A Multiobjective Differential Evolution Based on Decomposition for Multiobjective Optimization with Variable Linkages

Hui Li and Qingfu Zhang

Department of Computer Science, University of Essex
Wivenhoe Park, Colchester CO4 3SQ, United Kingdom
{hli1, qzhang}@essex.ac.uk

Abstract. Although a number of multiobjective evolutionary algorithms have been proposed over the last two decades, not much effort has been made to deal with variable linkages in multiobjective optimization. Recently, we have suggested a general framework of multiobjective evolutionary algorithms based on decomposition (MOEA/D) [1]. MOEA/D decomposes a MOP into a number of scalar optimization subproblems by a conventional decomposition method. The optimal solution to each of these problems is a Pareto optimal solution to the MOP under consideration. An appropriate decomposition could make these individual Pareto solutions evenly distribute along the Pareto optimal front. MOEA/D aims at solving these scalar optimization subproblems simultaneously. In this paper, we propose, under the framework of MOEA/D, a multiobjective differential evolution based decomposition (MODE/D) for tackling variable linkages. Our experimental results show that MODE/D outperforms several other MOEAs on several test problems with variable linkages.

1 Introduction

In many real-world applications, there are several objectives to be optimized. Often, these objectives conflict with each other. There is no single solution that can optimize all objectives at the same time. Pareto optimal solutions are optimal trade-offs among the objectives. Over the past two decades, a number of multiobjective evolutionary algorithms (MOEAs) have been proposed [2][3]. MOEAs are able to find a set of representative Pareto optimal solutions in a single run.

Like their counterparts for scalar optimization, most MOEAs employ a selection operator to direct their search into promising areas in the decision space. Since Pareto domination is not a complete ordering, conventional selection operators, which were originally developed for scalar optimization, cannot be directly applied to multiobjective optimization. Furthermore, the task of most MOEAs is to produce a set of solutions which are uniformly distributed in the Pareto front. A selection operator in MOEAs should not encourage the search to converge to a single point. Therefore, it is not a trivial job to assign a relative fitness

T.P. Runarsson et al. (Eds.): PPSN IX, LNCS 4193, pp. 583–592, 2006.

value to each individual solution for reflecting its utility in selection in MOEAs. Fitness assignment has been subject to much research over the last two decades [2]. Some techniques such as fitness sharing and crowding have been frequently used within fitness assignment for maintaining the diversity of search [4][5].

Most existing MOEAs treat the MOP under consideration as a whole. They use Pareto domination for ranking solutions during their search. Their selection operators are often very complicated and time-consuming, and it is hard to balance the diversity and convergence. Note that a Pareto optimal solution for a MOP, under mild condition, could be an optimal solution of a scalar optimization problem in which the objective is an aggregation of all the f_i's. Therefore, approximation of the Pareto front of a MOP can be decomposed into a number of scalar objective optimization subproblems. In [1], we proposed a general multiobjective evolutionary algorithm based on decomposition (MOEA/D). MOEA/D explicitly decomposes a MOP into N scalar optimization subproblems. Each scalar optimization subproblem has its best solution found so far in the current population. Each subproblem is optimized in MOEA/D by using information from its neighboring subproblems. It has been proved that MOEA/D has a lower complexity than NSGA-II, the most popular MOEA, at each generation.

Although variable linkages exist in many applications, not much effort has been devoted to how to deal with variable linkages in MOEAs. As shown in [6], recombination operators play a key role in tackling variable linkages. In this paper, under the framework of MOEA/D, we design a multiobjective differential evolution based on decomposition (MODE/D) for continuous MOPs with variable linkages. MODE/D maintains a population which contains the best solution found so far to each of the subproblems. The DE operator is used for generating new trail solutions. We define a neighborhood relationship among all the subproblems such that neighboring subproblems have similar optimal solutions. Consequently we obtain a neighborhood relationship among all the individual solutions in the current population. The DE recombination operator is restricted to neighboring solutions since otherwise it may generate poor solutions for MOPs with variable linkages.

