Multiobjective Differential Evolution with External Archive

V. L. Huang, P. N. Suganthan, A. K. Qin and S. Baskar
School of Electrical and Electronic Engineering
Nanyang Technological University
Singapore 639798
{huangling, qinkai}@pmail.ntu.edu.sg, epnsugan@ntu.edu.sg, baskar_mani@yahoo.com

Abstract: This paper presents an approach to incorporate Pareto dominance into differential evolution (DE) in order to handle optimization problems with more than one objective function by using the DE. Unlike the existing proposals to extend the DE to solve multiobjective optimization problems, our algorithm uses an external archive to store nondominated solutions. In order to generate next generation solutions, the current population and the nondominated solutions stored in the external archive are used. We also propose a new harmonic average distance to measure the crowding degree of the solutions more accurately. Simulation results on nine test problems show that the proposed MODE, in most problems, is able to find much better spread of solutions and better convergence to the true Pareto-optimal front compared to three other multiobjective optimization evolutionary algorithms, and the new crowding degree estimation method has been proven to improve the diversity of the nondominated solutions along the Pareto front.

Keywords: multiobjective evolutionary algorithm, multiobjective optimization, multiobjective differential evolution, external archive.

1. Introduction

The development of evolutionary algorithms to solve multiobjective optimization problems has attracted much interest recently and a number of multiobjective evolutionary algorithms (MOEAs) have been suggested (Srinivas and Deb, 1994; Zitzler and Thiele, 1999; Knowles and Corne, 2000; Zitzler, Laumanns and Thiele, 2001; Deb et al. 2002). While most of the these algorithms were developed taking into consideration two common goals, namely fast convergence to the Pareto-optimal front and good distribution of solutions along the front, each algorithm employs a unique combination of specific techniques to achieve these goals. SPEA (Zitzler and Thiele, 1999) uses a secondary population to store the nondominated solutions and cluster mechanism to ensure diversity. PAES (Knowles and Corne, 2000) use a histogram-like density measure over a hyper-grid division of the objective space. NSGA-II (Deb et al, 2002) incorporates elitist and crowding approaches.

The main advantage of evolutionary algorithms (EAs) in solving multi-objective optimization problems is their ability to find multiple Pareto-optimal solutions in one single run. In 1995s, a new global optimization algorithm, differential evolution (DE), was proposed (Storn and Price, 1995). As DE shares many similarities with evolutionary computation techniques such as genetic algorithms (GA), and has been found to be successful in single objective optimization problems, recently there are several proposals to extend DE to handle multiobjective problems. One approach is presented to optimise train movement through tuning fuzzy membership functions in mass transit systems (Chang, Xu and Quek, 1999). Abbass, Sarker, and Newton (2001) introduce a Pareto-frontier Differential Evolution algorithm (PDE) to solve multiobjective problem by incorporating Pareto dominance. This PDE is also extended with self-adaptive crossover and mutation (Abbass, 2002). Madavan (2002) extended DE to solve multiobjective optimization problems by incorporating a nondominated sorting and ranking selection scheme of NSGA-II. In Babu and Jehan's study, DE is applied for solving multi-objective optimization problems using penalty function method and weighting factor method (Babu and Jehan, 2003). Another approach involves Pareto-based evaluation to DE for solving multiobjective decision problems and has been applied to an enterprise planning problem with two objectives namely, cycle time and cost (Xue, 2003; Xue, Sanderson and Graves, 2003). And further Feng Xue et al. (2005) use a Fuzzy Logic Controller (FLC) to dynamically adjust the parameters of this multiobjective evolutionary algorithm. Recently researchers also have devoted rotated problem (Antony and Li, 2004), parallel multi-population DE algorithm (Parsopoulos et al 2004) and constrained multiobjective problems (Kukkonen and Lampinen, 2004a; 2004b).

However, none of the existing approach extends DE to deal with multiobjective optimization problems with an external archive, which is an effective notion of elitism and has been successfully used in other MOEAs (Zitzler and Thiele, 1999; Knowles and Corne, 2000; Zitzler, Laumanns and Thiele, 2001; Coello, Pulido and Lechuga, 2004). Further, existing multiobjective DE implementations have not been comprehensively evaluated and compared with the other multiobjective evolutionary algorithms. In this paper, we present an approach to extend DE algorithm to solve multiobjective optimization problems with an external archive, which we call "multiobjective differential evolution" (MODE). From the simulation results on several standard test functions, we find that the MODE overall outperforms three highly competitive MOEAs: the nondominated sorting genetic algorithm-II (NSGA-II) (Deb et al, 2002), the multiobjective particle swarm optimization (MOPSO) (Coello, Pulido and Lechuga, 2004), Pareto archive evolution strategy (PAES) (Knowles and Corne, 2000).

The remainder of the paper is organized as follows. Section 2 summarizes differential evolution algorithm. In Section 3, we describe multiobjective differential evolution algorithm in details. Section 4 presents simulation and comparative results of MODE and three other competitive MOEAs. To improve diversity, section 5 proposes a new crowding degree estimation method. The paper is concluded in Section 6.

2. The Differential Evolution Algorithm

Differential evolution (DE) is a simple population-based, direct-search algorithm for global optimization (Price, 1996) and was proposed by Storn and Price (1995). It has demonstrated its robustness and effectiveness in a variety of applications, such as neural network learning (Ilonen, Kamarainen, and Lampinen, 2003), IIR-filter design (Storn, 1996), and the optimization of aerodynamic shapes (Rogalsky, Derksen, and Kocabiyik, 1999). The main features of DE are summarized in the following.

Let $S \subset \Re^n$ be the search space of the problem under consideration. Then, the DE utilizes NP, n–dimensional vectors,

$$X_i = (x_{i1}, \ldots, x_{in}) \in S, \qquad i = 1, \ldots, NP,$$

as a population for each generation of the algorithm. The initial population is chosen randomly and should cover the entire parameter space. At each generation, DE employs both mutation and crossover (recombination) to produce one trial vector $U_{i,G}$ for each 'parent' vector $X_{i,G}$. Then, a selection phase takes place, where each individual of the new population is compared to the corresponding individual of the old population, and the best between them is selected as a member of the population in the next generation.

For each parent vector $X_{i,G}$, a mutant vector $V_{i,G+1}$ is generated using the following equation (Price, 1996), (Storn and Price, 1997)

$$V_{i,G+1} = X_{best,G} + F(X_{r_1,G} - X_{r_2,G}) + F(X_{r_3,G} - X_{r_4,G}), \tag{1}$$

where, random indexes $r_1, r_2, r_3, r_4, r_5 \in \{1, 2, \ldots, NP\}$ are mutually different integers and also different from the current object vector index i. F is a scaling factor $\in [0, 2]$ and $X_{best,G}$ is the best individual of the population at generation G.

After the mutation phase, the crossover operator is used to increase the diversity.

$$u_{j,i,G+1} = \begin{cases} v_{j,i,G+1}, & \text{if}(rand_j[0,1] < CR)\text{or}(j = j_{rand}) \\ x_{j,i,G}, & \text{otherwise} \end{cases} \tag{2}$$

and $$U_{i,G+1} = \left(u_{1,i,G+1}, u_{2,i,G+1}, \ldots u_{n,i,G+1}\right) \tag{3}$$

where, $j = 1, 2, \ldots n$; CR is user-specified crossover constant $\in [0,1]$; j_{rand} is a randomly chosen index from $\{1, 2, \ldots, NP\}$, which ensures that the vector $U_{i,G+1}$ will differ from its parent $X_{i,G}$ by at least one parameter.

To decide whether the vector $U_{i,G+1}$ should be a member of generation $G+1$, it is compared with parent vector $X_{i,G}$.

$$X_{i,G+1} = \begin{cases} U_{i,G+1}, & \text{if} \quad f(U_{i,G+1}) < f(X_{i,G}), \\ X_{i,G}, & otherwise \end{cases} \tag{4}$$

With the membership of the next generation thus selected, the evolutionary cycle in DE repeats until a stopping condition is satisfied.

