A Novel Differential Evolution Algorithm based on ϵ-domination and Orthogonal Design Method for Multiobjective Optimization

Zhihua Cai[1,2], Wenyin Gong[1], and Yongqin Huang[1]

[1] School of Computer Science
China University of Geosciences, Wuhan 430074, P. R. China
zhcai@cug.edu.cn, cug11100304@yahoo.com.cn
[2] Dept. of computer science
The University of Western Ontario,London,Ontario,N6A 5B7,Canada

Abstract. To find solutions as close to the Pareto front as possible, and to make them as diverse as possible in the obtained non-dominated front is a challenging task for any multiobjective optimization algorithm.ϵ-dominance is a concept which can make genetic algorithm obtain a good distribution of Pareto-optimal solutions and has low computational time complexity,and the orthogonal design method can generate an initial population of points that are scattered uniformly over the feasible solution space.In this paper, combining ϵ-dominance and orthogonal design method, we propose a novel Differential Evolution (DE) algorithm for multiobjective optimization .Experiments on a number of two- and three-objective test problems of diverse complexities show that our approach is able to obtain a good distribution with a small computational time in all cases. Compared with several other state-of-the-art evolutionary algorithms, it achieves not only comparable results in terms of convergence and diversity metrics, but also a considerable reduction of the computational effort.

1 Introduction

Evolutionary Algorithms (EAs) (including genetic algorithms, evolution strategies, evolutionary programming, and genetic programming) are heuristics that have been successfully applied in a wide set of areas. In 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. Simply combining the different associated objective functions in a linear way is usually unsatisfactory. Instead, one is interested in a so-called Pareto optimal set of solutions, i.e., any solution that cannot be improved with respect to one objective without worsening the situation with respect to the other objectives. Consequently, there are two goals in multiobjective optimization: (i) to find solutions as close to the Pareto front as possible, and (ii) to find solutions as diverse as possible in the obtained non-dominated front. Satisfying the two goals is a challenging task for any multiobjective optimization algorithm. Special strategies are therefore needed to deal with such multiobjective optimization problems. Since EAs work on populations of candidate solutions, they represent a promising basic framework for multiobjective optimization.

In the last few years, many variants and extensions of classical EAs have been developed for Multiobjective Optimization Problems (MOPs). Such as Nondominated Sorting GA (NSGA-II) [1], Strength Pareto EA (SPEA2) [2], Vector Evaluated GA (VEGA) [3], Hajela and Lins GA (HLGA) [4], Pareto-based Ranking Procedure(FFGA) [5], Niched Pareto GA (NPGA) [6], Pareto Archived Evolution Strategy (PAES) [7], and so on. Among these, the NSGA-II by Deb *et al*. [1] and SPEA2 by Zitzler *et al*. [2] are the most popular approaches.

Differential evolution (DE) [8] is a novel evolutionary algorithm for faster optimization, which mutation operator is based on the distribution of solutions in the population. And DE has won the third place at the first International Contest on Evolutionary Computation on a real-valued function test-suite. Unlike Genetic Algorithm (GA) that uses binary coding to represent problem parameters, 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. Among the DE's advantages are its simple structure, ease of use, speed and robustness. Price & Storn [8] gave the working principle of DE with single strategy. Later on, they suggested ten different strategies of DE [9]. It has been successfully used in solving single-objective optimization problems [10]. Hence, several researchers have tried to extend it to handle MOPs. Such as Pareto DE (PDE) [11,12], Pareto DE Approach (PDEA) [13], Multiobjective DE (MODE) [14], and DE for Multiobjective Optimization (DEMO) [15].

Combing orthogonal array (OA) and factor analysis (such as the statistical optimal method),Orthogonal design method [16] is developed to sample a small and representative set for all possible combinations to obtain good combinations. Recently, some researchers applied the orthogonal design method incorporated with EAs to solve optimization problems. Leung and Wang [17] incorporated orthogonal design in genetic algorithm for numerical optimization problems and found such method was more robust and statistically sound than the classical GAs. OMOEA [18] and OMOEA-II [19] presented by Sangyou Zeng *et al*. adopted the orthogonal design method to solve the MOPs. Numerical results demonstrated the efficiency of the two tools.

ϵ-MOEA [20] is a steady-state Multiobjective EA (MOEA) based on the ϵ-dominance concept introduced in [21]. Also, it incorporated efficient parent and archive update strategies to obtain a good distribution of Pareto-optimal solutions within less computational time. The ϵ-dominance does not allow two solutions with a difference ϵ_i in the i-th objective to be nondominated to each other, thereby allowing a good diversity to be maintained in the population. Besides, the method is quite pragmatic, because it allows the user to choose a suitable ϵ_i depending on the desired resolution in the i-th objective [20].

Inspired by the ideas from OGA/Q [17] and ϵ-MOEA [20], in this paper, we propose an extension of DE algorithm based on the ϵ-dominance concept and orthogonal design method. Our proposed DE algorithm is named ϵ-ODEMO. Our algorithm has three novelties. Firstly, the proposed approach adopts orthogonal design method with quantization technique to generate an initial population of points. And then, it uses the DE/rand/1/exp strategy to produce new candidate solutions. Thirdly, ϵ-dominance concept and efficient parent and archive update strategies introduced in [20] are used to update the archive and population. To evaluate the efficiency of the proposed ϵ-ODEMO, we test it on a number of two- and three-objective problems.