The remainder of this paper is organized as follows. Section II introduces basic definitions in MOPs. The proposed algorithm is then described in Section III. Section IV presents multiobjective test problems with variable linkages. Experimental results are given in Section V. The final section concludes the paper.

2 Multiobjective Optimization

We consider a multiobjective optimization problem of the form:

$$\begin{aligned} \text{minimize } & F(x) = (f_1(x), f_2(x), \ldots, f_m(x))^T \\ \text{s.t. } & x \in X \end{aligned} \tag{1}$$

where $x = (x_1, \ldots, x_n)$ is called decision vector, $X \subset R^n$ is the decision space, $f_i : R^n \to R, i = 1, \ldots, m$ $(m \geq 2)$ are objective functions. $F(x)$ is the objective vector. The objective space Y consists of the set of all objective vectors. The optimal solutions of (1) can be defined in terms of Pareto optimality [7].

Let $a = (a_1, \ldots, a_m)^T$, $b = (b_1, \ldots, b_m)^T \in R^m$ be two vectors, a is said to *dominate* b, denoted by $a \prec b$, if $a_i \leq b_i$ for all $i = 1, \ldots, n$, and $a \neq b$. A point $x^\star \in X$ is called *(globally) Pareto optimal* if there is no $x \in X$ such that $F(x) \prec F(x^\star)$. The set of all Pareto optimal points, denoted by PS, is called the *Pareto set*. The set of all Pareto objective vectors, $PF = \{y \in R^m | y = F(x), x \in PS\}$, is called the *Pareto front* [7].

3 Algorithm

MODE/D is an MOEA/D which uses DE operator to generate new solutions. In our implementation of MODE/D in this paper, the weighted Tchebycheff approach is used to decompose the MOP (1). In this approach, the scalar optimization problem is in the form

$$g^{te}(x|\lambda, z^*) = \max_{i \in \{1,\ldots,m\}} \lambda_i \, |f_i^* - f_i(\mathbf{x})| \tag{2}$$

where $\lambda = (\lambda_1, \ldots, \lambda_m)$ is the scalar weight vector,

$$\sum_{i=1}^{m} \lambda_i = 1$$

and $\lambda_i \geq 0$ for all $i = 1, \ldots, m$. $z^* = \{f_1^*, \ldots, f_m^*\}$ is the ideal point, i.e.,

$$f_i^* = \min\{f_i(x) | x \in X\}$$

for each $i = 1, 2, \ldots, m$.

Under some mild conditions, for each Pareto optimal solution x^* there exists a weight vector λ such that x^* is the optimal solution of (2), and each optimal solution of (2) is a Pareto optimal solution of (1).

MODE/D first selects N evenly distributed weight vectors $\lambda^1, \lambda^2, \ldots, \lambda^N$. Then the MOP (1) is decomposed into N optimization subproblems, where the objective in j-th subproblem is $g^{te}(x|\lambda^j, z^*)$. MODE/D aims at minimizing these subproblems in a single run. Note that g^{te} is continuous of λ, the optimal solution of $g^{te}(x|\lambda^i, z^*)$ should be close to that of $g^{te}(x|\lambda^j, z^*)$ if λ^i and λ^j are close to each other. Therefore, any information about these g^{te}'s with weight vectors close to λ^i should be helpful for optimizing $g^{te}(x|\lambda^i, z^*)$.

At each iteration, MODE/D maintains:

- a population of N points $x^1, \ldots, x^N \in X$, where x^i is the current solution to the i-th subproblem;
- a population of objective vectors $FV^1, \ldots, FV^N$, where FV^i is the F-value of x^i, i.e., $FV^i = F(x^i)$;
- a temporary reference point $z = (z_1, \ldots, z_m)^T$, where z_i is the smallest value found so far for objective f_i;

MODE/D works as follows:

Input:
- MOP (1);
- a stopping criterion;
- N : the number of the subproblems considered in MOEA/D;
- an uniform spread of N weight vectors: $\lambda^1, \ldots, \lambda^N$;
- K : the number of the weight vectors in the neighborhood of each weight vector;

Output: $x^1, \ldots, x^N$ and $FV^1, \ldots, FV^N$.