3. MODE with External Archive

In our work, we propose a novel approach to extend DE to multiobjective optimization problem. The main differences of our approach with respect to the other proposals in the literature are:

a. We make use of an external archive to serve three separate purposes. First, it saves and updates well spread nondominated solutions that would be the Pareto optimal solutions when the algorithm converges to the Pareto front. Second, it is used as an aid to choose between parent and child when they do not dominate each other. This process provides the selection pressure by pushing the archive to retain better solutions. Finally, we use the external archive as a pool to select the $X_{best,G}$ for DE. In this way, the approach could converge fast and at the same time maintain good diversity.

b. We use crowding distance to measure the crowding degree of the solutions, which is used as a criterion to select less crowded one between nondominated parent and child as the new parent for next generation, and delete the crowded archive members when the external archive population has reached its maximum size.

c. We compare our MODE with other representative state-of-the-art in multiobjective evolution algorithms on convergence and diversity metrics.

Table 1: MODE with External Archive

Input: N (population size), $N_{\max}$ (maximum size of archive)

F (scaling factor), CR (crossover probability), Stopping criterion

Output: A （Archived solutions）

Step 1. Initialize the parent population P.

Step 2. Evaluate the fitness value of the parent population.

Step 3. WHILE stopping criterion is not satisfied

DO

Step 3.1 Compute child solutions according to DE

For i=1 to i=N

Mutation:

$$V_{i,G} = P_{best,G} + F(P_{r_1,G} - P_{r_2,G}) + F(P_{r_3,G} - P_{r_4,G}), \tag{5}$$

where $P_{best,G}$ is a value randomly taken from A. $P_{r_1}, P_{r_2}, P_{r_3}$, and P_{r_4} are randomly selected mutually different individuals in the parent population.

Crossover:

$$\forall j \le D, c_{j,i,G+1} = \begin{cases} v_{j,i,G+1}, & \text{if}(rand_j[0,1] < CR)\text{or}(j = j_{rand}) \\ p_{j,i,G}, & \text{otherwise} \end{cases} \tag{6}$$

end For

Step 3.2 Maintain the child solution within the search space if any variable is outside its boundaries. When a variable exceeds its search bound, we set the variable equal to the corresponding bound. This method is also used in MOPSO and NSGA-II approach (Coello, Pulido, and Lechuga, 2004) (Deb et al. 2002).

Step 3.3 Evaluate the fitness of each child solution.

Step 3.4 Select a new parent solution between parent p_G and child c_G:

If p_G dominates c_G, c_G is rejected, $p_{G+1} = p_G$.

If c_G dominates p_G, $p_{G+1} = c_G$, use Update_archive in Table 2 to compare c_G with A_G.

If p_G and c_G are nondominated to each other, use Update_archive to compare c_G with A_G and the less crowded one will be the new parent p_{G+1}.

Step 3.5 Update the external archive

When A_G exceeds the maximum size, we select the less crowded solutions based on crowding distance to keep the archive size at $N_{\max}$.

Step 3.6 Increment the generation count.

Step 4. END WHILE

3.1. External Archive

We use an external archive to keep a historical record of the nondominated solutions obtained during the search process. Initially, this archive is empty. As the evolution progresses, good solutions enter the archive. However, the size of the true nondominated set can be very large.

The computational complexity of maintaining the archive increases with the archive size. Moreover, considering the use of an external archive as a pool to select the $X_{best,G}$, the archive size also affects the complexity of selection. Hence, the size of the archive will be restricted to a pre-specified value as many other researchers have involved (Zitzler and Thiele, 1999; Knowles and Corne, 2000; Zitzler, Laumanns and Thiele, 2001; Coello, Pulido and Lechuga, 2004).

Table 2. Update_archive

(Updating the archive when a nondominated child is obtained in Step 3.4)

If c_G is dominated by any member of A_G,

discard c_G

else if c_G dominates a set of members $D(c_G)$ from A_G

$A_G = A_G \setminus D(c_G)$

$A_G = A_G \cup \{c_G\}$

else A_G and c_G are nondominated,

$A_G = A_G \cup \{c_G\}$

The nondominated child solutions obtained at each generation are compared one by one in Step 3.4 (Table 1) with the current archive, which contains the set of nondominated solutions found so far. There are three cases as illustrated in Table 2: 1. If the new solution is dominated by a member of the external archive, the new solution is rejected. 2. If the new solution dominates some member(s) of the archive, then the dominated members in the archive are deleted and the new solution enters the archive. 3. The new solution does not dominate any archive members and none of the archive member dominates the solution. This implies that the new solution belongs to the nondominated front and it enters the archive. Finally, when the external archived population reaches its maximum allowed capacity, we use crowding distance measure to reduce its size to maximum size (Step 3.5 in Table 1). In this way, we keep the diversity of the archived solutions.

The archive maintenance we use in MODE approach is similar to the schemes employed in PAES except in the third case. In PAES, if archive is not full, a new nondominated child solution enters archive. Otherwise, the solution replaces a member of the archive residing in the most crowded grid location to maintain the maximum archive size. In this way, every time we have to compare the new solution with the most crowded archive member. While in MODE, whether or not the archive is full, all nondominated solutions enter the archive and after all nondominated solutions in one generation enter the archive, we use crowding distance measure explained below to enforce the maximum archive size.

3.2. Crowding Degree Estimation

There are several crowding degree estimation methods employed in MOEA to maintain the archive when the archive is full. PAES and MOPSO use adaptive hypercubes, where we need choose an appropriate depth parameter (PAES) or the number of divisions (MOPSO) to control the hypercube size. Since the size of the hypercubes is adaptive with the bounds of the entire search space, when solutions converge near the Pareto front, the hypercubes are comparatively large.

In our approach, we estimate the density of solution with respect to crowding distance in which no user-defined parameter is required (Deb et al, 2002). This crowding degree estimation method is invoked in two situations. First, when parent and child do not dominate each other, we calculate the crowding distances of the parent and child. The one with the larger distance is chosen as the new parent of the next generation. Figure 1 shows an example where the crowding distance of the child is smaller than that of the parent implying that the child is located in a less crowded region. Hence, the child is chosen as the new parent.

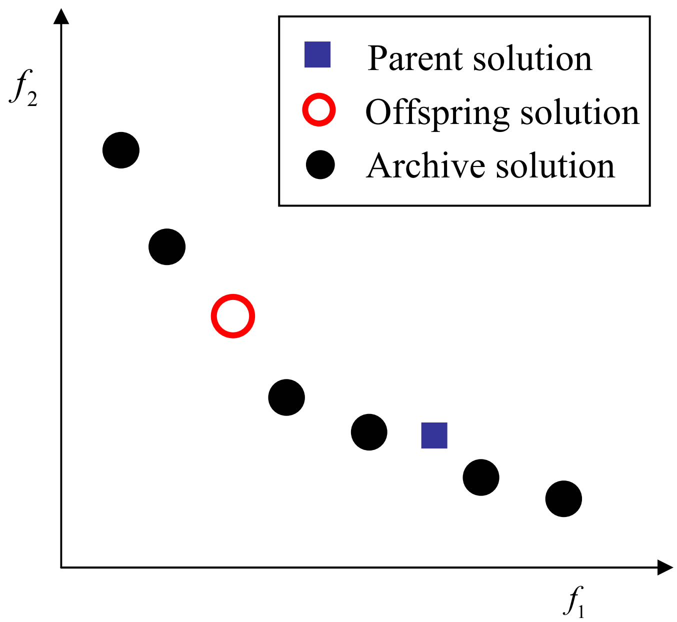

Figure 1: An example of nondominated parent and child

Secondly, if the external archived population reaches its maximum size, the crowding distance values of all archive members are calculated and sorted from large to small. The first N_{max} (maximum size of archive) members are kept while the remaining ones are deleted from the archive.