The rest of this paper is organized as follows. In Section 2, we briefly introduce the background of ϵ-MOEA. In Section 3, we describe the function optimization problems in conventional DE. A detailed description of the proposed ϵ-ODEMO algorithm is provided in Section 4. In Section 5, we test our algorithm through a number of two- and three-objective problems. This is followed by results and discussions of the optimization experiments for ϵ-ODEMO in Section 6. The last section, Section 7, is devoted to conclusions and future studies.

2 Background of ϵ-MOEA

ϵ-MOEA [20] is a new approach for MOPs, which is a steady-state MOEA based on the ϵ-dominance concept [21]. In ϵ-MOEA, the search space is divided into a number of grids (or hyper-boxes) and diversity is maintained by ensuring that a grid or hyper-box can be occupied by only one solution. There are two co-evolving populations: (i) an EA population $P(t)$ and (ii) an archive population $E(t)$ (where t is the iteration counter). The archive population stores the nondominated solutions and is updated iteratively with the ϵ-dominance concept. And the EA population is updated iteratively with the usual domination. ϵ-MOEA is described as follows:

Algorithm 1 ϵ-MOEA algorithm

```
Generate an initial population P(0) uniform randomly
Create the archive E(0) with the ε-nondominated solutions of P(0)
while the halting criterion is not satisfied do
  Choose one solution each from P(t) and E(t) for mating
  Use crossover and mutation to produce λ offspring solutions
  Compare each of these offspring solutions with the archive and the EA population to update
  them respectively
end while
```

As mentioned above, The archive population is updated by the offspring solutions iteratively with the ϵ-dominance concept. In ϵ-MOEA, every solution in the archive is assigned an identification array ($\boldsymbol{B}$) which can be calculated by:

$$B_j(\boldsymbol{f}) = \begin{cases} \lfloor (f_j - f_j^{min})/\epsilon_j \rfloor, \text{ for minimizing} f_j \\ \lceil (f_j - f_j^{min})/\epsilon_j \rceil, \text{ for maxmizing} f_j \end{cases} \quad (1)$$

where f_j^{min} is the minimum possible value of the j-th objective (if the decision-makers don't know the minimum possible value exactly, use $f_j^{min} = 0$) and ϵ_j is the allowable tolerance in the j-th objective beyond which two values are significant to the user. This ϵ_j value is similar to the ϵ used in the ϵ-dominance definition [21]. The identification arrays make the whole search space into grids having ϵ_j size in the j-th objective [20]. The ϵ-domination is used first when the archive is updated with the offspring solutions. More details about the ϵ-MOEA can be found in [20].

3 Function Optimization by Conventional DE

A general MOP includes a set of n parameters (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

$$\begin{aligned} minimize : \boldsymbol{y} &= f(\boldsymbol{x}) = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \cdots, f_k(\boldsymbol{x})) \\ subject\ to : \boldsymbol{e}(\boldsymbol{x}) &= (e_1(\boldsymbol{x}), e_2(\boldsymbol{x}), \cdots, e_m(\boldsymbol{x})) \geq 0 \\ where : \boldsymbol{x} &= (x_1, x_2, \cdots, x_n) \in \boldsymbol{X} \\ \boldsymbol{y} &= (y_1, y_2, \cdots, y_k) \in \boldsymbol{Y} \end{aligned} \quad (2)$$

and $\boldsymbol{x}$ 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

$$l_i \leq x_i \leq u_i, i = 1, 2, \cdots, n \quad (3)$$

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

The algorithm of DE with the DE/rand/1/exp strategy [8] is as follows:

1. Generate the initial population with *NP* individuals, and set current iteration $k = 1$. Each individual is taken as a real valued vector $X_i, \forall i \in \{1, 2, \cdots, NP\}$, where X_is are objective variables.
2. Evaluate the fitness score for each individual $X_i, \forall i \in \{1, 2, \cdots, NP\}$, of the population based on the objective function, $f(X_i)$.
3. Stop if the halting criterion such as $k = MAX\_GEN$ is satisfied; otherwise, continue.
4. For each individual i, $\forall i \in \{1, 2, \cdots, NP\}$, select r_1, r_2, r_3 uniform randomly from $O \in \{1, 2, \cdots, NP\}$ with $r_1 \neq r_2 \neq r_3 \neq i$.
5. Generate the offspring using DE crossover-mutation operator as following:
   Mutation:
   $$X_i' = X_{r_1} + F \times (X_{r_2} - X_{r_3}) \quad (4)$$
   where $F > 0$ is a scaling factor, and x_{r_1} is known as the base vector. The trial point Y_i is found from its parents X_i and X_i' using the following crossover rule:
   Crossover:
   $$Y_i^j = \begin{cases} X_i'^j & \text{if } R^j \leq CR \text{ or } j = t \\ X_i^j & \text{if } R^j > CR \text{ and } j \neq t \end{cases} \quad (5)$$
   where t is a randomly chosen integer in the set $Q \in \{1, 2, \cdots, n\}$; the superscript j represents the j-th component of corresponding vectors; $R^j \in (0, 1)$, drawn uniformly for each j. And $CR > 0$ is the user defined probability of the crossover operator.
6. Select each trial vector Y_i for the $k+1$ iteration using the acceptance criterion:replace $X_i \in S$ with Y_i if $f(Y_i) \prec f(X_i)$, otherwise retain X_i. Set $k = k + 1$ and go to Step 3.