Step 1 Initialization

Step 1.1 For each $i = 1, \ldots, N$, set $B(i) = \{i_1, \ldots, i_K\}$ where $\lambda^{i_1}, \ldots, \lambda^{i_K}$ are the K closet weight vectors to λ^i.

Step 1.2 Randomly generate an initial population $x^1, \ldots, x^N$. Set $FV^i = F(x^i)$;

Step 1.3 For each $j = 1, \ldots, m$, $z_j = \min_{i \in \{1,\ldots,N\}} f_j(x^i)$.

Step 2 Reproduction Randomly choose a solution x^r and then randomly select three indexes a, b, c from $B(r)$. A new solution $y = (y_1, \ldots, y_n)^T$ is generated in the following way.

For each $i = 1, \ldots, n$

$$y_i = \begin{cases} x_i^a + R \cdot (x_i^b - x_i^c) & \text{if rand} < CR \\ x_i^r, & \text{otherwise} \end{cases} \tag{3}$$

where R and CR are two control parameters.

Step 3 Update of Reference Point z**:** For each $j = 1, \ldots, m$, if $f_j(y) < z_j$, then set $z_j = f_j(y)$.

Step 4 Update of Neighboring Solutions: For each $j \in B(r)$, if $g^{te}(y|\lambda^j, z) \leq g^{te}(x^j|\lambda^j, z)$, then set $x^j = y$ and $FV^j = F(y)$.

Step 5 Stopping Criteria If stopping criteria are satisfied, then stop and output $x^1, \ldots, x^N$ and $FV^1, \ldots, FV^N$. Otherwise go to **Step 2**.

In **step 1**, the neighborhood $B(i)$ is defined as the set of the indexes of K nearest weight vectors to λ^i. The initial population consists of N solutions randomly chosen from the decision space. The ideal point $z^* = (f_1^*, \ldots, f_m^*)$ in (2) is replaced with a reference point $z = (z_1, \ldots, z_m)$. In **step 2**, four neighboring solutions x^r, x^a, x^b, x^b undergo the DE operator for producing a new trail solution y. The reference vector z is updated by y in **step 3**. In **step 4**, the neighboring solutions of x^r are replaced by y if they are worse than y with respect to its associated g^{te}.

4 Multiobjective Test Problems with Variable Linkage

Various features of MOPs might cause difficulties for MOEAs, such as non-convexity, multi-modality, discontinuity, and non-uniformality. Deb et al [8][9] constructed a number of test problems using the following model:

$$\begin{aligned}\text{Minimize } & F(x) = \{f_1(x_1), f_2(x)\} \\ \text{s.t. } & f_2(x) = g(x_2, \dots, x_n)h(f_1(x_1), g(x_2, \dots, x_n))\end{aligned} \tag{4}$$

where $f_1 : R \rightarrow R$, $g : R^{n-1} \rightarrow R$, and $h : R^n \rightarrow R$ are three real functions. The main features of test problems of (4) can be controlled by setting f_1, g, and h with specific properties.

As Deb et al noticed [6], several widely-used test instances induced from the above models don't not have variable linkages. More precisely, the *PS*s of these test instances are linear and parallel to coordinate axes. Therefore, it is relatively easy for MOEAs to find *PS*.

Recently, some researchers have studied the multiobjective test problems with variable linkages [6] [10][11]. In this paper, two OKA test instances [10] are used.

