \in X''; \exists a' \in X' : a' \preceq a''|}{|X''|} \tag{15}$$

where $X', X'' \in X$ are two sets of objective vectors, and $a' \preceq a''$ means that a' *covers* a'' if and only if $a' \prec a''$ or $a' = a''$. Function C maps the ordered pair (X_i, X_j) to the interval $[0, 1]$, where X_i and X_j denote the final populations resulting from algorithm i and j, respectively. The value $C(X_i, X_j) = 1$ implies that all points in X_j are dominated by or equal to points in X_i. The opposite, $C(X_i, X_j) = 0$, represents the situation when none of the points in X_j are covered by the set X_i. Note that both $C(X_i, X_j)$ and $C(X_j, X_i)$ need to be considered independently since they have the distinct meanings.

Therefore, three metrics represent qualitative measures that describe the quality of the final result of selected MOEAs - the average convergence value shows the distance between the obtained nondominated front Q and POF, the average diversity value measures the extent of spread achieved among the obtained nondominated front Q, and the C value compares the domination relationship of a pair of MOEAs. All the values of three performance metrics generated at the final generation are illustrated by box plots to derive the statistical comparison results. In addition, in order to visualize the performance of our proposed $pa\epsilon$-ODEMO, we also present the resulting Pareto fronts obtained by $pa\epsilon$-ODEMO after 5,000 NFFEs on all test problems in Fig. A.1 and A.2 in Appendix.

6.3. *Bi-objective Benchmark Problems*

First, we choose five problems out of six benchmark problems proposed by Zitzler *et al.* [33] and call them ZDT1, ZDT2, ZDT3, ZDT4, and ZDT6, which were frequently used as benchmark problems in the literature [5], [7], [11], [14], and [22]. All problems have two objective functions. None of these problems have any inequality or equality constraints. All objective functions are to be minimized. Since we do not make any changes to the problems, we only briefly describe them in Table 1. More details can be found in [33].

The box plots for the average values of the convergence metric and diversity metric over 50 runs are illustrated in Figs. 1, 3, 5, 7, and 9 for ZDT1-4 and ZDT6, respectively. And the performance measures of $C(X_i, X_j)$ for comparison sets between algorithm i and j are shown in Figs. 2, 4, 6, 8, and 10 for ZDT1-4 and ZDT6, where algorithms 1-6 represent NSGA-II, ϵ-MOEA, DEMO, $pa\epsilon$-MyDE, $pa\epsilon$-ODEMO, and

Table 1
Brief information of the test problems in this study.

Problem	n	k	Pareto front
ZDT1	30	2	high dimensionality, convex
ZDT2	30	2	high dimensionality, non-convex
ZDT3	30	2	high dimensionality, convex, disconnected
ZDT4	10	2	99 local Pareto fronts
ZDT6	10	2	non-convex, non-uniformly spaced
DTLZ1	7	3	hyper-plane, $11^5 - 1$ local Pareto fronts
DTLZ3	12	3	$3^{10} - 1$ local Pareto fronts
DTLZ4	12	3	biased density
DTLZ6	12	3	curve, local Pareto fronts
DTLZ7	22	3	disconnected

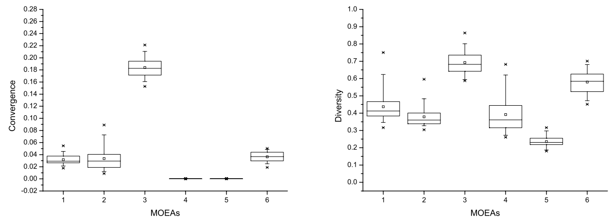
SPEA2, respectively.

From Figs. 1 - 10, we can see that

– For all bi-objective benchmark problems, $pa\epsilon$-ODEMO produces the best performance with respect to the average convergence (γ) value and average diversity (Δ) value than the other five MOEAs, it also provides the highest $C(X_5, X_{1-6})$ values, which means that the solution set resulted from $pa\epsilon$-ODEMO most likely dominate the rest of the solution sets resulted from the other selective MOEAs.

– For ZDT1-3, they have high-dimensionality, but many MOEAs have achieved very good results on this problem in 25,000 NFFEs [22]. However, in this study, all approaches are only run for a maximum of 5,000 NFFEs. Hence, for most of MOEAs, they neither converge to the Pareto-optimal front nor diverse among the solutions obtained. Apparently, comparing the metric values, we can see that DEMO has the lowest performance in terms of all the metric values, while $pa\epsilon$-MyDE and $pa\epsilon$-ODEMO provide competitive results. However, for ZDT2, the diversity of $pa\epsilon$-MyDE is very bad. It has the largest variance among all of selected MOEAs. The reason is that sometimes $pa\epsilon$-MyDE can not generate the initial Pareo front approximation F as soon as possible, so it can not generate the efficient grid and the performance is very bad.

– ZDT4 has 99 local Pareto fronts, which is difficult for many optimizers to find the global Pareto-optimal front. Comparing the metric values, we can see that DEMO has also the lowest performance in terms of all the metric values, while $pa\epsilon$-MyDE and $pa\epsilon$-ODEMO provide competitive results. And $pa\epsilon$-ODEMO is slightly better than $pa\epsilon$-MyDE with respect to all of the three metrics. Because there are many local Pareto fronts, when the NFFEs are small, four MOEAs - NSGA2, ϵ-MOEA, DEMO, and SPEA2 - can not converge towards global Pareto-optimal front in all 50 runs. For $pa\epsilon$-MyDE, it also converges to the local Pareto front 26 out of 50 runs. While for our proposed $pa\epsilon$-ODEMO, it converges to the global Pareto-optimal front in all 50 runs.



(a) (b)

Fig. 1. Box plots of average γ value (a) and average Δ value (b) on test problem ZDT1, where algorithms 1-6 represent NSGA-II, ϵ-MOEA, DEMO, $pa\epsilon$-MyDE, $pa\epsilon$-ODEMO, and SPEA2, respectively, hereinafter.

– For ZDT6, the Pareto-optimal solutions are nonuniformly distributed along the global Pareto-optimal front and the density of the solutions is lowest near the Pareto-optimal front and highest away from the front. From the results in Fig. 9 and Fig. 10, we can see that $pa\epsilon$-ODEMO produces the best performance, followed by $pa\epsilon$-MyDE, NSGA2, SPEA2, DEMO, and ϵ-MOEA. Except $pa\epsilon$-MyDE and $pa\epsilon$-ODEMO, the rest four MOEAs can not converge to the global Pareto-optimal front at all, the average convergence of $\gamma > 1$ for these MOEAs.

Moreover, in order to give a quantitative comparison with the results reported in the literature, we select five state-of-the-art MOEAs - NSGA-II (real code), SPEA2, DEMO (DEMO/parent), MOEO, and ϵ-ODEMO - to make indirect comparisons with $pa\epsilon$-ODEMO. The experimental results of NSGA-II, SPEA2, DEMO, MOEO, and ϵ-ODEMO shown in Tables 2 and 3 come from [5], [14], [22], [14], and [31] respectively. It is worth to point out that for NSGA-II, SPEA2, DEMO, and MOEO, the maximal NFFEs of $Max\_eval = 25,000$. For ϵ-ODEMO, the maximal NFFEs of $Max\_eval = 20,000$. However, for our approach, $pa\epsilon$-ODEMO, the maximal NFFEs of $Max\_eval = 5,000$, which is considerably less than the selective MOEAs.