3.3. Handling Constraints

Since constraints are frequently associated with real-world optimization problems, we also use constrained-domination to handle constraints (Deb et al. 2002).

Definition 1: A solution i is said to constrained-dominate a solution j, if any of the following conditions is true.

a. Solution i is feasible and solution j is not.

b. Solution i and j are both infeasible, but solution i has a smaller overall constrain violation.

c. Solutions i and j are feasible and solution i dominates solution j.

According to this constrained-domination principle, MODE can deal with constraint problems without changing the modularity or computational complexity. The rest of the MODE procedure remains the same as described previously.

3.4. The Time Complexity of MODE

MODE requires M comparison to compare the mutant solution and the current solution and in the worst case, MODE requires further $M \times \bar{N}$ comparisons to compare the current solution with archive. Where M is the number of objectives in the problem, $\bar{N}$ is the current archive size. In the worst case, $\bar{N} = N + N_{\max}$. The complexity of calculating and sorting the crowding distance is $\mathrm{O}(M\bar{N}\log\bar{N})$ and $\mathrm{O}(\bar{N}\log\bar{N})$ respectively. If $\bar{N}$ and N are of the same order, the overall complexity of MODE is $\mathrm{O}(MN\log N)$.

4. Simulation Results

4.1. Test problems

In multi-objective evolutionary computation, researchers have used a number of different test problems with known sets of Pareto-optimal solutions. In this paper, we use nine test problems that are chosen from the standard multiobjetive evolutionary algorithm literature.

- Test problem 1: Schaffer's problem (SCH) has a convex Pareto front (Schaffer, 1987).

$$f_1(x) = x^2$$
$$f_2(x) = (x-2)^2$$

where $n = 1$ and $x \in [-10^3, 10^3]$. The optimal solutions are $x \in [0, 2]$.

- Test problem 2: Fonseca and Fleming (1998) proposed a two-objective problem (FON) having a nonconvex Pareto front.

$$f_1(x) = 1 - \exp(-\sum_{i=1}^{3} (x_i - \frac{1}{\sqrt{3}})^2)$$
$$f_2(x) = 1 - \exp(-\sum_{i=1}^{3} (x_i + \frac{1}{\sqrt{3}})^2)$$

where $n = 3$ and $x_i \in [-4, 4]$. The optimal solutions are $x_1 = x_2 = x_3 \in [-1/\sqrt{3}, 1/\sqrt{3}]$.

- Test problem 3: KUR, proposed by Kursawe (1990), is more complicated. The Pareto-optimal set is disconnected as well as nonconvex.

$$f_1(x) = \sum_{i=1}^{n-1} (-10 \exp(-0.2\sqrt{x_i^2 + x_{i+1}^2}))$$
$$f_2(x) = \sum_{i=1}^{n} (|x_i|^{0.8} + 5\sin(x_i^3))$$

where $n = 3$ and $x_i \in [-5, 5]$. For the optimal solutions, refer Deb K. (2001).

The following test problem 4, 5, 6, 7 and 8 are all chosen from Zitzler-Deb-Thiele's test set (Zitzler et al. 2000).

- Test problem 4: ZDT1 has a convex Pareto front

$$f_1(x) = x_1$$
$$f_2(x) = g(x)[1 - \sqrt{x_1 / g(x)}]$$
$$g(x) = 1 + 9 \cdot (\sum_{i=2}^{n} x_i)/(n-1)$$

where $n = 30$ and $x_i \in [0, 1]$. The optimal solutions are $x_1 \in [0, 1]$ and $x_i = 0,\ i = 2, \ldots, n$.

- Test problem 5: ZDT2 has a nonconvex Pareto-optimal front

$$
\begin{aligned}
f_1(x) &= x_1 \\
f_2(x) &= g(x)[1 - (x_1 / g(x))^2] \\
g(x) &= 1 + 9(\textstyle\sum_{i=2}^{n} x_i) / (n-1)
\end{aligned}
$$

where $n = 30$ and $x_i \in [0,1]$. The optimal solutions are $x_1 \in [0,1]$ and $x_i = 0,\ i = 2, \dots, n$.

- Test problem 6: ZDT3 has several disconnected Pareto-optimal fronts

$$
\begin{aligned}
f_1(x) &= x_1 \\
f_2(x) &= g(x)[1 - \sqrt{x_1 / g(x)} - \frac{x_1}{g(x)} \sin(10\pi x_1)] \\
g(x) &= 1 + 9(\textstyle\sum_{i=2}^{n} x_i) / (n-1)
\end{aligned}
$$

where $n = 30$ and $x_i \in [0,1]$. The optimal solutions are $x_1 \in [0,1]$ and $x_i = 0,\ i = 2, \dots, n$.

- Test function 7: ZDT4 contains multiple local Pareto-optimal fronts.

$$
\begin{aligned}
f_1(x) &= x_1 \\
f_2(x) &= g(x)\left[1 - \sqrt{x_1 / g(x)}\right] \\
g(x) &= 1 + 10(n-1) + \textstyle\sum_{i=2}^{n} [x_i^2 - 10\cos(4\pi x_i)]
\end{aligned}
$$

where $n = 10$ and $x_1 \in [0,1],\ x_i \in [-5,5],\ i = 2, \dots n$. The optimal solutions

$x_1 \in [0,1]\ x_i = 0,\ i = 2, \dots, n$

- Test problem 8: ZDT6 has non-convex and non-uniformly spaced Pareto-optimal fronts

$$
\begin{aligned}
f_1(x) &= 1 - \exp(-4x_1)\sin^6(6\pi x_1) \\
f_2(x) &= g(x)\left[1 - \left(f_1(x) / g(x)\right)^2\right] \\
g(x) &= 1 + 9\left[(\textstyle\sum_{i=2}^{n} x_i) / (n-1)\right]^{0.25}
\end{aligned}
$$

where $n = 10$ and $x_i \in [0,1]$. The optimal solutions are $x_1 \in [0,1]\ x_i = 0,\ i = 2, \dots, n$.

All objective functions of Test problem 1-8 are to be minimized.

- Test problem 9: We used the problem proposed by Kita (Kita et al, 1996).

$$
\begin{array}{ll}
\text{Maximize} & f_1(x) = -x_1^{\,2} + x_2 \\
 & f_2(x) = \frac{1}{2} x_1 + x_2 + 1 \\
\text{Subject to} & \frac{1}{6} x_1 + x_2 - \frac{13}{2} \le 0 \\
 & \frac{1}{2} x_1 + x_2 - \frac{15}{2} \le 0 \\
 & 5x_1 + x_2 - 30 \le 0
\end{array}
$$

where $x_1, x_2 \in [0,7]$.

4.2. Performance Measures

There are two goals in a multiobjective optimization: 1) convergence to the Pareto-optimal set and 2) diversity of solutions in the Pareto-optimal set. Since these two goals are distinct, we require two different metrics to evaluate the performance of an MOEA.

Convergence metric (γ): This metric finds an average distance between nondominated solutions found and the actual Pareto-optimal front, as follows (Veldhuizen, 1999):

$$\gamma = \frac{\sum_{i=1}^{N} d_i}{N}$$

where N is the number of nondominated solutions obtained with an algorithm and d_i is the Euclidean distance (in objective space) between the each of the nondominated solutions and the nearest member of the actual Pareto optimal front. A smaller value of γ demonstrates a better convergence performance.

Spread (Δ): Deb et al. (2000a) proposed such a metric to measure the spread in solutions obtained by an algorithm. This metric is defined as

$$\Delta = \frac{\sum_{m=1}^{M} d_m^e + \sum_{i=1}^{N-1} \left| d_i - \bar{d} \right|}{\sum_{m=1}^{M} d_m^e + (N-1)\bar{d}}$$

Here, the parameters d_m^e are the Euclidean distance between the extreme solutions of Pareto optimal front and the boundary solutions of the obtained nondominated set corresponding to m^{th} objective function. The parameter d_i is the Euclidean distance between neighboring solutions in the obtained nondominated solutions set and $\bar{d}$ is the mean value of these distances. Δ is zero for an ideal distribution when $d_m^e = 0$ and all d_i equal to $\bar{d}$. Smaller the value of Δ, the better the diversity of the nondominated set.