– OKA-1

$$\begin{aligned}f_1(x) &= x_1 \\ f_2(x) &= \pi - x_1 + |x_2 - 5\cos(x_1)|\end{aligned} \tag{5}$$

where $x_1 \in [-\pi, \pi]$ and $x_2 \in [-5, 5]$. The *PS* of OKA-1 is $\{(x_1, x_2)|x_2 = 5\cos(x_1), x_1 \in [-\pi, \pi]\}$ and the *PF* is $\{(f_1, f_2)|f_2 = \pi - f_1, f_1 \in [-\pi, \pi]\}$

– OKA-2

$$\begin{aligned}f_1(x) &= \eta(x_1) \\ f_2(x) &= \pi - \eta(x_1) + |x_2 - 5\cos(x_1)| \\ \eta(x_1) &= \begin{cases} x_1^{\frac{1}{3}} & \text{if } x_1 \geq 0 \\ -x_1^{\frac{1}{3}} & \text{if } x_1 < 0 \end{cases}\end{aligned} \tag{6}$$

where $x_1 \in [-\pi^3, \pi^3]$ and $x_2 \in [-5, 5]$. The *PS* of OKA-1 is $\{(x_1, x_2)|x_2 = 5\cos(x_1), x_1 \in [-\pi, \pi]\}$ and the *PF* is $\{(f_1, f_2)|f_2 = \pi - f_1, f_1 \in [-\pi, \pi]\}$

Inspired by the construction of OKA test instances, we propose the following two variants of ZDT1 and ZDT2:

– ZDT1-L

$$\begin{aligned}f_1(x) &= x_1 \\ g(x) &= 1 + \frac{1}{n-1}\sum_{i=2}^{n} |x_1 - sin(0.5x_i\pi))| \\ h(x) &= 1 - \sqrt{f_1/g}\end{aligned}$$

where $x \in [0, 1]^n$. The *PS* of ZDT1-L is $\{x|x_1 = sin(0.5x_i\pi), i = 2, \dots, n\}$.

– ZDT2-L

$$\begin{aligned}f_1(x) &= x_1 \\ g(x) &= 1 + \frac{1}{n-1}\sum_{i=2}^{n} |x_1 - sin(0.5x_i\pi)| \\ h(x) &= 1 - (f_1/g)^2\end{aligned}$$

where $x \in [0, 1]^n$. The *PS* of ZDT2-L is $\{x|x_1 = sin(0.5x_i\pi), i = 2, \dots, n\}$.

There are nonlinear variable linkages in ZDT1-L and ZDT2-L. Obviously, the *PF*s of ZDT1-L and ZDT2-L are the same as those of ZDT1 and ZDT2 in the objective space.

In [6], Deb et al proposed using linear or nonlinear variable transformation to construct test instances with variable linkages. Their approach is more complicated than ours.

5 Experimental Results

5.1 Performance Metrics

In MOEAs, various performance metrics can be used to measure convergence and diversity [12]. In our experiments, we use set coverage ($\mathcal{C}$-metric) and generational distance ($\mathcal{D}$-metric) to assess the performance of the algorithms.

Let A and B be two approximations to the PF of a MOP, $C(A, B)$ is defined as the percentage of the solutions in B that are dominated by at least one solution in A, i.e.,

$$C(A, B) = \frac{|\{u \in B | \exists v \in A : v \text{ dominates } u\}|}{|B|}$$

$C(A, B)$ is not necessarily equal to $1 - C(B, A)$. $C(A, B) = 1$ means that all solutions in B are dominated by some solutions in A, while $C(A, B) = 0$ implies that no solution in B is dominated by a solution in A.

Veldhuizen proposed a distance-based metric, called generational distance, as follows.

$$\mathcal{D}_p(A, P^*) = \frac{1}{|A|} \left(\sum_{i=1}^{|A|} d_i(a_i, P^*)^p \right)^{1/p} \tag{7}$$

where P^* is the reference set of representative Pareto optimal solutions and $d_i(a_i, P^*) = \min_{r \in P^*} \left\{ \sqrt{\sum_{k=1}^{m} (f_k(a_i) - f_k(r))^2} \right\}$. $\mathcal{D}_1$ represents the average distance from A to P^* when $p = 1$. $\mathcal{D}_1$ only measures the closeness between A and P^*.

We also can calculate the average Euclidean distance from P^* to A, denoted by $\mathcal{D}_2 = \mathcal{D}_1(P^*, A)$.

In our experiments, uniformly distribution points in PF are selected to form the reference set P^*.