Table 2 shows the mean and variance of the convergence metric γ obtained using the six MOEAs. It can be observed that although the NFFEs of $pa\epsilon$-ODEMO are very small, it is able to converge better than any other algorithm on four problems

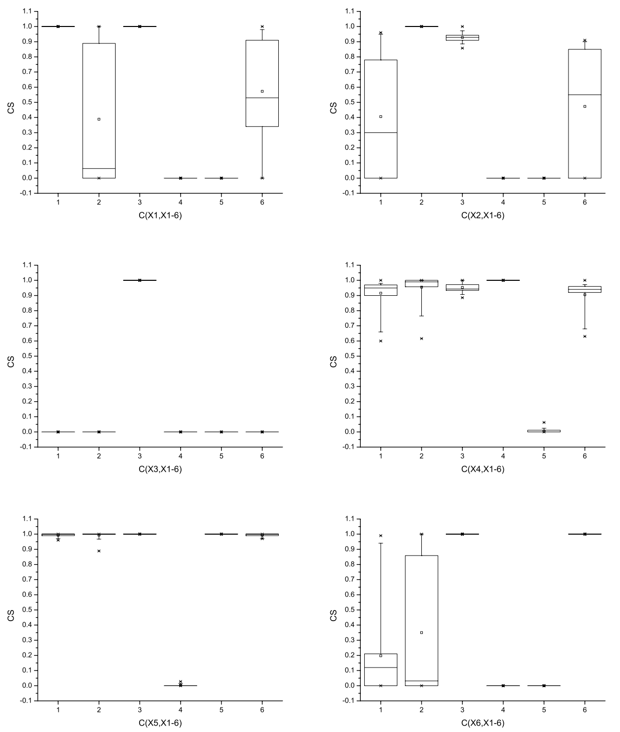


Fig. 2. Box plots based on C measure on test problem ZDT1.

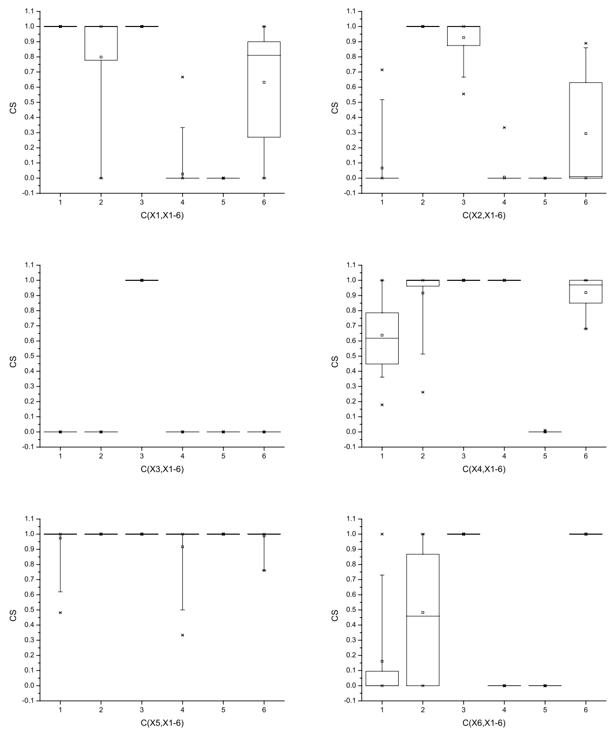
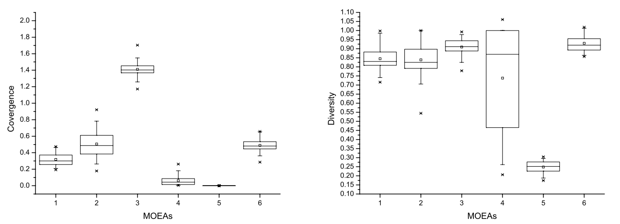
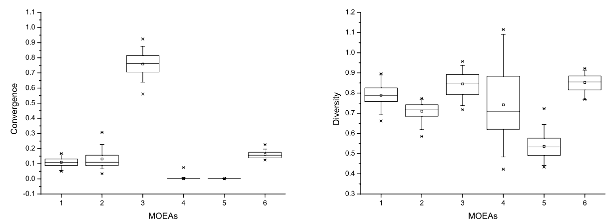


Fig. 4. Box plots based on C measure on test problem ZDT2.



(a) (b)

Fig. 3. Box plots of average γ value (a) and average Δ value (b) on test problem ZDT2.



(a) (b)

Fig. 5. Box plots of average γ value (a) and average Δ value (b) on test problem ZDT3.

(ZDT1, ZDT2, ZDT3, and ZDT4). For ZDT6, $pa\epsilon$-ODEMO provides better performance than NSGA-II and SPEA2, but a little worse than DEMO, MOEO, and ϵ-ODEMO. In all bi-objective problems with $pa\epsilon$-ODEMO, the variance of the convergence metric over 50 runs is very small, it means that $pa\epsilon$-ODEMO is very robust.

The mean and variance of the diversity metric Δ obtained by the six MOEAs are shown in Table 3. From Table 3, we can see that $pa\epsilon$-ODEMO is able to find a better spread of solutions than any other algorithms on all bi-objective problems except ZDT3. This indicates that our approach has ability to find a well-distributed set of non-dominated solutions than many other state-of-the-art MOEAs. In all cases with $pa\epsilon$-ODEMO, the variance of the diversity metric over 50 runs is also very small. For ZDT3, the spread values are the worst among all of the other four problems. This is due to the discontinuity feature of the corresponding Pareto front. $pa\epsilon$-ODEMO is worse than DEMO and ϵ-ODEMO, but better than NSGA-II, SPEA2 and MOEO in terms of the diversity metric for ZDT3.

Once more, it is important to note that our approach is only run for a maximum of 5,000 NFFEs which is considerably less than the selective MOEAs (NSGA-II, SPEA2, DEMO, MOEO, and ϵ-ODEMO). $pa\epsilon$-ODEMO is capable of escaping from the local Pareto optimal front (e.g. ZDT4) and is suitable to deal with those problems with non-uniformly-spaced Pareto front (e.g. ZDT6). Compared with the five state-of-the-art MOEAs

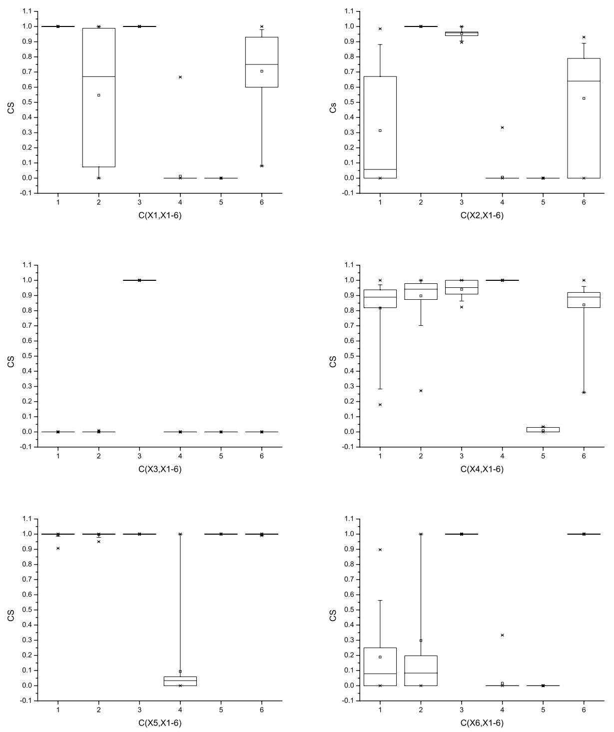
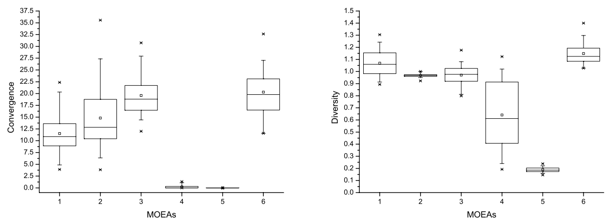


Fig. 6. Box plots based on C measure on test problem ZDT3.



(a) (b)

Fig. 7. Box plots of average γ value (a) and average Δ value (b) on test problem ZDT4.