4 Our Approach: ϵ-ODEMO

Here, we propose an extension DE algorithm called ϵ-ODEMO for MOPs.

4.1 Orthogonal Initial Population

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 feasible solution space. In our presented manner, the algorithm can evenly scan the feasible 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.We apply the quantization technique and the orthogonal design to generate this initial population.

4.1.1 Design of the orthogonal array Although the proposed algorithm may require different orthogonal arrays (OAs) for different optimization problems. We will only need a special class of OAs. To design an 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 in the above equation. The orthogonal array needs to find a proper J to satisfy

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

where n is the number of the variables. In this study, we adopt the algorithm described in [17] to construct an orthogonal array. In particular, we use $L(R, C)$ to indicate the orthogonal array; and $a(i, j)$ 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.

4.1.2 Quantization For one decision variable with the boundary $[l, u]$, we quantize the domain into Q levels $\alpha_1, \alpha_2, \cdots, \alpha_Q$, where the design parameter Q is odd and α_i is given by

$$\alpha_i = \begin{cases} l & i = 1 \\ l + (i-1)(\frac{u-l}{Q-1}) & 2 \leq i \leq Q-1 \\ u & Q \end{cases} \tag{8}$$

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

4.1.3 Generation of Initial Population After constructing a proper OA and quantizing the domain of each decision variable, we can generate the initial population which can scatter uniformly over the feasible solution space. The algorithm for generating the initial population is omitted here, please refer [17] for details. Regularly, the number of the rows of the OA is larger than the population size NP, so we first create the archive with ϵ-nondominated solutions ,And then we generate the initial EA population from the archive and the orthogonal solutions. If $ar\_size > NP$, we select NP solutions from the archive randomly; or all of the ar_size solutions in the archive are inserted the EA population, and the remainder $NP - ar\_size$ solutions are selected from the orthogonal solutions randomly.

4.2 Producing New Solutions With DE/rand/1/exp Strategy

In this study, we use DE/rand/1/exp strategy described in Section 3 to produce the offspring solutions. Firstly, when the size of the archive $ar\_size \geq 5$, we select the mating parents from the archive to generate new solutions, which are used to update the archive with ϵ-domination and EA population with usual domination respectively. Secondly, we get the mating parents from the EA population to generate new solutions and update the archive and EA population.

4.3 Procedure of ϵ-ODEMO

The procedure of ϵ-ODEMO is similar to the ϵ-MOEA with the exception that in ϵ-ODEMO we generate the initial population with orthogonal design method and use DE/rand/1/exp to produce new solutions. The algorithm is followed by

Algorithm 2 The procedure of the proposed ϵ-ODEMO

```
Generate a proper OA and generate the orthogonal solutions OS
Create the archive E(0) with the ϵ-nondominated solutions of OS
Create the orthogonal initial population P(0) from E(0) and OS
while The maximum number of the fitness function evaluations (NFE) does not reach do
  if ar_size ≥ 5 then
    for i = 1 to ar_size do
      Produce the new solution with DE/rand/1/exp with archive members
      Update the archive using ϵ-dominance concept
      Update the EA population with usual domination
    end for
  end if
  for i = 1 to NP do
    Produce the new solution with DE/rand/1/exp with EA population members
    Update the archive using ϵ-dominance concept
    Update the EA population with usual domination
  end for
end while
```

5 Simulation Results

In order to test the performance of ϵ-ODEMO a number of two- and three-objective problems were used, where two-objective test problems (ZDT1, ZDT2, ZDT3, ZDT4 and ZDT6) are introduced in [22], and also have been used in [1,13,14,15,18]. And three-objective test problems (DTLZ1 and DTLZ6) are introduced in [23]. The brief information of the test problems is described in Table 1, where k is the number of the objective functions and n is the dimension of the decision vector. We also test these problems with three other approaches: (i) ϵ-DEMO, which is similar to ϵ-ODEMO except using random initial population, (ii) ϵ-MOEA proposed in [20], and (iii) ϵ-OMOEA, which uses orthogonal initial population mentioned above instead of the random initial population in ϵ-MOEA.

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

Problem	k	n	Property
ZDT1	2	30	high dimensionality, convex Pareto front
ZDT2	2	30	high dimensionality, non-convex Pareto front
ZDT3	2	30	high dimensionality, discontinuous Pareto front
ZDT4	2	10	many (21^9) local Pareto fronts
ZDT6	2	10	low density of solutions near Pareto front
DTLZ1	3	7	many ($11^5 - 1$) local Pareto fronts, linear hyper-plane Pareto front
DTLZ6	3	22	high dimensionality, 2^{19} disconnected Pareto-optimal regions

5.1 Performance Measures

There are three metrics used in this study. The smaller the value of these metrics, the better the performance of the algorithm.

- **Convergence metric** γ [1]: It measures the distance between the obtained nondominated front Q and the set P^*. of Pareto-optimal solutions.
- **Diversity metric** Δ [1]: It measures the extent of spread achieved among the nondominated solutions.
- **Generational distance** GD [15]: It is similar to the convergence metric. It measures the distance between the obtained nondominated front Q and the set P^*. of Pareto-optimal solutions.