4.3. Other MOEAs for Comparisons

We compare the proposed MODE with three multiobjective evolutionary algorithms that are representative of the state-of-the-art.

Pareto Archived Evolution Strategy: Knowles and Corne (2000) developed this algorithm. In its simplest form, PAES consists of a $(1+1)$ evolution strategy and maintains an archive of the best solutions previously found. Nondominated solutions are emphasized and placed in the archive. When nondominated solutions compete for a space in the archive, PAES evaluates the crowding in the objective space by dividing the space into hypercubes. Each archived solution is placed in a certain grid location according to its objective values. The solutions located in the most crowded regions are removed when the archive exceeds a pre-defined size.

Nondominated Sorting Genetic Algorithm II: This algorithm is a revised version of the original NSGA (Deb et al. 2002) (Srinivas and Deb, 1994). NSGA-II is a fast and elitist MOEA based on a nondominated sorting approach. This algorithm first combines the parent and offspring populations and uses a nondominated sorting to classify the entire population, then selects the best (with respect to fitness and spread) solutions. With elitism and crowded comparison operator, the NSGA-II is more efficient than the original NSGA.

Multiobjective Particle Swarm Optimization: MOPSO was proposed by Coello et al (2004). MOPSO incorporates Pareto dominance into particle swarm optimization (PSO) in order to handle multiobjective problems. This algorithm uses an external repository of nondominated solutions to guide the particles' future flight during the evolution. At the completion of evolution process, the repository will hold the final nondominated solutions. It also incorporates a special mutation operator to enhance the exploratory capabilities.

3.4. Parameter Settings

In our simulations, all MOEAs are run for a maximum of 25000 fitness function evaluations. MODE uses the following parameter values: population size $NP = 50$, archive size $N_{\max} = 100$, scaling factor $F = 0.3$ and crossover probability $CR = 0.3$. MOPSO uses a population size of 50, a repository size of 100 and 30 divisions for the adaptive grid with mutation as presented in Coello et al. (2004). PAES uses a depth of four and an archive size of 100. For these three approaches, we use all members in the archive after 25000 fitness evaluations to calculate the performance metrics. For NSGA-II (real-coded), we use a population size of 100, crossover probability of 0.9 and mutation probability of $1/n$ (where is the number of decision variables), distribution indexes for crossover and mutation operators as $\eta_c = 20$ and $\eta_m = 20$ as presented in Deb et al (2002). The population obtained at the end of 250 generations is used to calculate the performance metrics. The means and variances presented in Tables 3-5 are obtained by repeatedly running each problem 30 times.

4.4. Discussion of Results

Table 3 shows the means and variances of the convergence metric obtained using the four algorithms MODE, MOPSO, NSGA-II and PAES. MODE is able to converge better than the other three algorithms except in SCH and FON, where PAES and MOPSO yielded better convergence measures. Table 4 shows the mean and variance of the diversity metric obtained using the four algorithms. MODE outperforms the other algorithms in all test problems with respect to the diversity measure. The performance of PAES is the worst in all problems. In the following, we will discuss the performance of the four approaches on each test problem.

Test problem SCH is the simplest among the nine problems with only a single variable. All the four algorithms perform well on this problem, and almost get the same convergence measure. However, MODE and NSGA-II performs better than MOPSO and PAES with respect to diversity on SCH. We show the results of MODE on problem SCH in Figure 2.

The FON is a two-objective optimization problem with three variables. The Pareto optimal front is a single non-convex curve. Figure 3 shows that MODE effectively finds a well spread solution set along the front.

For the remaining seven test problems (KUR, ZDT1, ZDT2, ZDT3, ZDT4, ZDT6 and KITA), we can observe that MODE outperforms the three MOEAs with respect to convergence and diversity except ZDT4, on which none of these algorithms could converge. The KUR problem has three disconnected Pareto-optimal regions, which may cause difficulty in finding nondominated solutions in all regions. MODE performs well as shown in Figure 4, obtaining nondominated solutions in all regions.

Table 3. Comparison of MODE, MOPSO, NSGA-II and PAES based on Convergence Metric (γ)

(mean in 1st rows and variance in 2nd rows , the best mean result is emphasized in **boldface**.)

Algorithm	SCH	FON	KUR	ZDT1	ZDT2	ZDT3	ZDT6	KITA
MODE	0.006502	0.003031	**0.030819**	**0.001999**	**0.001554**	**0.002642**	**0.005998**	**0.048741**
	3.79E-07	4.47E-08	8.42E-06	2.23E-08	2.75E-08	5.47E-08	4.53E-06	2.92E-03
MOPSO	0.006508	**0.002165**	0.035290	0.018682	0.012671	0.033759	0.255940	0.079239
	3.86E-07	2.88E-08	9.39E-06	2.05E-06	3.27E-05	4.88E-05	9.09E-02	6.12E-03
NSGA-II	0.006536	0.003038	0.060342	0.070086	0.192318	0.640421	3.120325	0.203116
	4.72E-07	4.55E-08	1.83E-02	6.81E-04	4.37E-03	1.36E-03	1.30E-01	7.56E-02
PAES	**0.006451**	0.050884	1.200243	0.005238	0.001611	0.074629	7.600697	0.180029
	3.44E-07	2.22E-02	4.77E+00	2.98E-05	7.50E-07	1.27E-05	7.42E-01	5.97E-02

Table 4. Comparison of MODE, MOPSO, NSGA-II and PAES based on Diversity Metric (Δ)

(mean in 1st rows and variance in 2nd rows , the best mean result is emphasized in **boldface**.)

Algorithm	SCH	FON	KUR	ZDT1	ZDT2	ZDT3	ZDT6	KITA
MODE	0.347156	**0.220099**	**0.401911**	**0.306235**	**0.298449**	**0.504275**	**0.335594**	**0.555137**
	1.16E-03	3.93E-04	5.48E-04	1.13E-03	5.80E-04	2.00E-04	1.90E-02	3.09E-02
MOPSO	0.587726	0.603447	0.625978	0.594930	0.767260	0.604377	0.975030	0.655123
	2.74E-03	2.54E-03	1.86E-03	1.35E-03	4.21E-02	1.53E-03	9.65E-04	2.71E-03
NSGA-II	**0.281940**	0.448451	0.768280	0.545618	0.938612	0.797206	0.963956	0.972649
	7.84E-04	1.39E-03	2.99E-03	2.18E-03	1.99E-02	2.56E-04	4.75E-04	2.89E-03
PAES	0.643277	0.767554	0.708826	1.111025	1.138642	0.580257	0.826468	1.384021
	2.27E-03	7.77E-03	2.02E-03	3.39E-02	5.20E-02	1.47E-03	1.04E-02	8.59E-02

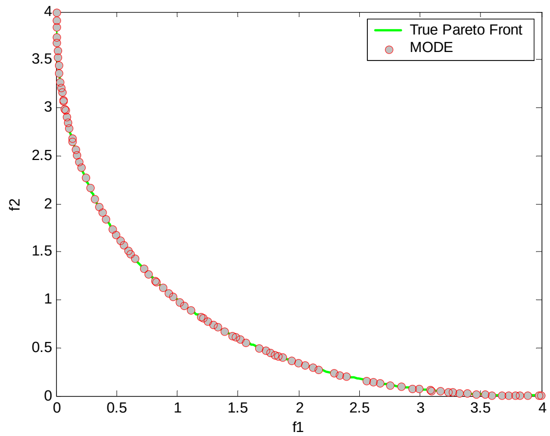

Figure 2. Pareto front with MODE on SCH.