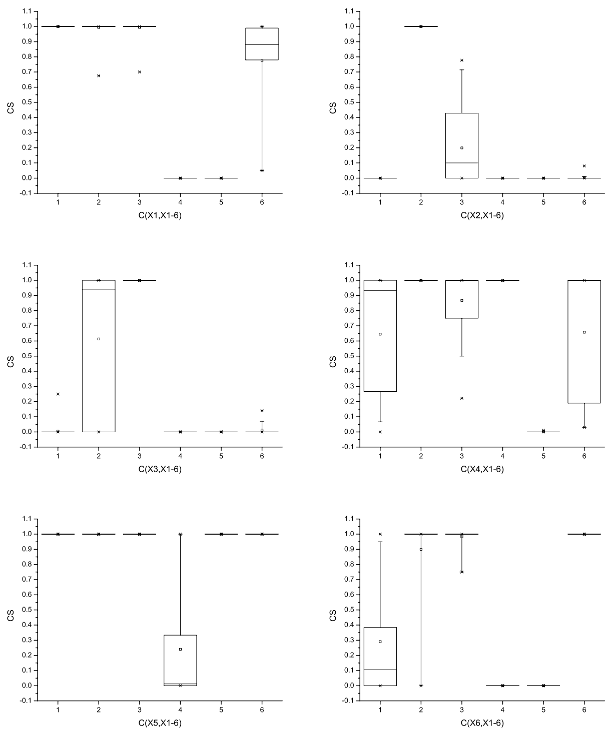
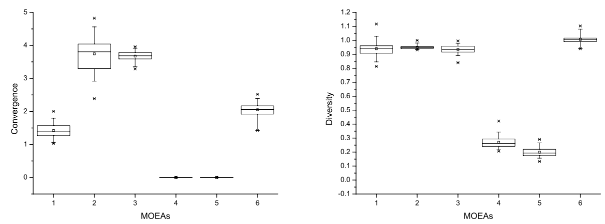


Fig. 8. Box plots based on C measure on test problem ZDT4.



(a) (b)

Fig. 9. Box plots of average γ value (a) and average Δ value (b) on test problem ZDT6.

in terms of the average convergence and diversity metrics in Table 2 and Table 3, it can be concluded that our approach is very competitive. Furthermore, it can ensure both properties of convergence towards the Pareto-optimal set and properties of diversity among the solutions found in small NFFEs.

6.4. *Tri-objective Benchmark Problems*

In order to show the efficacy of $pa\epsilon$-ODEMO in handling problems having more than two objectives, in this study, we choose five problems out of nine benchmark problems proposed by Deb *et al.* [34] and call them DTLZ1, DTLZ3, DTLZ4, DTLZ6, and DTLZ7. All problems have three objective functions. None of these problems have any inequality or equality constraints. All objective functions are to be minimized. Since we do not make any changes to the problems, we only briefly describe them in Table 1. More details can be found in [34].

The box plots for the average values of the convergence metric and diversity metric over 50 runs are illustrated in Figs. 11, 13, 15, 17, and 19 for DTLZ1, DTLZ3, DTLZ4, DTLZ6, and DTLZ7, respectively. And the performance measures of $C(X_i, X_j)$ for comparison sets between algorithm i and j are shown in Figs. 12, 14, 16, 18, and 20 for these tri-objective problems, where algorithms 1-6 represent NSGA-II, ϵ-MOEA, DEMO, $pa\epsilon$-MyDE, $pa\epsilon$-ODEMO, and SPEA2, respectively. Furthermore, for $pa\epsilon$-ODEMO, we illustrate the mean and variance of convergence and diversity metric on the

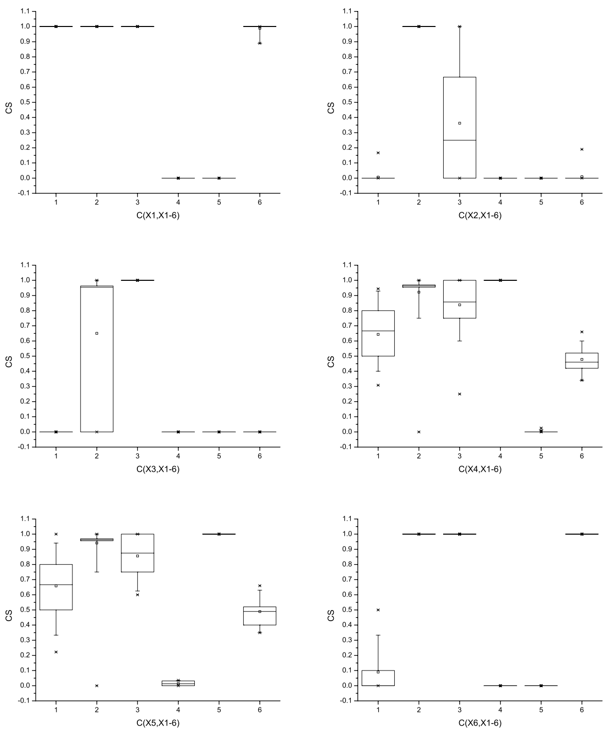


Fig. 10. Box plots based on C measure on test problem ZDT6.

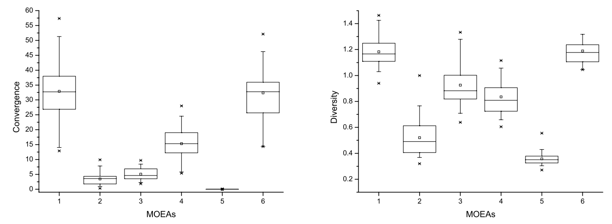
Table 2
Mean (first rows) and variance (second rows) of the convergence metric γ over 50 independent runs. A result with **boldface** indicates better value found.

Algorithm	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6
NSGA-II	0.033482	0.072391	0.114500	0.513053	0.296564
	0.004750	0.031689	0.007940	0.118460	0.013135
SPEA2	0.001448	0.000743	0.003716	0.028492	0.011643
	0.000317	8.33E-05	0.000586	0.047482	0.002397
DEMO	0.001083	0.000755	0.001178	0.001037	0.000629
	0.000113	4.50E-05	5.90E-05	0.000134	4.40E-05
MOEO	0.001277	0.001355	0.004385	0.008145	0.000630
	0.000697	0.000897	0.001910	0.004011	3.26E-05
ϵ-ODEMO	0.000761	0.000764	0.000915	0.000712	**0.000581**
	0.000058	0.000035	0.000050	0.000056	0.000030
$pa\epsilon$-ODEMO	**0.000207**	**0.000198**	**0.000255**	**0.000198**	0.00117
	2.93E-05	1.39E-05	0.000155	1.54E-05	0.000124

five DTLZ problems over 50 independent runs in Table 4.

From Figs. 11 - 20 and Table 4, we can see that

– Be similar to the bi-objective problems, for all tri-objective problems, $pa\epsilon$-ODEMO produces the best performance in terms of the average convergence (γ) value and average diversity (Δ) value than the other five MOEAs, and it also provides the highest $C(X_5, X_{1-6})$ values, which means that



(a) (b)

Fig. 11. Box plots of average γ value (a) and average Δ value (b) on test problem DTLZ1.

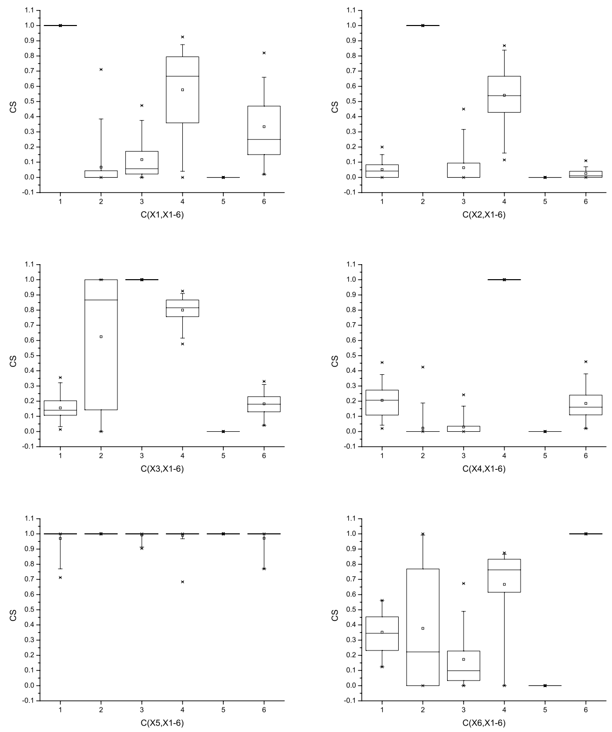
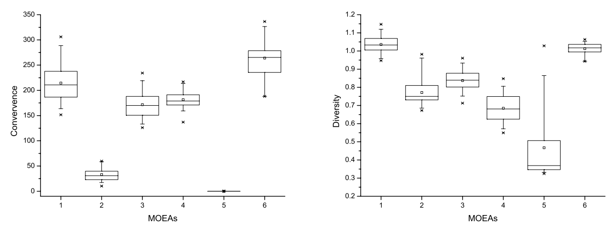


Fig. 12. Box plots based on C measure on test problem DTLZ1.