For all the three metrics, we need to know the true Pareto front for a problem. Since we are dealing with artificial test problems, the true Pareto front is not difficult to be obtained. In our experiments we use uniformly spaced Pareto-optimal solutions as the approximation of the true Pareto front (For ZDT test problems, they were made available online at http://www.scis.ecu.edu.au/research/wfg/datafiles.html. And for DTLZ test problems, they were made available online at http://www.lania.mx/˜ccoello/EMOO/testfuncs/.).

5.2 Experimental Setup

For all experiments, we used the following parameters:

- Population size: *NP* = 100;
- Number of fitness function evaluations: *NFE* = 20,000, which is less than the compared approaches (NSGA-II, SPEA, PAES, PDEA, MODE, and DEMO/parent), where the *NFE* of them is 25,000;
- Probability of crossover: *CR* = 0.5;
- Scaling factor: *F* = 0.5;
- Positive integer in orthogonal design: *J* = 2;
- Number of the quantization levels: if $n > 29$, *Q* = 29; else *Q* = 21;
- The ϵ values for different problems are described in Table 2 in order to get roughly 100 solutions in the archive after 20,000 fitness function evaluations.

Table 2. The ϵ values for different test problems.

	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6	DTLZ1	DTLZ6
ϵ	[0.0075,0.0075]	[0.0075,0.0075]	[0.0025,0.0035]	[0.0075,0.0075]	[0.0075,0.0075]	[0.02,0.02,0.05]	[0.05,0.05,0.05]

5.3 Experimental Results

The performance of the four methods are compared on all the test problems over ten independent trials each respectively. The average execution time (in seconds) in each run using a PC with an Intel Celeron IV 2.53 GHz processor and 512MB memory running Microsoft Windows XP operating system is shown in the column labeled Time(s).

5.3.1 Two-objective Test Problems Tables 3 and Table 4 present the mean and variance of the values of the convergence and diversity metric, averaged on ten runs. Results of other algorithms are taken from the literature (see [1] for the results and parameter settings of both versions of NSGA-II, SPEA and PAES, [13] for PDEA, [14] for MODE, and [15] for DEMO/parent). Although in [15] they proposed three variants of DE algorithm for MOPs, DEMO/parent obtained almost the same performance compared with the other two variants. So we only select the results of DEMO/parent to compare with our approaches.

We also present the additional comparison of generational distance results in Tables 5 for PDEA [14] and DEMO/parent [15]. Once more, we present the mean and variance of the values of generational distance, averaged over 30 runs.

Fig. 1 shows the nondominated fronts obtained by a single run of ϵ-ODEMO. Table 6 summarizes the values of the convergence and diversity metrics for the nondominated fronts from Fig. 1.

5.3.2 Three-objective Test Problems With respect to three-objective problems DTLZ1 and DTLZ2, we only make comparisons on ϵ-ODEMO, ϵ-DEMO, ϵ-OMOEA, ϵ-MOEA. Table 7 shows the convergence metric values of the four approaches, averaged on 10 runs.

Fig. 2 and Fig. 3 shows the nondominated fronts obtained by a single run of ϵ-ODEMO and ϵ-OMOEA. Table 8 summarizes the values of the convergence metric for the nondominated fronts from Fig. 2 and Fig. 3.

6 Results Analysis

6.1 Two-objective Test Problems

For the five two-objective problems, [15] obtained superior results on these problems and gave a good discussion of the comparison between DEMO with NSGA-II, SPEA, PAES, PDEA and MODE. Hence, here we only discuss the comparison between our approaches and DEMO. From Table 3 - 6 and Fig. 1, we can see that