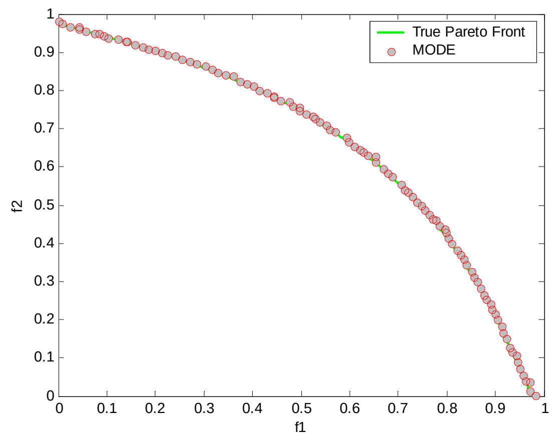

Figure 3. Pareto front with MODE on FON.

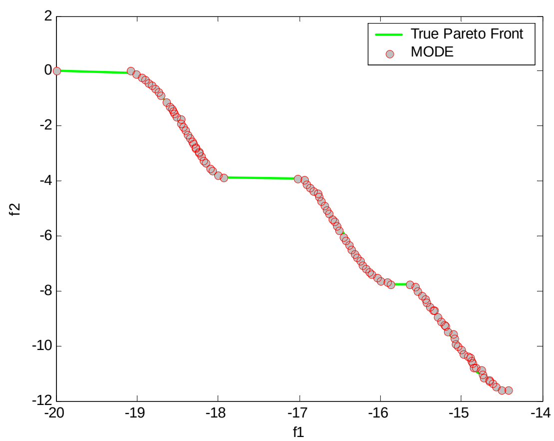

Figure 4. Pareto front with MODE on KUR.

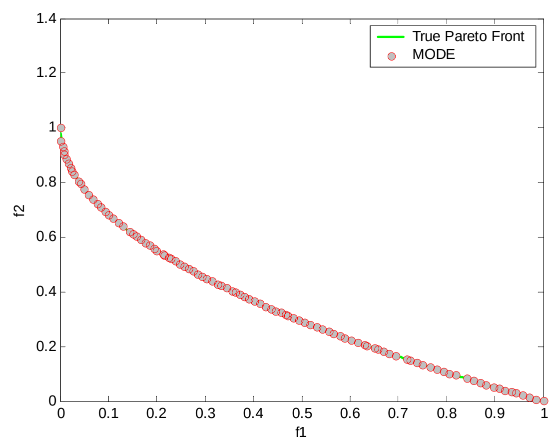

Figure 5. Pareto front with MODE on ZDT1.

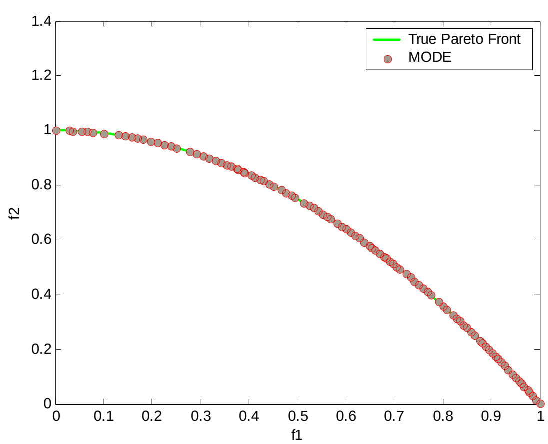

Figure 6. Pareto front with MODE on ZDT2.

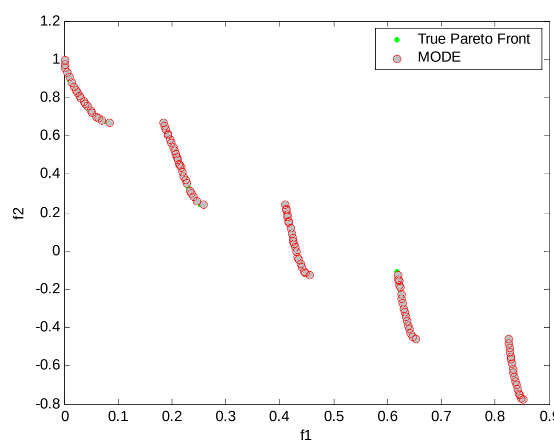

Figure 7. Pareto front with MODE on ZDT3.

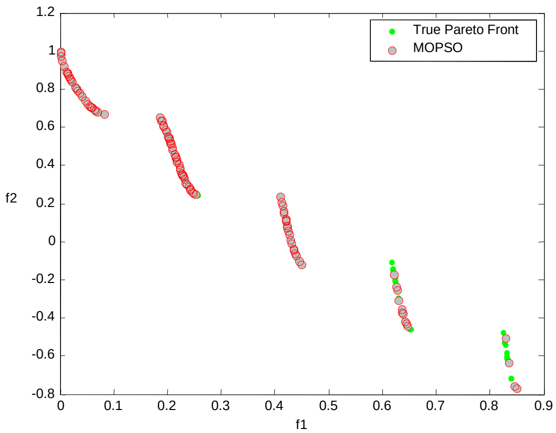

Figure 8. Pareto front with MOPSO on ZDT3.

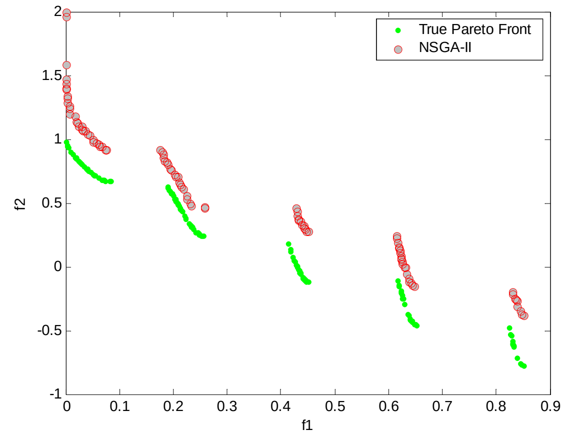

Figure 9. Pareto front with NSGA-II on ZDT3.

ZDT1 is probably the easiest of all of the ZDT problems, the only difficulty an MOEA may face in this problem is the large number of variables. MODE, MOPSO and NSGA-II all converged to the Pareto optimal front with good spread over the entire front. However, the convergence metric of MODE is much smaller than the other as shown in Table 3, which demonstrates a superior convergence ability of the proposed MODE.

Nondominated solutions obtained in MODE on ZDT2 are shown in Figure 6. Although it was not difficult for both MOPSO and NSGA-II to converge to the front, MODE found a better spread with a smaller convergence metric than the others.

The Pareto optimal front of ZDT3 is made up of five disjoint curves. Large values of γ and Δ obtained by NSGA-II in Table 3 demonstrate that this approach could not converge to the Pareto optimal front with diverse distributions, which is also shown in Figure 9. Although the front obtained in MOPSO, shown in Figure 8, almost converges to the true front, it could not perform as well as MODE, which produces the nondominated solutions well converged and spread out over the entire front, as shown in Figure 7.

The problem ZDT4 has 21^9 different local Pareto-optimal fronts in the search space, each with respect to $x_1 \in [0,1]$ and $x_i = 0.5m$, where m is any integer in $[-10,10]$, $i = 2, 3, \ldots, 10$. Among these fronts, only one is the global Pareto-optimal front, which corresponds to $x_1 \in [0,1]$ and $x_i = 0$ (Deb, 2001). Because of the hurdles due to the large number of local Pareto-optimal fronts, none of these approaches could converge to the global front. Hence, we did not incorporate the performance metric on problem ZDT4 in Tables 3 and 4. However, NSGA-II used a different distribution index $\eta_m = 10$ to obtain the global Pareto-optimal front on ZDT4 (Deb et al. 2002). Here, we use MODE with a smaller crossover probability $CR = 0.1$, and keep all the other parameters the same as defined in Section 4.4. A smaller crossover probability means less $c_{j,i,G+1}$ equal to $v_{j,i,G+1}$ (in Equation (6)), so that the solutions will be less affected by the $P_{best,G}$ which could avoid the solutions getting trapped in local Pareto-optimal fronts. In this way, MODE successfully finds the solutions on the global Pareto-optimal front as shown in Figure 10. We compare the convergence and diversity metrics of MODE and NSGA-II on ZDT4 in Table 5, which demonstrates that MODE performs much better than NSGA-II especially with respect to converge metric.