(a) (b)

Fig. 13. Box plots of average γ value (a) and average Δ value (b) on test problem DTLZ3.

Table 3

Mean (first rows) and variance (second rows) of the diversity metric Δ over 50 independent runs. A result with **boldface** indicates better value found.

Algorithm	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6
NSGA-II	0.390307	0.430776	0.738540	0.702612	0.668025
	0.001876	0.004721	0.019706	0.064648	0.009923
SPEA2	0.472254	0.473808	0.606826	0.705629	0.670549
	0.097072	0.093900	0.191406	0.266162	0.077009
DEMO	0.325237	0.329151	**0.309436**	0.359905	0.442308
	0.030249	0.032408	0.018603	0.037672	0.039255
MOEO	0.327140	0.285062	0.965236	0.275664	0.225468
	0.065343	0.056978	0.046958	0.183704	0.033884
ϵ-ODEMO	0.360154	0.276872	0.534329	0.354847	0.204142
	0.011059	0.007013	0.018301	0.003956	0.005012
$pa\epsilon$-ODEMO	**0.235700**	**0.248440**	0.53250	**0.188600**	**0.199090**
	0.030520	0.036200	0.060420	0.020480	0.032930

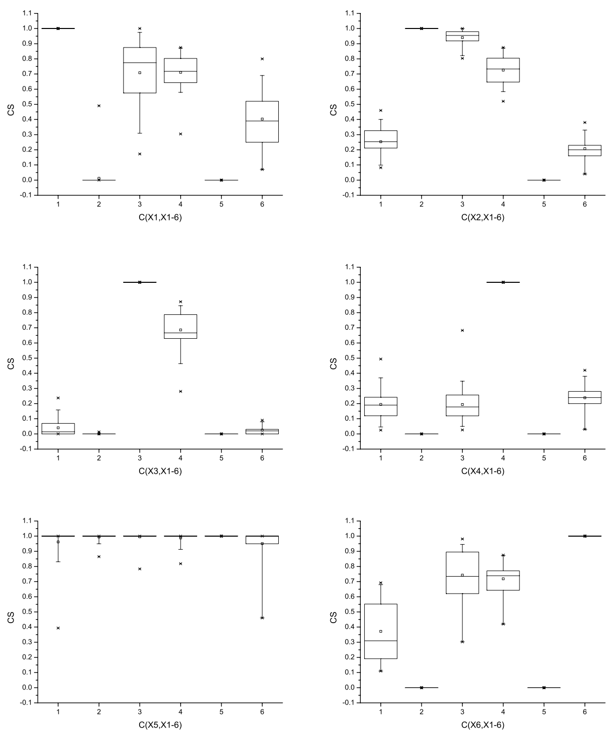
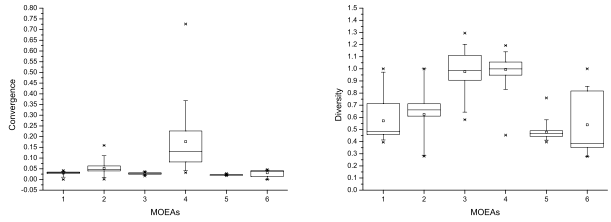


Fig. 14. Box plots based on C measure on test problem DTLZ3.

Table 4

Mean (first rows) and variance (second rows) of the convergence and diversity metric on five DTLZ problems over 50 independent runs.

Metric	DTLZ1	DTLZ3	DTLZ4	DTLZ6	DTLZ7
Convergence	0.00514	0.00931	0.02137	0.00403	0.01434
	0.00145	0.00891	0.00233	0.00034	0.00067
Diversity	0.35746	0.46763	0.47791	0.34507	0.43771
	0.04559	0.18511	0.05847	0.05588	0.04546



(a) (b)

Fig. 15. Box plots of average γ value (a) and average Δ value (b) on test problem DTLZ4.

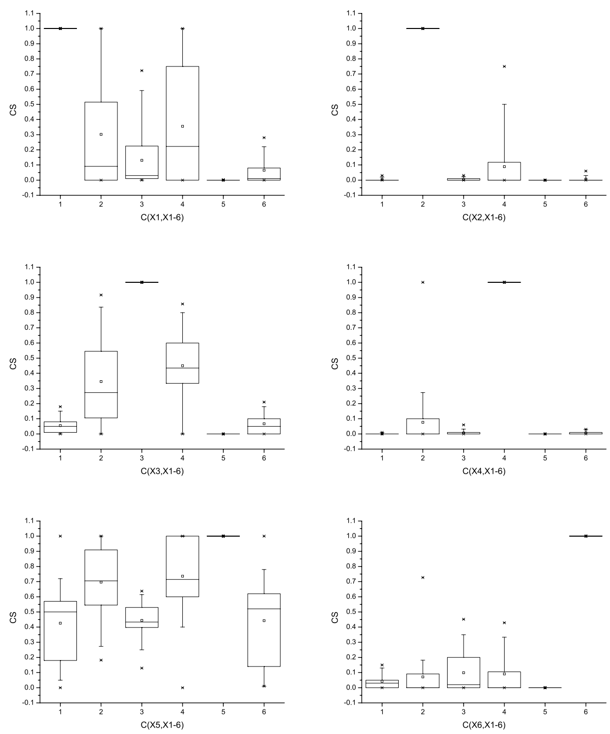
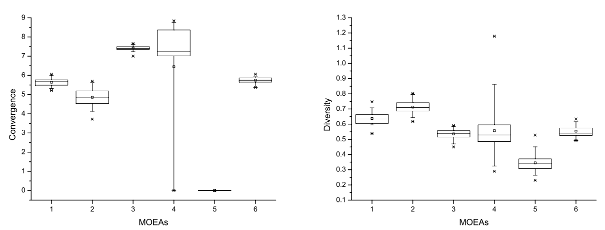


Fig. 16. Box plots based on C measure on test problem DTLZ4.



(a) (b)

Fig. 17. Box plots of average γ value (a) and average Δ value (b) on test problem DTLZ6.

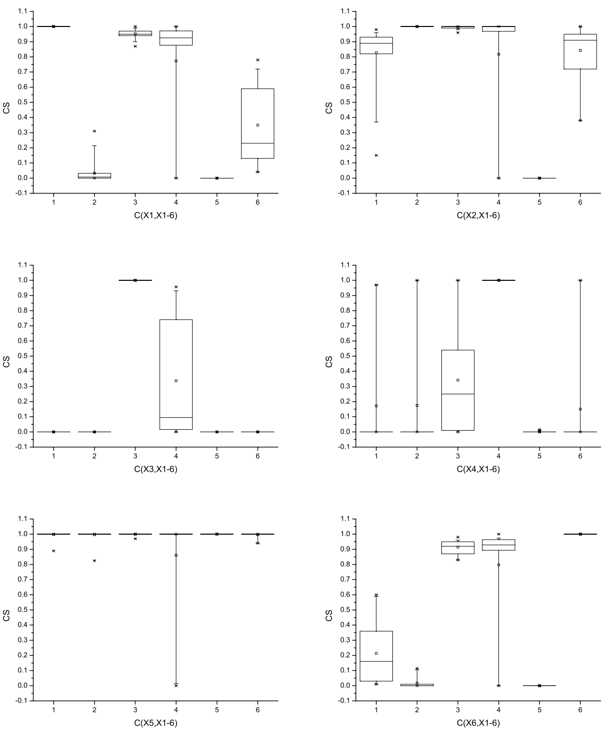


Fig. 18. Box plots based on C measure on test problem DTLZ6.