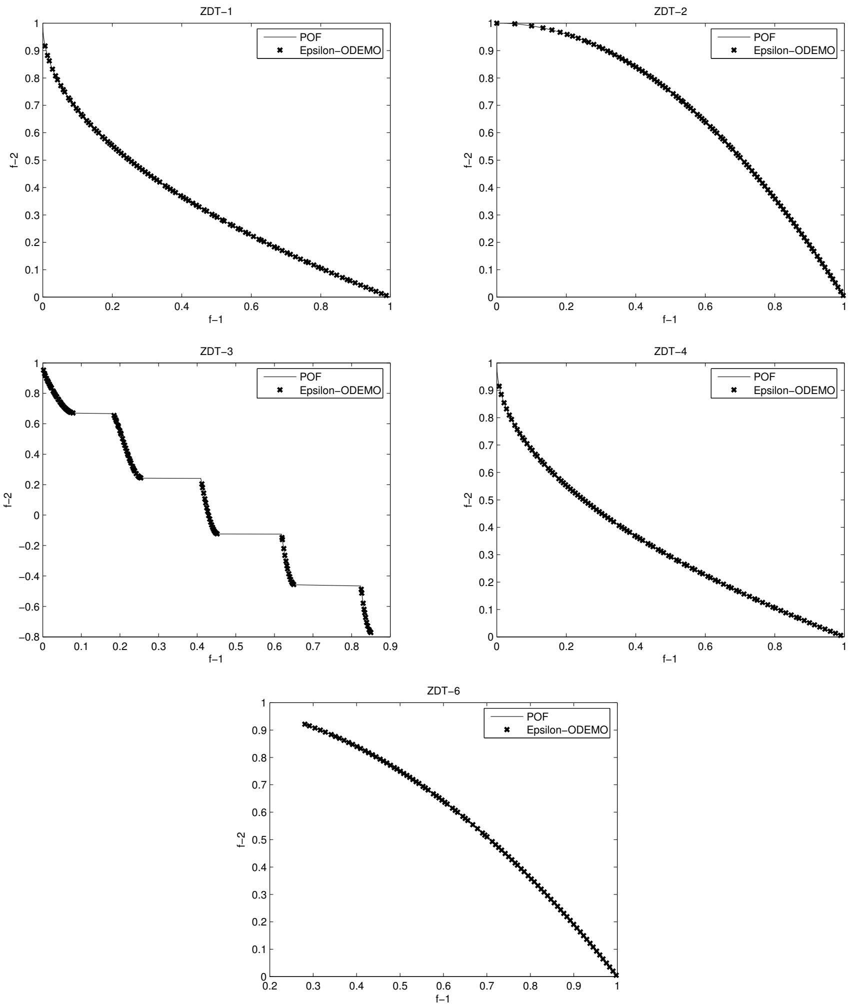

Fig. 1. Non-dominated solutions of the final archive obtained by ϵ-ODEMO on five ZDT test problems (see Table 6 for more details on these fronts). Where POF means Pareto-optimal front. The presented fronts are the outcome of a single run of ϵ-ODEMO.

Table 3. Statistics of the average results on test problems ZDT1, ZDT2, ZDT3 and ZDT4 over 30 independent runs. The results obtained by the proposed ϵ-ODEMO are shown in **boldface**. NA = Not Available.

Algorithm	ZDT1		
	Convergence γ	Diversity Δ	Time (s)
NSGA-II (real-code) [1]	0.033482±0.004750	0.390307±0.001876	NA
NSGA-II (binary-code) [1]	0.000894±0.000000	0.463292±0.041622	NA
SPEA [1]	0.001799±0.000001	0.784525±0.004440	NA
PAES [1]	0.082085±0.008679	1.229794±0.004839	NA
PDEA [13]	NA	0.298567±0.000742	NA
MODE [14]	0.005800±0.000000	NA	NA
DEMO/parent [15]	0.001083±0.000113	0.325237±0.030249	NA
ϵ-ODEMO	**0.000761±0.000058**	**0.360154±0.011059**	**2.553**
ϵ-DEMO	0.040202±0.018254	0.387340±0.040272	0.925
ϵ-OMOEA	0.000721±0.000023	0.358369±0.012453	6.201
ϵ-MOEA	0.020125±0.012800	0.364210±0.020230	6.188

Algorithm	ZDT2		
	Convergence γ	Diversity Δ	Time (s)
NSGA-II (real-code) [1]	0.072391±0.031689	0.430776±0.004721	NA
NSGA-II (binary-code) [1]	0.000824±0.000000	0.435112±0.024607	NA
SPEA [1]	0.001339±0.000000	0.755148±0.004521	NA
PAES [1]	0.126276±0.036877	1.165942±0.007682	NA
PDEA [13]	NA	0.317958±0.001389	NA
MODE [14]	0.005500±0.000000	NA	NA
DEMO/parent [15]	0.000755±0.000045	0.329151±0.032408	NA
ϵ-ODEMO	**0.000764±0.000035**	**0.276872±0.007013**	**2.502**
ϵ-DEMO	0.190147±0.081243	0.615820±0.051986	0.621
ϵ-OMOEA	0.000760±0.000015	0.283013±0.045573	6.567
ϵ-MOEA	0.034273±0.020354	0.402345±0.077234	6.342

Algorithm	ZDT3		
	Convergence γ	Diversity Δ	Time (s)
NSGA-II (real-code) [1]	0.114500±0.007940	0.738540±0.019706	NA
NSGA-II (binary-code) [1]	0.043411±0.000042	0.575606±0.005078	NA
SPEA [1]	0.047517±0.000047	0.672938±0.003587	NA
PAES [1]	0.023872±0.000010	0.789920±0.001653	NA
PDEA [13]	NA	0.623812±0.000225	NA
MODE [14]	0.021560±0.000000	NA	NA
DEMO/parent [15]	0.001178±0.000059	0.309436±0.018603	NA
ϵ-ODEMO	**0.000915±0.000050**	**0.534329±0.018301**	**2.453**
ϵ-DEMO	0.008754±0.003127	0.632701±0.025327	0.924
ϵ-OMOEA	0.006453±0.007956	0.687538±0.032879	6.583
ϵ-MOEA	0.005689±0.003357	0.673541±0.012586	6.037

Algorithm	ZDT4		
	Convergence γ	Diversity Δ	Time (s)
NSGA-II (real-code) [1]	0.513053±0.118460	0.702612±0.064648	NA
NSGA-II (binary-code) [1]	3.227636±7.307630	0.479475±0.009841	NA
SPEA [1]	7.340299±6.572516	0.798463±0.014616	NA
PAES [1]	0.854816±0.527238	0.870458±0.101399	NA
PDEA [13]	NA	0.840852±0.035741	NA
MODE [14]	0.638950±0.500200	NA	NA
DEMO/parent [15]	0.001037±0.000134	0.359905±0.037672	NA
ϵ-ODEMO	**0.000712±0.000056**	**0.354847±0.003956**	**2.325**
ϵ-DEMO	0.856829±0.702439	0.679368±0.120357	0.903
ϵ-OMOEA	0.010389±0.009354	0.180321±0.531570	6.237
ϵ-MOEA	8.137894±5.27689	0.927910±0.025221	5.832

Table 4. Statistics of the results on test problem ZDT6 over 30 independent runs. The results obtained by the proposed ϵ-ODEMO are shown in **boldface**. NA = Not Available.