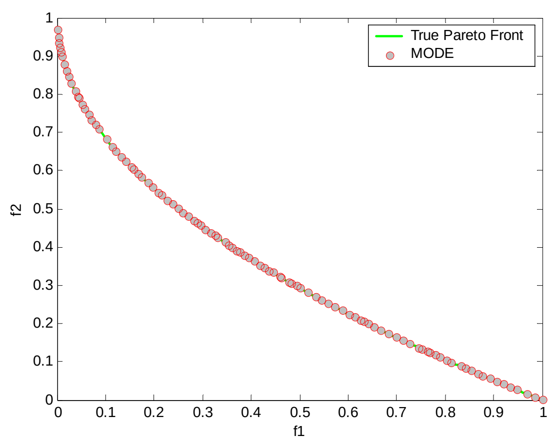

Figure 10. Pareto front with MODE on ZDT4 ($CR = 0.1$).

Table 5. Performance Comparison of MODE and NSGA-II on ZDT4

Algorithm	Converge metric γ		Diversity metric Δ	
	Mean	Variance	Mean	Variance
MODE	**0.030689**	0.004867	**0.338330**	0.003676
NSGA-II	0.039853	0.005275	0.540475	0.003487

The problem ZDT6 is another hard problem. The adverse density of solutions across the Pareto-optimal front, together with the non-convex nature of the front, makes it difficult for many multiobjective optimization algorithms to maintain a well-distributed nondominated set and converge to the true Pareto-optimal front. From table 3, we could observe that MOPSO, NSGA-II and PAES could not converge to the true Pareto front of ZDT6. However, the performances of MOPSO and NSGA-II are obviously better than PAES. Figure 12 shows that NSGA-II faces difficulties in converging to the Pareto-optimal front, while MODE performs well in converging to the true front with a good spread of solutions along the front as presented in Figure 11. As shown in Tables 3 and 4, the average values of γ and Δ obtained by MODE on problem ZDT6 are much better than the corresponding parameter values obtained by the other algorithms.

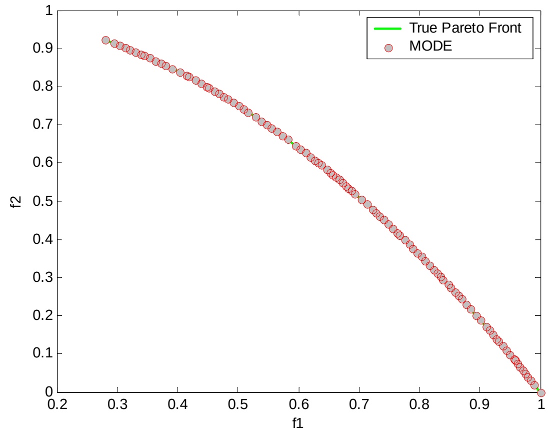

Figure.11. Pareto front with MODE on ZDT6.

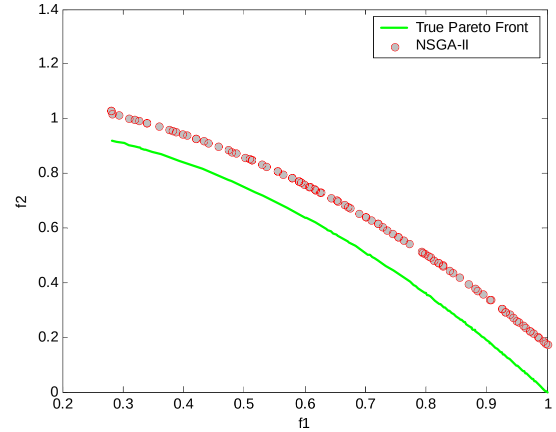

Figure.12. Pareto front with NSGA-II on ZDT6.

The ninth test problem KITA is a constrained problem that we choose to verify the performance of MODE on constrained problem. The good Pareto front obtained by MODE on KITA problem (shown in Figure 13) demonstrates that MODE could deal with constrained multiobjective problem well. Again the superiority of MODE to find solutions approximated to the true Pareto front is validated with respect to the smallest convergence and diversity metrics.

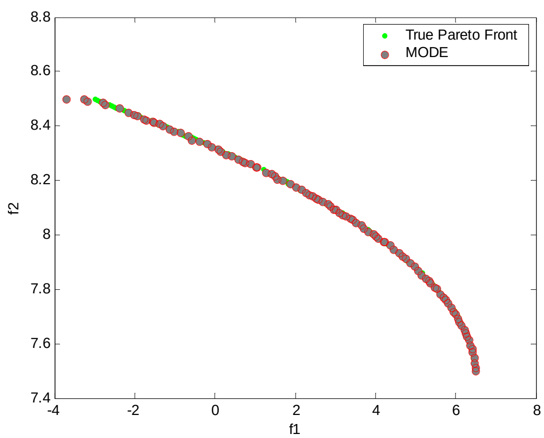

Figure.13. Pareto front with MODE on KITA.

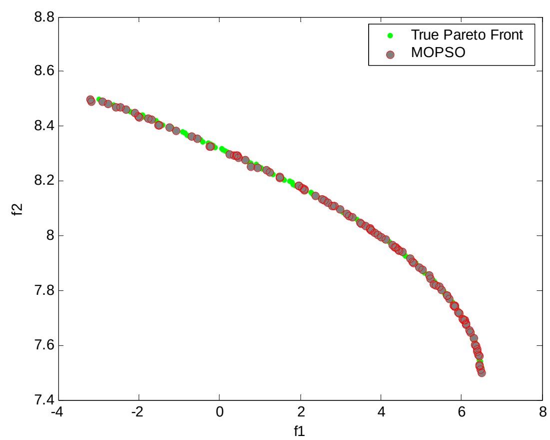

Figure.14. Pareto front with MOPSO on KITA.

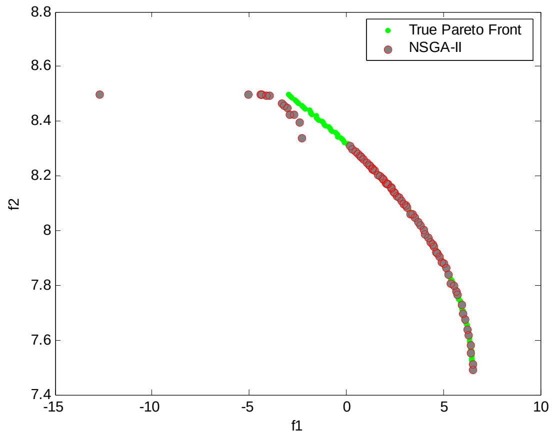

Figure.15. Pareto front with MOPSO on KITA.

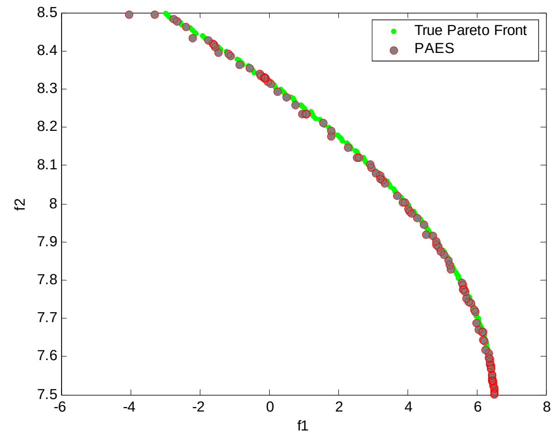

Figure.16. Pareto front with PAES on KITA.