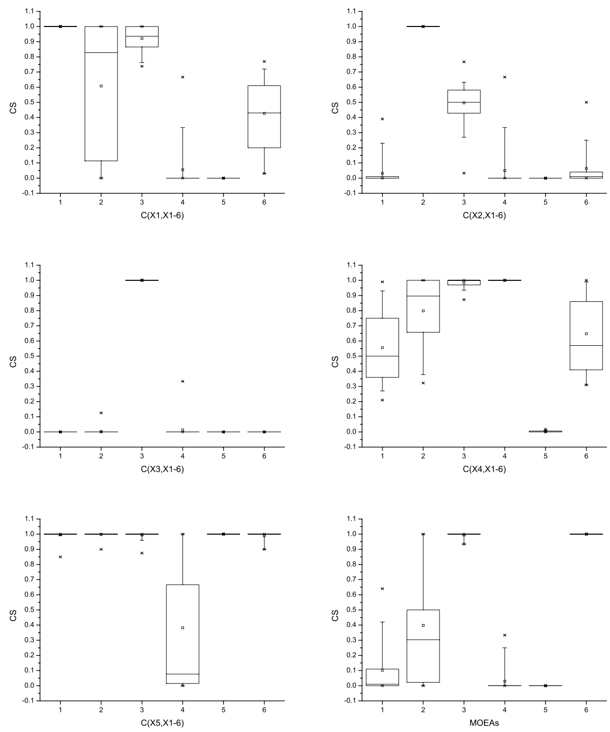
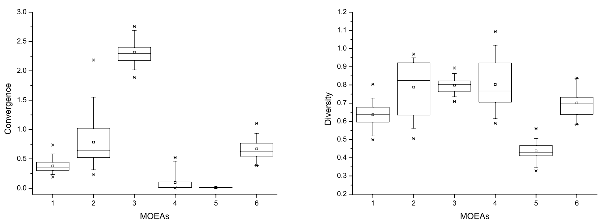


Fig. 20. Box plots based on C measure on test problem DTLZ7.



(a) (b)

Fig. 19. Box plots of average γ value (a) and average Δ value (b) on test problem DTLZ7.

the solution set resulted from $pa\epsilon$-ODEMO most likely dominate the rest of the solution sets resulted from the other selective MOEAs. Moreover, $pa\epsilon$-ODEMO is able to converge towards the global Pareto-optimal front and diverse among the solutions obtained for all tri-objective problems except DTLZ3.

- For DTLZ1, it has $11^5 - 1$ local Pareto fronts which can attract an MOEA. Its difficulty in this problem is to converge to its hyper-plane global Pareto-optimal front. It can be observed from Fig. 11 that only $pa\epsilon$-ODEMO can converge to the hyper-plane in all 50 runs. The other five MOEAs fall into the local Pareto front.
- For DTLZ3, it contains $3^{10} - 1$ local Pareto fronts. All local Pareto fronts are parallel to the global Pareto-optimal front and an MOEA can get stuck at any of these local Pareto fronts, before converging to the global Pareto-optimal front. In this problem, $pa\epsilon$-ODEMO falls into the local Pareto front in 6 out of 50 runs. The other five MOEAs fall into the local Pareto front in all 50 runs.
- DTLZ4 has more dense solutions near the f_3 - f_1 and f_2 - f_1 planes, an MOEA may get attracted to these planes. Because these planes are a part of the global Pareto-optimal front, many MOEAs can obtain a good convergence performance, e.g. NSGA2, ϵ-MOEA, DEMO, and SPEA2 shown in Fig. 15 (a). However, they can not diverse among the obtained solution shown in Fig. 15 (b). For $pa\epsilon$-ODEMO, it falls these plane only one time over the 50 independent runs.
- For DTLZ6, its true Pareto-optimal front is a curve. The g function in the problem makes an MOEA difficult to converge to the true Pareto-optimal front. It is clear from Fig. 17 that the only $pa\epsilon$-ODEMO can converge to the true Pareto-optimal front in all 50 runs. However, $pa\epsilon$-MyDE converges to the true Pareto-optimal front only one run. The other four MOEAs - NSGA-II, ϵ-MOEA, DEMO, and SPEA2 - can not converge to the true Pareto-optimal front at all.
- DTLZ7 has 4 disconnected Pareto-optimal regions in the

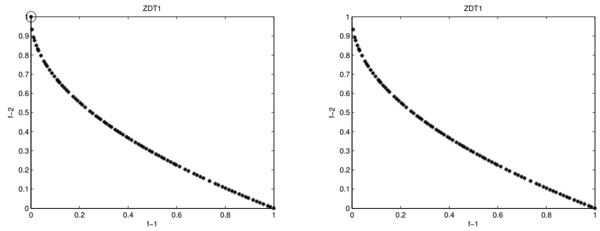
search space. From Fig. 19, the convergence values show that all MOEAs can converge some of the disconnected Pareto-optimal regions except DEMO. But only $pa\epsilon$-ODEMO can locate all of the disconnected Pareto-optimal regions in all 50 runs in terms of the diversity values.

6.5. *Effect of the Orthogonal Population Initialization*

In this section, we perform an additional experiment to show the effect of the orthogonal population initialization. Four $pa\epsilon$-dominance based MOEAs are selected to test the performance: (i) our proposed $pa\epsilon$-ODEMO, (ii) $pa\epsilon$-DEMO, which is similar to $pa\epsilon$-ODEMO except using the random population initialization, (iii) $pa\epsilon$-MyDE proposed in [32], and (iv) $pa\epsilon$-OMyDE, which is similar to $pa\epsilon$-MyDE except adopting the orthogonal population initialization. The parameter settings for the four approaches are used as described in Section 6.1. All approaches are only run for a maximum of 5,000 NFFEs on the test problems. Due to the tight space restrictions however, we only show the statistic results (mean, variance, and 95% confidence interval) of the five ZDT problems over 50 independent runs. Table 5 shows the statistic results of convergence metric obtained by the four MOEAs. The results of diversity metric are shown in Table 6.

From Table 5 and 6, it can be seen that 1) $pa\epsilon$-ODEMO can obtain significantly better performance than $pa\epsilon$-DEMO in terms of the convergence and diversity metrics. There is no overlap of the confidence intervals of both metrics for $pa\epsilon$-ODEMO and $pa\epsilon$-DEMO for the five ZDT problems. Especially, $pa\epsilon$-DEMO locates the local Pareto front many times for ZDT4 and ZDT6. 2) $pa\epsilon$-OMyDE gets slightly better performance than $pa\epsilon$-MyDE in terms of the convergence and diversity metrics for the five ZDT problems. The reason for this happening is mostly like to the orthogonal population initialization, which is able to help to improve search space exploitation and to save a considerable number of solution evaluations for further investigation at later generations. Furthermore, it can help to generate the initial Pareto front approximation with the higher number of efficient points as soon as possible.

It is interesting to note that $pa\epsilon$-ODEMO obtains better performance than $pa\epsilon$-OMyDE in terms of both convergence and diversity metrics. There is no overlap of the confidence intervals of both metrics for the $pa\epsilon$-ODEMO and $pa\epsilon$-OMyDE for the four out of five ZDT problems, except for ZDT6. The reason



(a) (b)

Fig. 21. Influence of storage of extreme points on test problem ZDT1.

for this happening is most likely that the hybrid selection mechanism used in $pa\epsilon$-ODEMO can guide the search efficiently. Moreover, the extreme points retained in the final archive can also slightly improve the performance.

6.6. *Influence of Storage of Extreme Points*

In order to show the influence of storage of extreme points, in this study, we select ZDT1 as an example. Fig. 21 illustrates the results of $pa\epsilon$-ODEMO with Fig. 21 (a) and without Fig. 21 (b) storing the extreme points. One of the extreme points marked by a circle is stored in Fig. 21 (a). The convergence value and diversity value of Fig. 21 (a) are 0.000199 and 0.206285, respectively. For Fig. 21 (b), the convergence value and diversity value are 0.000202 and 0.214519, respectively. Apparently, from the qualitative and quantitative comparison results, we can see that the storage of extreme points is capable of improving the performance in terms of the obtained Pareto front, convergence value and diversity value. Moreover, the obtained Pareto fronts of all test problems shown in Fig. A.1 and A.2 in Appendix show that the extreme points are stored in the final archive on most of the problems.