Algorithm	ZDT6		
	Convergence γ	Diversity Δ	Time (s)
NSGA-II (real-code) [1]	0.296564±0.013135	0.668025±0.009923	NA
NSGA-II (binary-code) [1]	7.806798±0.001667	0.644477±0.035042	NA
SPEA [1]	0.221138±0.000449	0.849389±0.002713	NA
PAES [1]	0.085469±0.006664	1.153052±0.003916	NA
PDEA [13]	NA	0.473074±0.021721	NA
MODE [14]	0.026230±0.000861	NA	NA
DEMO/parent [15]	0.000629±0.000044	0.442308±0.039255	NA
ϵ-ODEMO	**0.000581±0.000030**	**0.204142±0.005012**	**1.559**
ϵ-DEMO	0.875253±0.0573254	1.530470±0.046340	0.387
ϵ-OMOEA	0.000618±0.000102	0.180214±0.000864	6.189
ϵ-MOEA	0.675321±0.537001	0.795721±0.068538	6.057

Table 5. Generational distance achieved by PDEA, DEMO and ϵ-ODEMO on the test problems ZDT1, ZDT2, ZDT3, ZDT4 and ZDT6. The results obtained by the proposed ϵ-ODEMO are shown in **boldface**.

Algorithm	Generational distance				
	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6
PDEA [13]	0.00062±0.00000	0.00065±0.00000	0.00056±0.00000	0.61826±0.82688	0.02389±0.00329
DEMO/parent [15]	0.00023±0.00005	0.00009±0.00001	0.00016±0.00001	0.00020±0.00005	0.00007±0.00001
ϵ-ODEMO	**0.00010±0.00001**	**0.00009±0.00000**	**0.00013±0.00001**	**0.00009±0.00001**	**0.00007±0.00000**

Table 6. Metric values for the nondominated fronts shown in Fig. 1 by ϵ-ODEMO.

Problem	Convergence γ	Diversity Δ
ZDT1	0.000799	0.372004
ZDT2	0.000753	0.268825
ZDT3	0.000974	0.383188
ZDT4	0.000750	0.353701
ZDT6	0.000613	0.193624

Table 7. Statistics of the results on test problems DTLZ1 and DTLZ6 over 30 independent runs. The results obtained by the proposed ϵ-ODEMO are shown in **boldface**.

Algorithm	DTLZ1		DTLZ6	
	Convergence γ	Time(s)	Convergence γ	Time(s)
ϵ-ODEMO	**0.004389±0.000204**	**1.872**	**0.020387±0.000789**	**3.427**
ϵ-DEMO	1.732468±0.182287	17.320	0.039879±0.025680	1.199
ϵ-OMOEA	0.003790±0.000158	6.251	0.013768±0.002786	7.350
ϵ-MOEA	7.789134±1.867357	67.035	0.063201±0.031879	6.044

Table 8. Convergence metric values for the nondominated fronts shown in Fig. 2 and Fig. 3.

Algorithm	DTLZ1	DTLZ6
ϵ-ODEMO	0.004370	0.020155
ϵ-OMOEA	0.003997	0.013433

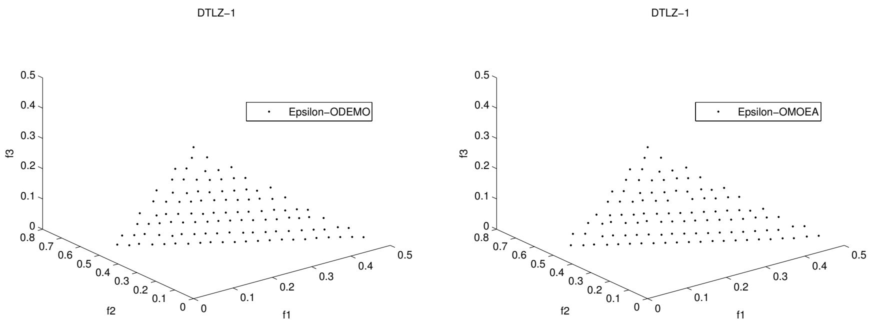

Fig. 2. Non-dominated solutions of the final archive obtained by ϵ-ODEMO and ϵ-OMOEA on DTLZ1 (see Table 8 for more details on these fronts). The presented fronts are the outcome of a single run.

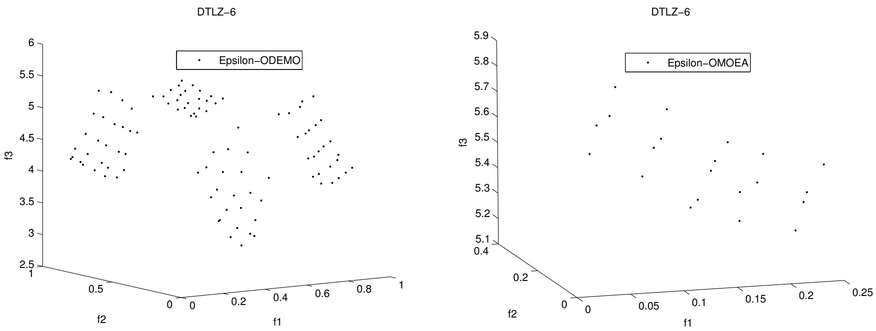

Fig. 3. Non-dominated solutions of the final archive obtained by ϵ-ODEMO and ϵ-OMOEA on DTLZ6 (see Table 8 for more details on these fronts). The presented fronts are the outcome of a single run.