5. Improvement

Choosing a good crowding measure method is very important to accurately estimate the crowding degree around one solution, otherwise, the diversity of the nondominated solution obtained will be bad. Unfortunately, there is a case that crowding distance that we used in MODE may not correctly estimate the crowding degree.

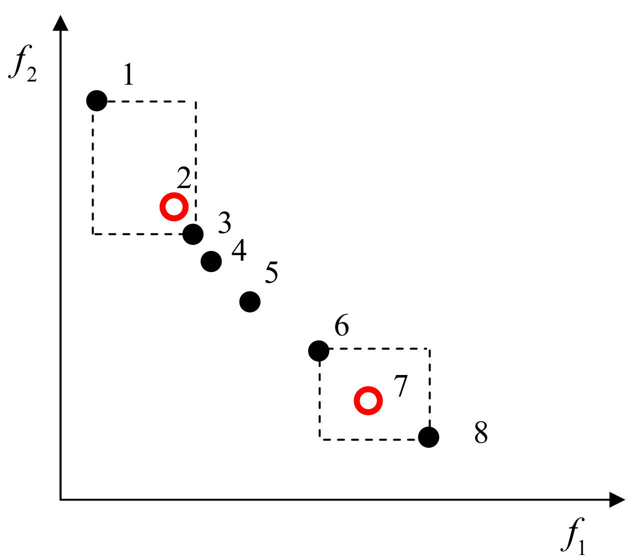

Figure 17 Crowding degree estimation

In Figure 17, we could observe that solution 2 is crowded than solution 7, while the crowding distance of solution 2 is larger than solution 7 since solution 1 is far away from solution 2. This outlier (solution 1) affects accurate measurement of crowding degree.

Therefore, we employ *k*-nearest neighbor approach to estimate the density around one solution in the objective function space. Here, the harmonic average of all *k*-nearest neighbors around one solution is adopted instead of the arithmetic average or just take the distance from the solution to its k^{th} nearest neighbor. For example, assuming *k*-nearest neighbor distances around one solution is $d_1, d_2, d_3, \ldots d_k$, the harmonic average is: $d = \frac{1}{\frac{1}{d_1} + \frac{1}{d_2} + \ldots \frac{1}{d_k}}$, even if one distance among $d_1, d_2, d_3, \ldots d_k$ is very large and other solutions are small, the harmonic average d is still small. In this way, the possible surrounding outlier influence to the solutions may be avoided.

Table 6. Truncate the archive solutions by harmonic average

Step 1: Calculate the distance matrix using all the solutions in the external archive.

Step 2: Sort the distance matrix along row

Step 3: Take the harmonic average along the first *k* rows positive value to form a row vector

Step 4: The index corresponding to the smallest element of the vector is obtained.

Step 5: Remove this point according to its obtained index in the original solution set, and set all the distances corresponding to this solution in the distance matrix calculated in step 1 to be -1.

Step 6: Repeat step 3 until the predefined number of solutions is left.

The complexity of this procedure is mainly decided by sorting the distance matrix along row in step 2, which has $O(\bar{N}^2 \log \bar{N})$ computational complexity, where $\bar{N}$ is number of the solution in the external archive. The computational complexity is $O(N^2 \log N)$ when N and $\bar{N}$ are of the same order. This crowding measure method can substitute for the crowding distance of the original MODE, and we call this algorithm MODE-II.

To demonstrate this new harmonic average method, we compared the MODE-II to the original MODE, $k = 3$, all the other related parameters are same as described in section 4.4. The results are presented in Table 7 and 8.

Table 7. Comparison of MODE and MODE-II based on Convergence Metric (γ)

(mean in 1^{st} rows and variance in 2^{nd} rows , the best mean result is emphasized in **boldface**.)

Algorithm	SCH	FON	KUR	ZDT1	ZDT2	ZDT3	ZDT6	KITA
MODE	**0.006502**	0.003031	0.030819	**0.001999**	**0.001554**	0.002642	0.005998	0.048741
	3.79E-07	4.47E-08	8.42E-06	2.23E-08	2.75E-08	5.47E-08	4.53E-06	2.92E-03
MODE-II	0.006562	**0.002899**	**0.030398**	0.002106	0.001565	**0.002621**	**0.005356**	**0.048203**
	4.17E-07	2.65E-08	3.77E-06	1.32E-08	7.85E-09	2.83E-08	2.62E-08	5.22E-03

Table 8. Comparison of MODE and MODE-II on Diversity Metric (Δ)

(mean in 1^{st} rows and variance in 2^{nd} rows , the best mean result is emphasized in **boldface**.)

Algorithm	SCH	FON	KUR	ZDT1	ZDT2	ZDT3	ZDT6	KITA
MODE	0.347156	0.220099	0.401911	0.306235	0.298449	0.504275	0.335594	0.555137
	1.16E-03	3.93E-04	5.48E-04	1.13E-03	5.80E-04	2.00E-04	1.90E-02	3.09E-02
MODE-II	**0.134487**	**0.146656**	**0.338024**	**0.122807**	**0.122300**	**0.436701**	**0.104131**	**0.532897**
	1.33E-04	1.65E-04	5.13E-04	1.02E-04	1.13E-04	3.50E-05	1.35E-04	3.16E-02

From the results in Table 7 and 8, we find that the MODE-II obviously obtains better diversity than original MODE. It demonstrates that the harmonic average based crowding degree estimation method performs better than crowding distance in maintaining distribution of the nondominated solutions. However, we have to mention that the convergence metrics of SCH, ZDT1 and ZDT2 obtained by MODE-II are a little worse than the results obtained by the original MODE.

6. Conclusions

This paper presented a novel proposal to extend the DE to tackle multi-objective optimization problems with an external archive. We evaluated the proposed approach on nine test problems currently adopted in the literature. The proposed MODE significantly outperforms other three representative multiobjective evolutionary algorithms, obviously on larger dimensional problems. It also demonstrates a good performance when dealing with a multimodal problem (ZDT4).

Although NSGA-II has no external archive, it combines the parent and offspring populations, which has the same effect as external archive to avoid missing the nondominated solutions. MODE, MOPSO and PAES all incorporate external archive. To make good diversity of the solution, MODE and NSGA-II use crowding distance. PAES and MOPSO use adaptive grids. But MODE performs best among these four algorithms, which demonstrates that the mutation and crossover operator of DE is as effective as in single-objective optimization when dealing with multiobjective optimization problem. Combining the robust and effective DE with crowding distance based archive maintenance strategy can yield a simple and effective multiobjective evolutionary algorithm capable of converging to the true Pareto optimal front and maintaining good diversity along the Pareto front.

However, the crowding distance could not always accurately measure the crowding degree of the solution. The proposed harmonic average can estimate the crowding degree more precisely and guarantee a much better diversity of the nondominated solution, without deteriorating the convergence probability of the algorithm.

At present, the proposed MODE is only tested on two objectives problem. It is expected to verify these new algorithms for higher objective space dimensions. Another aspect we would like to explore in the future is to extend this algorithm to rotated problem.

Reference:

Abbass, H. A., Sarker, R. and Newton, C. (2001). "PDE: A Pareto-frontier Differential Evolution Approach for Multiobjective Optimization Problems." In *Proc. of IEEE Congress on Evolutionary Computation*, pp. 971-978.

Abbass, H. A. (2002). "The Self-Adaptive Pareto Differential Evolution Algorithm." In *Proc. of Congress on Evolutionary Computation* Vol.1, IEEE Press, 831-836.

Babu, B.V. and Jehan, M. M. L.(2003). "Differential Evolution for Multi-Objective Optimization," In *Proceedings of the 2003 Congress on Evolutionary Computation (CEC'2003)*, Canberra, Australia. Vol 4, pp. 2696--2703, IEEE Press.