6.7. *Influence of Selection Parameter λ*

In this study, we do not make any serious attempt to find the best parameter setting for our proposed $pa\epsilon$-ODEMO. But in this section, we perform an additional experiment to show the effect of different λ settings on the performance of $pa\epsilon$-ODEMO. Due to the tight space restrictions however, we only select ZDT1 as an example.

We keep all other parameters as before, but using 11 different λ values, i.e. $\lambda = 0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0$. Especially, when $\lambda = 0$, it means that starting from evolution one of the archive solutions is selected to generate the offspring.

Table 5

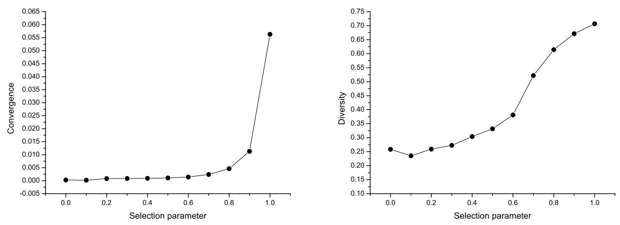
Mean, variance, and 95% confidence interval of convergence metric on five ZDT problems over 50 independent runs.

Algorithm	ZDT1			ZDT2			ZDT3			ZDT4			ZDT6		
	mean	variance	CI 95%	mean	variance	CI 95%	mean	variance	CI 95%	mean	variance	CI 95%	mean	variance	CI 95%
$pa\epsilon$-ODEMO	2.1e-4	2.9e-5	[2.0e-4,2.2e-4]	0.0002	1.4e-5	[1.9e-4,2.0e-4]	0.0003	0.0002	[2.0e-4,3.1e-4]	2.0e-4	1.5e-5	[1.9e-4,2.0e-4]	0.0012	0.0001	[0.0011,0.0012]
$pa\epsilon$-DEMO	0.0034	0.0030	[2.3e-3,4.5e-3]	0.0544	0.1262	[7.3e-3,1.0e-1]	0.0028	0.0061	[4.8e-4,5.0e-3]	11.892	5.4928	[9.8e+0,1.3e+1]	4.0362	2.2005	[3.2155,4.8569]
$pa\epsilon$-OMyDE	3.1e-4	7.0e-5	[2.8e-4,3.4e-4]	0.0121	0.0160	[6.2e-3,1.8e-2]	0.0012	9.2e-4	[8.2e-4,1.5e-3]	0.1001	0.2162	[2.0e-2,1.8e-1]	0.0012	1.1e-4	[0.0011,0.0012]
$pa\epsilon$-MyDE	5.2e-4	2.7e-5	[5.1e-4,5.3e-4]	0.0621	0.0616	[3.9e-2,8.5e-2]	0.0027	0.0010	[0.002,0.003]	0.2013	0.3182	[8.3e-2,3.2e-1]	0.0013	1.5e-4	[0.0012,0.0013]

Table 6

Mean, variance, and 95% confidence interval of diversity metric on five ZDT problems over 50 independent runs.

Algorithm	ZDT1			ZDT2			ZDT3			ZDT4			ZDT6		
	mean	variance	CI 95%	mean	variance	CI 95%	mean	variance	CI 95%	mean	variance	CI 95%	mean	variance	CI 95%
$pa\epsilon$-ODEMO	0.2357	0.0305	[2.2e-1,2.5e-1]	0.2484	0.0362	[2.4e-1,2.6e-1]	0.5362	0.0604	[0.5137,0.5588]	0.1886	0.0205	[1.8e-1,2.0e-1]	0.1991	0.0329	[0.1868,0.2114]
$pa\epsilon$-DEMO	0.3796	0.0671	[3.6e-1,4.1e-1]	0.8095	0.0350	[8.0e-1,8.2e-1]	0.6124	0.0629	[0.5890,0.6359]	0.9136	0.0340	[9.0e-1,9.3e-1]	0.9694	0.0289	[0.9586,0.9802]
$pa\epsilon$-OMyDE	0.2791	0.0374	[2.7e-1,2.9e-1]	0.6326	0.3069	[5.2e-1,7.5e-1]	0.7418	0.1783	[0.6753,0.8083]	0.5206	0.2720	[4.2e-1,6.2e-1]	0.2011	0.0581	[0.1795,0.2228]
$pa\epsilon$-MyDE	0.3920	0.1095	[3.5e-1,4.3e-1]	0.7382	0.2919	[6.3e-1,8.5e-1]	0.7723	0.1956	[0.6993,0.8452]	0.6415	0.2761	[5.4e-1,7.4e-1]	0.2700	0.0414	[0.2545,0.2854]



(a) (b)

Fig. 22. Influence of different selection parameter settings on the performance of test problem ZDT1.

In this case, the inefficient archive member may mislead the search. However, $\lambda = 1.0$ indicates that any archive solution is not selected to generate the offspring at all. It can be observed from Fig. 22 that different λ provides different performance in terms of convergence metric and diversity metric. The smaller value of λ can produce better performance on ZDT1. If the archive solutions do not take part in the generating process (i.e. $\lambda = 1.0$), the performance is very bad with respect to convergence and diversity of solutions.

It is worth to note that this experiment does not show how to set the λ value to produce the best performance for different MOPs. However, we can conclude that properly allowing the archive solutions to take part in the evolutionary process can guarantee to be the best performance produced.

6.8. *Advanced Features of $pa\epsilon$-ODEMO*

From the above results, we can conclude that our approach has the following advanced features:

- It is very efficient in terms of the small NFFEs and values of three metrics. It can find the *uniformly distributed*, *near-complete*, and *near-optimal* Pareto fronts. It is a very competitive MOEA compared with five state-of-the-art MOEAs.
- The external archive provides the elitist mechanism for our approach.
- Our approach produces good performance in terms of the average convergence, diversity and C metric.
- Our approach is able to handle those problems with many local Pareot fronts (ZDT4, DTLZ1 and DTLZ3), non-uniformly-spaced Pareto-optimal front (ZDT6), disconnected Pareto-optimal front (ZDT3 and DTLZ7), high-dimensional decision space (ZDT1-3 and DTLZ7) or high-dimensional objective space problem (DTLZs).
- Our approach is more robust and efficient than $pa\epsilon$-MyDE, another MOEA using $pa\epsilon$-dominance. The reason is that in our proposed $pa\epsilon$-ODEMO it adopts the orthogonal design method to generate the initial archive and evolutionary population, thus our approach can evenly scan the feasible solution space once to locate good points for further exploration in subsequent iterations. Moreover, our approach can generate the initial Pareto front approximation with the higher number of efficient points fast.

– Incorporated with our proposed hybrid selection mechanism, our approach can use the archive solutions efficiently and make the algorithm converge faster.
– With the strategy of storage of extreme points, our approach can store the extreme points in the final archive.

7. Conclusion

In this paper, an improved multiobjective DE algorithm, $pa\epsilon$-ODEMO, is presented. $pa\epsilon$-ODEMO adopts the orthogonal design method with quantization technique to generate the initial archive and evolutionary population. In this manner, the initial population can uniformly scatter over the search space, so that the algorithm can evenly generate an efficient initial Pareto front approximation required to generate the first grid in $pa\epsilon$-dominance method. A new relaxed form of Pareto dominance, $pa\epsilon$-dominance, is used to ensure both properties of convergence towards the Pareto-optimal set and properties of diversity among the solutions found. Moreover, the $pa\epsilon$-dominance can remedy some limitations of the original ϵ-dominance. In addition, we propose a new hybrid selection mechanism to allow the archive solutions to take part in the evolutionary process in order to guarantee to be the best performance produced. A simple strategy to store the extreme points is also proposed, which can store the ϵ-dominated extreme points in the final archive and improve the diversity of solutions obtained.