– ϵ-ODEMO and ϵ-OMOEA can get very good results on all of the five test problems in both goals of multiobjective optimization (convergence to the true Pareto front and uniform spread of solutions along the front). However, the performance of ϵ-DEMO and ϵ-MOEA are not good. This indicates that the orthogonal initial population can evenly scan the feasible solution space once to locate good points for further exploration in subsequent iterations.
– For ZDT1, ZDT2 and ZDT3, they have high-dimensionality, but many MOEAs have achieved very good results on these problems in both goals of multiobjective optimization. The results for ZDT1, ZDT2 and ZDT3 shown in Table 3 demonstrate that ϵ-ODEMO and ϵ-OMOEA can obtain better convergence metrics than DEMO/parent. But the diversity metrics for ZDT1 and ZDT3 are slightly worse than DEMO due to the absence of extreme solutions in the Pareto-optimal front (POF). Since the ϵ-dominance concept is used in ϵ-ODEMO and ϵ-OMOEA, the extreme solutions usually get dominated by solutions within ϵ and which are better

in other objectives [20]. ϵ-OMOEA outperforms ϵ-ODEMO on ZDT1 and ZDT2. But ϵ-ODEMO gets better results on ZDT3.

– ZDT4 has 21^9 local Pareto fronts, which is difficult for many optimizers. Our proposed ϵ-ODEMO can find the true POF in all of the ten runs and get better results than DEMO/parent. Another three approaches (ϵ-OMOEA, ϵ-MOEA and ϵ-DEMO) are trapped into a local Pareto front sometimes.
– For the results from ZDT6 in Table 4, ϵ-ODEMO, ϵ-OMOEA and DEMO/parent obtain similar results on convergence metric. However, ϵ-ODEMO and ϵ-OMOEA get better results of diversity metric than DEMO/parent. But ϵ-DEMO and ϵ-MOEA get worse results in both goals of multiobjective optimization.
– With respect to the generational distance metric, Table 5 shows that among PDEA, DEMO/parent and ϵ-ODEMO, ϵ-ODEMO can find the best results on all of the test problems.
– From Fig. 1, we can see that all of the solutions obtained by ϵ-ODEMO are scattered on the true POF on all of five test problems.
– Regarding computational cost, we don't run NSGA-II, SPEA, PAES, PDEA, MODE and DEMO in my PC, but from the analysis in [20], all of them are more time consuming than the approaches only used the ϵ-dominance concept. Although ϵ-DEMO requires the least computational time on all of five test problems, it obtains worse results on the convergence and diversity results. However, ϵ-ODEMO can get very good results on all of five test problems in both goals of multiobjective optimization with less time.

6.2 Three-objective Test Problems

With respect to three-objective problems DTLZ1 and DTLZ2, we summarize the results in Table 7. And the nondominated fronts obtained by a single run of ϵ-ODEMO and ϵ-OMOEA are illustrated in Fig. 2 and Fig. 3. The results show that

– ϵ-OMOEA obtains the best results of convergence metric on two test problems, followed by ϵ-ODEMO. ϵ-DEMO and ϵ-MOEA find worse results, especially for DLTZ1.
– On DLTZ1, ϵ-OMOEA can get a better result than ϵ-ODEMO with more computational time. They can all find the true POF. But ϵ-DEMO and ϵ-MOEA find many local Pareto solutions in the final archive ($ar\_size \gg 100$), hence, they get the worst results of convergence.
– With respect to DTLZ6, there is an interesting result from Table 8 and Fig. 3 that although ϵ-OMOEA finds a better result of convergence than ϵ-ODEMO, ϵ-OMOEA can obtain only one Pareto-optimal region, however, ϵ-ODEMO can find the true POF in all disconnected Pareto-optimal front regions. Therefore, ϵ-ODEMO was able to get a better distribution of the Pareto solutions than ϵ-OMOEA.

From the comparison above, we can conclude that our proposed approach, ϵ-ODEMO, produced competitive results based on quality with respect to many other techniques representative of the state-of-the-art in multiobjective optimization. ϵ-ODEMO can deal with two- and three-objective problems of diverse complexities; problems with low

(ZDT4, ZDT6 and DTLZ1) and high (ZDT1, ZDT2, ZDT3 and DTLZ6) dimensionality, with different types of Pareto fronts (convex, non-convex, discontinuous, thin density and non-uniform spread) and with many local Pareto fronts (ZDT4 and DTLZ1). Furthermore, the approach is very fast in terms of the computational time on each of the test problems.

7 Conclusion

In this paper, we proposed a novel DE algorithm based on ϵ-dominance concept and orthogonal design method for MOPs. ϵ-ODEMO implies the orthogonal design method with quantization technique to generate the initial population of points that are scattered uniformly over the feasible solution space, so that the algorithm can evenly scan the feasible solution space once to locate good points for further exploration in subsequent iterations. And it uses DE/rand/1/exp strategy to produce offspring solutions. Meanwhile, in order to find good distribution Pareto solutions with less computational time, ϵ-dominance concept and efficient parent and archive update strategies are adopted to update the archive and population. We tested our proposed ϵ-ODEMO on a number of two and three objective problems. From the analysis of the results we can conclude that ϵ-ODEMO can obtain a good distribution Pareto solutions on all of the test problems. Moreover, it requires small computational time. Although ϵ-OMOEA got slightly better results on some test problems (ZDT1, ZDT2, ZDT6 and DTLZ1) than ϵ-ODEMO, it needed more computational time and it obtained worse results on ZDT3, ZDT4 and DTLZ6. Hence, we recommend our proposed ϵ-ODEMO be used in future experimentation. Our future work consists on using the proposed ϵ-ODEMO to solve the constrained MOPs and dynamic MOPs.