Chang, C. S. and Xu, D. Y. (2000). "Differential Evolution of Fuzzy Automatic Train Operation for Mass Rapid Transit System." In *IEEE Proc. of Electric Power Applications* Vol. 147, No. 3 206-212.

Chang, C.S., Xu, D.Y. and Quek, H.B. (1999). "Pareto-optimal set based multiobjective tuning of fuzzy automatic train operation for mass transit system." In *IEE Proceedings on Electric Power Applications,* 146(5):577–583, September 1999.

Coello C.A.C, Pulido G.T and Lechuga M.S. (2004). "Handling multiple objectives with particle swarm optimization." *IEEE Trans. on Evolutionary Computation*. 8(3): 256-279.

Deb K. (2001). "Multiobjective Optimization Using Evolutionary Algorithms." Chischester, U.K.:Wiley.

Deb K., Pratap A., Agarwal,S. and Meyarivan T. (2002). "A fast and elitist multi-objective genetic algorithms: NSGA-II." *IEEE Trans. on Evolutionary Computation*. 6(2): 182-197.

Fonseca, C.M. and Fleming, P. J. (1995). "An overview of evolutionary algorithms in multi-objective optimization," *Evolutionary Computation Journal* 3(1), 1-16.

Fonseca C.M. and Fleming, P. J. (1998). "Multiobjective optimization and multiple constraint handling with evolutionary algorithms-Part II: Application example." *IEEE Trans. Syst., man, Cybern. A,* vol.28, pp.38-47, Jan.1998.

Ilonen, J., Kamarainen, J.-K. and Lampinen, J. (2003). "Differential Evolution Training Algorithm for Feed-Forward Neural Networks." In: *Neural Processing Letters* Vol. 7, No. 1 93-105.

Iorio, A. and Li, X. (2004). "Solving Rotated Multi-objective Optimization Problems Using Differential Evolution." In *Proceeding of the 17th Joint Australian Conference on Artificial Intelligence, Lecture Notes in Computer Science, Lecture Notes in Computer Science (LNCS 3339)*, eds. G.I. Webb and Xinghuo Yu, p.861-872.

Knowles J. and Corne, D. (1999)"The Pareto archived evolution strategy: A new baseline algorithm for Pareto multiobjective optimisation," in *Proceedings of the 1999 Congress on Evolutionary Computation*. Piscataway, NJ: IEEE Press, 1999, pp. 98–105.

Knowles, J.D. and Corne, D.W. (2000). "Approximating the nondominated front using the Pareto archive evolutionary strategy." *Evolutionary Computation*, 8(2): 149-172.

Kita,H., Yabumoto, Y. , Mori,N. and Nishikawa, Y.(1996). "Multi-objective optimization by means of thermodynamical genetic algorithm." In *PPSN IV*, H.-M. Voigt, W.Ebeling, I.Rechenberg, and H.-P.Schwefel, Eds. Berlin, Germany; Springer-Verlag, Sept. 1996, LNCS, pp.504-512.

Kokkonen S. and Lampinen J. (2004a), "Mechanical Component Design for Multiple Objectives Using Generalized Differential Evolution," in I.C. Parmee (editor), *Adaptive Computing in Design and Manufacture VI*, pp. 261--272, Springer, London.

Kokkonen S. and Lampinen J. (2004b). "An Extension of Generalized Differential Evolution for Multi-objective Optimization with Constraints," in Xin Yao et al. (editors), *Parallel Problem Solving from Nature - PPSN VIII*, Springer-Verlag, Lecture Notes in Computer Science, Vol. 3242, pp. 752--761, Birmingham, UK.

Kursawe, F. (1991) "A variant of evolution strategies for vector optimization," in *Parallel Problem Solving From Nature*, H. P. Schwefel and R. Männer, Eds. Berlin, Germany: Springer-Verlag,1990 PPSN I, pp. 193–197.

Laumanns M., Zitzler E. and Thiele L. (2000). "A unified model for multiobjective evolutionary algorithms with Elitism." In *Proc. of Congress on Evolutionary Computation*. 46-53.

Madavan, N. K. (2002). “Multiobjective Optimization Using a Pareto Differential Evolution Approach.” In: *Proc. of Congress on Evolutionary Computation (CEC'2002)* Vol. 2, IEEE Press, 1145-1150.

Parsopoulos, K.E., Taoulis, D.K., Pavlidis, N.G., Plagianakos V.P. and Vrahatis M.N. (2004). “Vector Evaluated Differential Evolution for Multiobjective Optimization,” In *2004 Congress on Evolutionary Computation (CEC'2004)*, IEEE Service Center, Vol. 1, pp. 204--211, Portland, Oregon, USA, June 2004

Price, K. V. (1996). “Differential evolution: a fast and simple numerical optimizer.” In: Smith, M., Lee, M., Keller, J., Yen., J. (eds.): *Biennial Conference of the North American Fuzzy Information Processing Society, NAFIPS*. IEEE Press, New York (1996) 524-527.

Price, K. V. (1999). “An Introduction to Differential Evolution.” In: Corne, D., Dorigo, M.,and Glover, F. (eds.): *New Ideas in Optimization*. McGraw-Hill, London (UK) 79-108.

Rogalsky, T., Derksen, R.W. and Kocabiyik, S (1999). “Differential Evolution in Aerodynamic Optimization.” In: *Proc. of 46th Annual Conf of Canadian Aeronautics and Space Institute*. 29-36.

Schaffer, J. D. (1985). “Multiple objective optimization with vector evaluated genetic algorithms.” In *Proceedings of the First International Conference on Genetic Algorithms*, J. J. Grefensttete, Ed. Hillsdale, NJ: Lawrence Erlbaum, 1985, pp.93-100.

Srinivas N. and Deb K. (1994). “Multiobjective optimization using nondominated sorting in genetic algorithms.” *Evol. Comput.*, vol. 2, no. 3, pp. 221–248.

Storn, R., Price, K. (1995). “Differential Evolution - a simple and efficient adaptive scheme for global optimization over continuous spaces,” Technical Report TR-95-012, ICSI.1995.

Storn, R. (1996). “Differential evolution design of an IIR-filter.” In: *Proceedings of IEEE Int. Conference on Evolutionary Computation ICEC'96*. IEEE Press, New York 268-273.

Storn, R. and Price, K. (1997). “Differential evolution-A simple and Efficient Heuristic for Global Optimization over Continuous Spaces.” *Journal of Global Optimization* 11:341-359.

Xue, F. (2003). “Multi-Objective Differential Evolution and its Application to Enterprise Planning.” In: *Proc. of 2003 IEEE Int. Conf. on Robotics and Autom ation (ICRA'03)* Vol. 3, IEEE Press, 3535-3541.

Xue, F., Sanderson, A. C. and Graves, R. J. (2003). “Pareto-based Multi-objective Differential Evolution. In: *Proc. of the 2003 Congress on Evolutionary Computation(CEC'2003)* Vol. 2, IEEE Press, 862-869.

Xue, F., Sanderson, A. C., Bonissone, P. and Graves, R. J. (2005). “Fuzzy Logic Controlled Multi-Objective Differential Evolution", *Proc. FUZZ-IEEE 2005,* Reno NV, USA, May 22-25.

Zitzler, E. and Thiele L. (1999). “Multi-objective evolutionary algorithms: a comparative case study and the strength Pareto approach.” *IEEE Trans. on Evolutionary Computation*, 3(4): 257-271.

Zitzler, E. and Deb K. and Thiele L. (2001). “Comparison of multiobjective evolutionary algorithms: Empirical study.” *Evolutionary Computation* 8(2): 173-195.

Zitzler, E., Laumanns, M. and Thiele L. (2001). “SPEA2: Improving the strength Pareto evolutionary algorithm.” *Swiss Federal Institute of Technology*, Lausanne, Switzerland, Tech. Rep. TIK-Rep, 103.