In order to validate the performance of our proposed $pa\epsilon$-ODEMO, we select ten unconstrained real-valued artificial MOPs (five bi-objective problems and five tri-objective problems), which have been well designed to exploit various complications in finding different true Pareto-optimal fronts. Experimental results show that our approach produces good performance in terms of the quality of the approximation of the Pareto-optimal front and the considerable reduction of NFFEs when solving these problems. Compared with five state-of-the-art MOEAs - NSGA-II, ϵ-MOEA, DEMO, $pa\epsilon$-MyDE, and SPEA2 - the results show that $pa\epsilon$-ODEMO produces statistically competitive results in finding a *uniformly distributed*, *near-complete*, and *near-optimal* Pareto fronts in the test problems.

Our future work consists on investigating the effect of different parameter settings on the performance of our approach. In this work, we only considered the unconstrained real-valued artificial problems (ZDTs and DTLZs), which have some limitations, such as non-deceptive, non-separable, and so on [35], another future direction is using the proposed $pa\epsilon$-ODEMO to solve more complex MOPs, the constrained MOPs and dynamic MOPs.

Appendix A. Appendix

In order to visualize the performance of $pa\epsilon$-ODEMO, the nondominated solutions of the final archive obtained by $pa\epsilon$-ODEMO on all test problems are shown in Fig. A.1 and A.2. The presented fronts are the outcome of a single typical run after 5,000 NFFEs. From Fig. A.1 and A.2, we can see that (i) $pa\epsilon$-ODEMO is capable of converging towards the true Pareto-optimal front on all test problems; (ii) it can find a well-distributed and near-complete set of nondominated solutions on all test problems except ZDT3; and (iii) the extreme points are saved in the final archive on most of test problems.

Acknowledgment

The authors sincerely thank Dr. A.G. Hernández-Díaz for providing us with the $pa\epsilon$-MyDE code and his suggestions on our work. This work was supported by the China Scholarship Council under Grant No. 2008641008, the Humanities Base Project of Hubei Province under grant No. 2004B0011 and the Natural Science Foundation of Hubei Province under grant No. 2003ABA043. Because this work is the extension of our previous work [31] presented in EMO-07, the authors would also like to thank the three anonymous reviewers in EMO-07 for their constructive advice.

References

[1] C.A.C. Coello, Evolutionary multi-objective optimization: A historical view of the Field, IEEE Computational Intelligence Magazine 1(1) (2006) 28 - 36.
[2] D. Nam, and C.H. Park, Multiobjective simulated annealing: A comparative study to evolutionary algorithms, International Journal of Fuzzy Systems 2(2) (2000) 87 - 97.
[3] A. Baykasoglu, Goal programming using multiple objective tabu search, Journal of the Operational Research Society 52(12) (2001) 1359 - 1369.
[4] E. Zitzler and L. Thiele, Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach, IEEE Transactions on Evolutionary Computation 3(4) (1999) 257 - 271.
[5] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA - II, IEEE Transactions on Evolutionary Computation 6(2) (2002) 182 - 197.

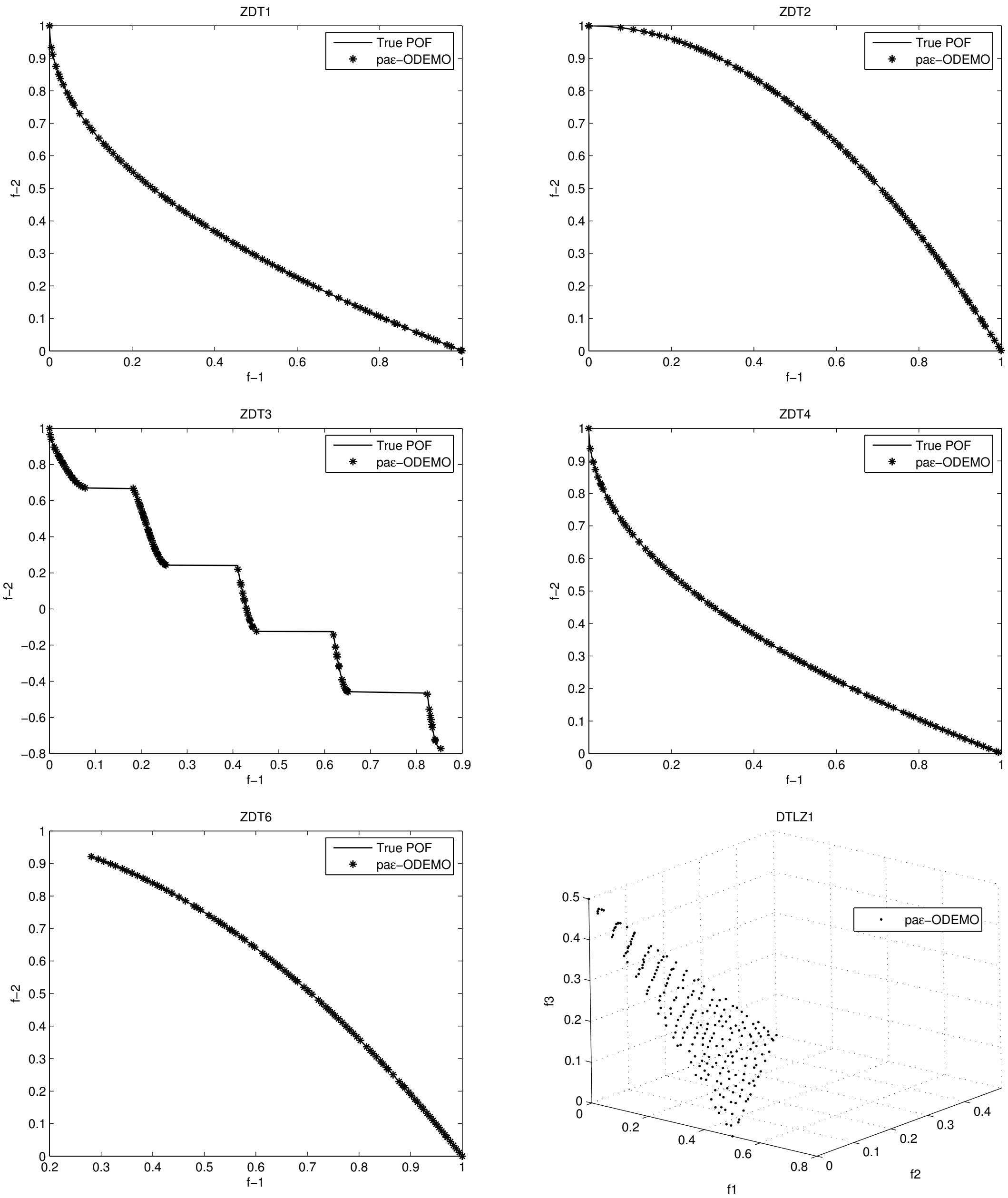


Fig. A.1. Nondominated solutions of the final archive obtained by $pa\epsilon$-ODEMO on some test problems. The presented fronts are the outcome of a typical run.

[6] N. Srinivas, K. Deb, Multi-objective function optimization using nondominated sorting genetic algorithms, Evolutionary Computation 2 (1995) 221 - 248.

[7] E. Zitzler, M. Laumanns, L. Thiele, SPEA2: Improving the strength Pareto evolutionary algorithm, Technical Report 103, Computer Engineering and Networks Laboratory, (2001).

[8] J.D. Knowles, D.W. Corne, Approximating the nondominated front using the Pareto archived evolution strategy, Evolutionary Computation 8 (2000) 149 - 172.

[9] H. Lu, G.G. Yen, Rank-density-based multiobjective genetic algorithm and benchmark test function study, IEEE Transactions on Evolutionary Computation 7(4) (2003) 325 - 343.

[10] H. Eskandari, C.D. Geiger, and G.B. Lamont, FastPGA: A dynamic population sizing approach for solving expensive multiobjective optimization problems, Fourth International Conference on Evolutionary Multi-Criterion Optimization (EMO-07), LNCS 4403 (2007) 141 - 155.

[11] R.P. Beausoleil, "MOSS" multiobjective scatter search applied to non-linear multiple criteria optimization, European Journal of Operational Research 169 (2006) 426 - 449.

[12] T. Hanne, A multiobjective evolutionary algorithm for approximating the efficient set, European Journal of Operational Research 176 (2007) 1723 - 1734.