Acknowledgment

The authors would like to thank the anonymous reviewers for their constructive advice and Kanpur Genetic Algorithms Laboratory (KanGAL) for making the source code of ϵ-MOEA available online at http://www.iitk.ac.in/kangal/codes.shtml. This work is supported by the Humanities Base Project of Hubei Province of China(Grant No. 2004B0011) and the Natural Science Foundation of Hubei Province of China(Grant No. 2003ABA043).

References

1. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGACII. IEEE Transactions on Evolutionary Computation 6 (2002) 182 - 197
2. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: Improving the strength pareto evolutionary algorithm. Technical Report 103, Computer Engineering and Networks Laboratory, (2001)
3. Schaffer, J. D.: Multiple Objective Optimization with Vector Evaluated Genetic Algorithms. Ph. D. thesis, Vanderbilt University. Unpublished. (1984)
4. Hajela, P. and Lin, C. Y.: Genetic search strategies in multicriterion optimal design. Structural Optimization 4 (1992) 99 - 107

5. Fonseca, C. M. and Fleming, P. J.: Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. In S. Forrest (Ed.), Proceedings of the Fifth International Conference on Genetic Algorithms, (1993) 416 - 423
6. Horn, J. and Nafpliotis,N.: Multiobjective optimization using the niched pareto genetic algorithm. IlliGAL Report 93005, Illinois Genetic Algorithms Laboratory, University of Illinois, Urbana, Champaign. (1993)
7. Knowles,J.D., Corne, D.W.: Approximating the Nondominated Front Using the Pareto Archived Evolution Strategy. Evolutionary Computation 8(2) (2000) 149 - 172
8. Storn, R. and Price, K.: Differential evolution–A simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization. 11 (1997) 341 - 359
9. Storn, R. and Price, K.: Home Page of Differential Evolution. Available online at http://www.ICSI.Berkeley.edu/~storn/code.html. (2003)
10. Lampinen, J.: A bibliography of differential evolution algorithm. Available online at http://www2.lut.fi/~jlampine/debiblio.htm
11. Abbass, H.A., Sarker, R., Newton, C.: PDE: A pareto-frontier differential evolution approach for multi-objective optimization problems. In: Proceedings of the Congress on Evolutionary Computation 2001 (CEC2001). Volume 2, Piscataway, New Jersey, IEEE Service Center (2001) 971 - 978
12. Abbass, H.A.: The self-adaptive pareto differential evolution algorithm. In: Congress on Evolutionary Computation (CEC2002). Volume 1, Piscataway, New Jersey, IEEE Service Center (2002) 831 - 836
13. Madavan, N.K.: Multiobjective optimization using a pareto differential evolution approach. In: Congress on Evolutionary Computation (CEC2002). Volume 2, Piscataway, New Jersey, IEEE Service Center (2002) 1145 - 1150
14. Xue, F., Sanderson, A.C., Graves, R.J.: Pareto-based multi-objective differential evolution. Proceedings of the 2003 Congress on Evolutionary Computation (CEC2003). Volume 2, Canberra, Australia, IEEE Press (2003) 862 - 869
15. Robič, T. and Filipič, B.: DEMO: Differential Evolution for Multiobjective Optimization. proceedings of the Third International Conference on Evolutionary Multi-Criterion Optimization (EMO-05), (2005) 520 - 533
16. Fang, K. T. and Ma, C. X.: Orthogonal and Uniform Design (in Chinese). Science Press, (2001)
17. Leung, Y. W. and Wang, Y.: An Orthogonal Genetic Algorithm with Quantization for Global Numerical Optimization. IEEE Transactions on Evolutionary Computation, 5(1) (2001) 41 - 53
18. S. Zeng, L. Kang, L. Ding: An Orthogonal Multiobjective Evolutionary Algorithm for Multiobjective Optimization Problems with Constraints. Evolutionary Computation, 12(1) (2004) 77 - 98
19. S. Zeng, S. Yao, L., and Y. Liu: An Efficient Multi-objective Evolutionary Algorithm: OMOEA-II. In C. A. Coello Coello *et al*. (Eds.), Third International Conference on Evolutionary Multi-Criterion Optimization (EMO-05), (2005) 108 - 119
20. Deb, K., Mohan, M. and Mishra, S.: Towards a quick computation of well-spread Pareto-optimal solutions. Second Evolutionary Multi-Criterion Optimization (EMO-03),Springr LNCS 2632(2003) 222 - 236
21. Laumanns, M., Thiele, L., Deb, K., and Zitzler, E.: Combining convergence and diversity in evolutionary multi-objective optimization. Evolutionary Computation, 10(3) (2002) 263 - 282
22. Zitzler, E., Deb, K., Thiele, L.: Comparison of multiobjective evolutionary algorithms: Empirical results. Evolutionary Computation 8 (2000) 173 - 195
23. Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable Multi-Objective Optimization Test Problems. In: Congress on Evolutionary Computation (CEC2002), (2002) 825 - 830