[13] S. Elaoud, T. Loukil, J. Teghem, The Pareto fitness genetic algorithm: Test function study, European Journal of Operational Research 177 (2007) 1703 - 1719.

[14] M.R. Chen, Y.Z. Lu, A novel elitist multiobjective optimization

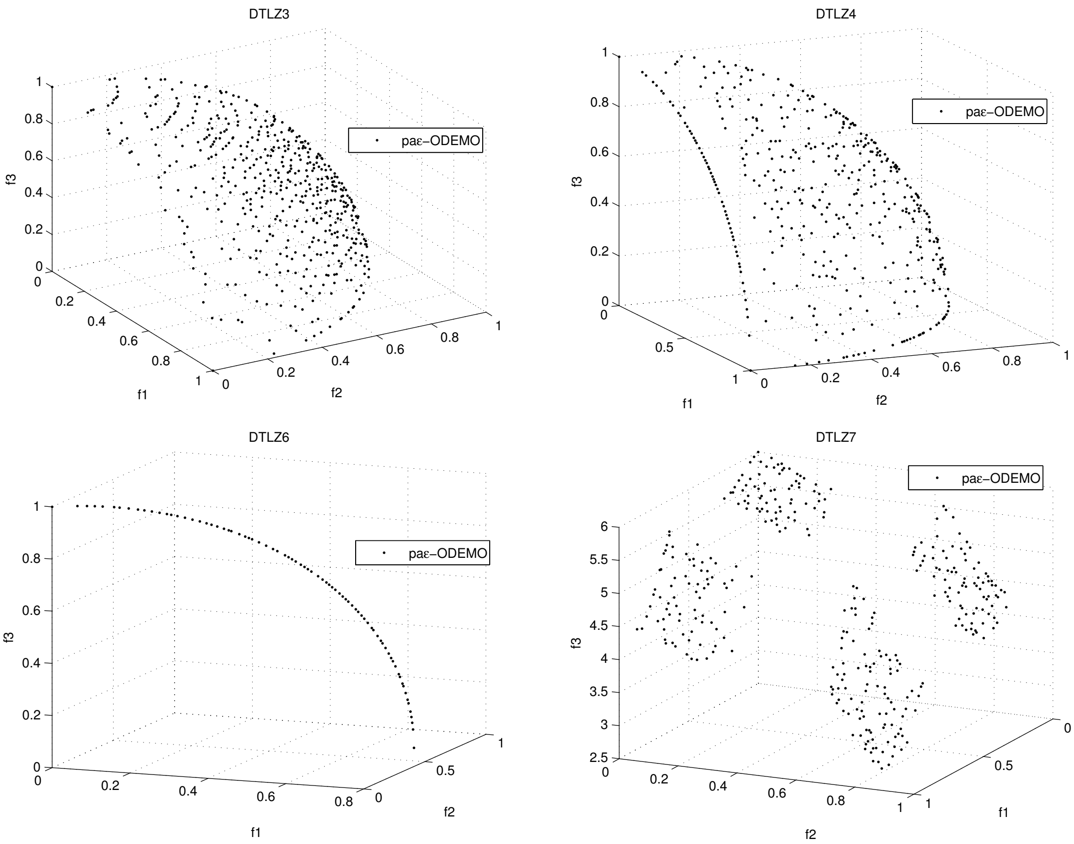


Fig. A.2. Nondominated solutions of the final archive obtained by $pa\epsilon$-ODEMO on some test problems. The presented fronts are the outcome of a typical run.

algorithm: multiobjective extremal optimization, European Journal of Operational Research 188 (2008) 637 - 651.

[15] N. Beume, B. Naujoks, M. Emmerich, SMS-EMOA: Multiobjective selection based on dominated hypervolume, European Journal of Operational Research 181 (2007) 1653 - 1669.

[16] R. Storn, K. Price, Differential evolution–A simple and efficient heuristic for global optimization over continuous spaces, Journal of Global Optimization 11 (1997) 341 - 359.

[17] R. Storn, K. Price, Home page of differential evolution. Available online at http://www.ICSI.Berkeley.edu/˜storn/code.html. (2003).

[18] T. Gong, and A.L. Tuson, Differential evolution for binary encoding, Soft Computing in Industrial Applications, ASC 39 (2007) 251C262.

[19] H.A. Abbass, R. Sarker, C. Newton, PDE: A Pareto-frontier differential evolution approach for multi-objective optimization Problems, Proceedings of the 2001 IEEE Congress on Evolutionary Computation (2) (2001) 971 - 978.

[20] N.K. Madavan, Multiobjective optimization using a Pareto differential evolution approach. Proceedings of the 2002 IEEE Congress on Evolutionary Computation (2) (2002) 1145 - 1150.

[21] F. Xue, A.C. Sanderson, R.J. Graves, Pareto-based multi-objective differential evolution, Proceedings of the 2003 IEEE Congress on Evolutionary Computation (2) (2003) 862 - 869.

[22] T. Robič, B. Filipič, DEMO: Differential evolution for multiobjective optimization, Third International Conference on Evolutionary Multi-Criterion Optimization (EMO-05) (2005) 520 - 533.

[23] S. Kukkonen and J. Lampinen, GDE3: The third evolution step of generalized differential evolution, Proceedings of the 2005 IEEE Congress on Evolutionary Computation (1) (2005) 443 - 450.

[24] L.V. Santana-Quintero, and C.A.C. Coello, An algorithm based on differential evolution for multi-objective problems, International Journal of Computational Intelligence Research 1(2) (2005) 151 - 169.

[25] K.T. Fang, and C.X. Ma, Orthogonal and uniform design (in Chinese). Science Press (2001).

[26] Y.W. Leung, and Y. Wang, An orthogonal genetic algorithm with quantization for global numerical optimization. IEEE Transactions on Evolutionary Computation 5(1) (2001) 41 - 53.

[27] S.Y. Zeng, L.S. Kang, L.X. Ding, An orthogonal multiobjective evolutionary algorithm for multi-objective optimization problems with constraints. Evolutionary Computation 12 (2004) 77 - 98.

[28] K. Deb, M. Mohan, S. Mishra, Towards a quick computation of well-spread Pareto-optimal solutions, Second International Conference on Evolutionary Multi-Criterion Optimization (EMO-03) (2003) 222 - 236.

[29] K. Deb, M. Mohan, S. Mishra, Evaluating the ϵ-domination based multi-objective evolutionary algorithm for a quick computation of Pareto-optimal solutions, Evolutionary Computation 13(4) (2005) 501 - 525.

[30] M. Laumanns, L. Thiele, K. Deb, E. Zitzler, Combining convergence and diversity in evolutionary multi-objective optimization, Evolutionary Computation 10(3) (2002) 263 - 282.

[31] Z.H. Cai, W.Y. Gong, Y.Q. Huang, A novel differential evolution algorithm based on ϵ-domination and orthogonal design method

for multiobjective optimization, Fourth International Conference on Evolutionary Multi-Criterion Optimization (EMO-07), LNCS 4403 (2007) 286 - 301.

[32] A.G. Hernández-Díaz, L.V. Santana-Quintero, C.A.C. Coello, *et al*, Pareto-adaptive ϵ-dominance, Evolutionary Computation 15(4) (2007) 493 - 517.

[33] E. Zitzler, K. Deb, L. Thiele, Comparison of multiobjective evolutionary algorithms: Empirical results. Evolutionary Computation 8 (2000) 173 - 195.

[34] K. Deb, L. Thiele, M. Laumanns, E. Zitzler, Scalable test problems for evolutionary multiobjective optimization. Evolutionary Multiobjective Optimization: Theoretical Advances and Applications (2005) 105 - 145.

[35] S. Huband, P. Hingston, L. Barone, et al. A Review of Multiobjective Test Problems and a Scalable Test Problem Toolkit, IEEE Transactions on Evolutionary Computation 10(5) (2006) 477 - 506.

[36] E. Zitzler, L. Thiele, M. Laumanns, C.M. Fonseca, *et al.* Performance assesement of multiobjective optimizer: An analysis and review. IEEE Transactions on Evolutionary Computation 7(2) (2003) 117 - 132.