5.2 Experimental Settings

We compared MODE/D with NSGA-II/SBX [11], NSGA-II/DE [13], and GDE3 [14] in our experiments.

The population size is set to be 100 for all the algorithms. In NSGA-II/SBX, The distribution index used in SBX and polynomial mutation is set to be 20, the mutation rate is set to be $1.0/n$, where n is the number of decision variables.

The DE operator is the same in NSGA-II/DE, GDE3 and MODE/D. R is set to be 0.5 and CR is 0.95. The size K of neighborhood in MODE/D is set to be 20 for all test instances.

For OKA-1 and OKA-2, all algorithms are run for 25,000 function evaluations. For ZDT1-L and ZDT2-L, n is set to be 10. We allow all the algorithms 50,000 function evaluations. All algorithms were implemented in C++. Each algorithm has been independently run for 20 times for each test instance on PC (Pentium (R) 2.4GHZ, 1.00 GB of RAM).

5.3 Results

Table 1 and 2 show the best, mean, and standard deviation of the $\mathcal{D}$-metric values between the obtained solutions and reference set P^*. In terms of the best and mean of $\mathcal{D}_1$-metric values in Table 1, MODE/D outperforms other three algorithms on OKA-1, OKA-2 and ZDT1-L. It is also evident that both GDE3 and NSGA-II/SBX can find a set of solutions with zero distance to the reference set. This is because GDE3 and NSGA-II/SBX reach the *PF*. Table 2 shows the best, mean and standard deviation of the $\mathcal{D}_2$-metric values from the reference set to the obtained solutions. In terms of this metric, MODE/D performs better than other three algorithms on all test problems. This also suggests that the diversity of the solutions found by MODE/D is better than those in other three algorithms.

Table 1. $\mathcal{D}_1$-metric values of four algorithms

D1-metric	MODE/D			GDE3			NSGA-II/DE			NSGA-II/SBX		
Test Problems	best	mean	std	best	mean	std	best	mean	std	best	mean	std
OKA-1	0.0051	0.0058	0.0004	0.0138	0.0166	0.0034	0.0195	0.0263	0.0052	0.0185	0.0581	0.0877
OKA-2	0.0090	0.0128	0.0034	0.0141	0.0175	0.0019	0.0177	0.0358	0.0202	0.0162	0.0355	0.0272
ZDT1-L	0.0028	0.0036	0.0007	0.0083	0.0099	0.0010	0.0072	0.0090	0.0009	0.0093	0.0118	0.0024
ZDT2-L	0.0033	0.0050	0.0009	0	0.0049	0.0100	0.0114	0.0161	0.0056	0	0.0669	0.1292

Table 2. $\mathcal{D}_2$-metric values of four algorithms

D2-metric	MODE/D			GDE3			NSGA-II/DE			NSGA-II/SBX		
Test Problems	best	mean	std	best	mean	std	best	mean	std	best	mean	std
OKA-1	0.0229	0.0231	0.0002	0.0283	0.0298	0.0009	0.0373	0.0421	0.0021	0.1070	0.1959	0.0976
OKA-2	0.0262	0.0295	0.0017	0.0299	0.0335	0.0022	0.0429	0.0520	0.0069	0.1019	0.1734	0.0363
ZDT1-L	0.0054	0.0195	0.0124	0.0099	0.0116	0.0009	0.0099	0.0113	0.0011	0.2769	0.7322	0.1353
ZDT2-L	0.0065	0.0178	0.0156	0.0331	0.5005	0.2241	0.0166	0.0243	0.0053	0.1034	0.3090	0.2054

Figure 1 illustrates the distributions of the nondominated solutions in the run with the lowest D-metric value. It is clear from Figure 1 that MODE/D performs better than other algorithms in terms of maintaining diversity. Intuitionally, NSGA-II/SBX is outperformed by other three DE-based MOEAs in both convergence and diversity for all test instances.

The box plot of $\mathcal{C}$-metric values between the MOEAs considered is visualized in Figure 2. As we can see, MODE/D performs slightly better than GDE3 on

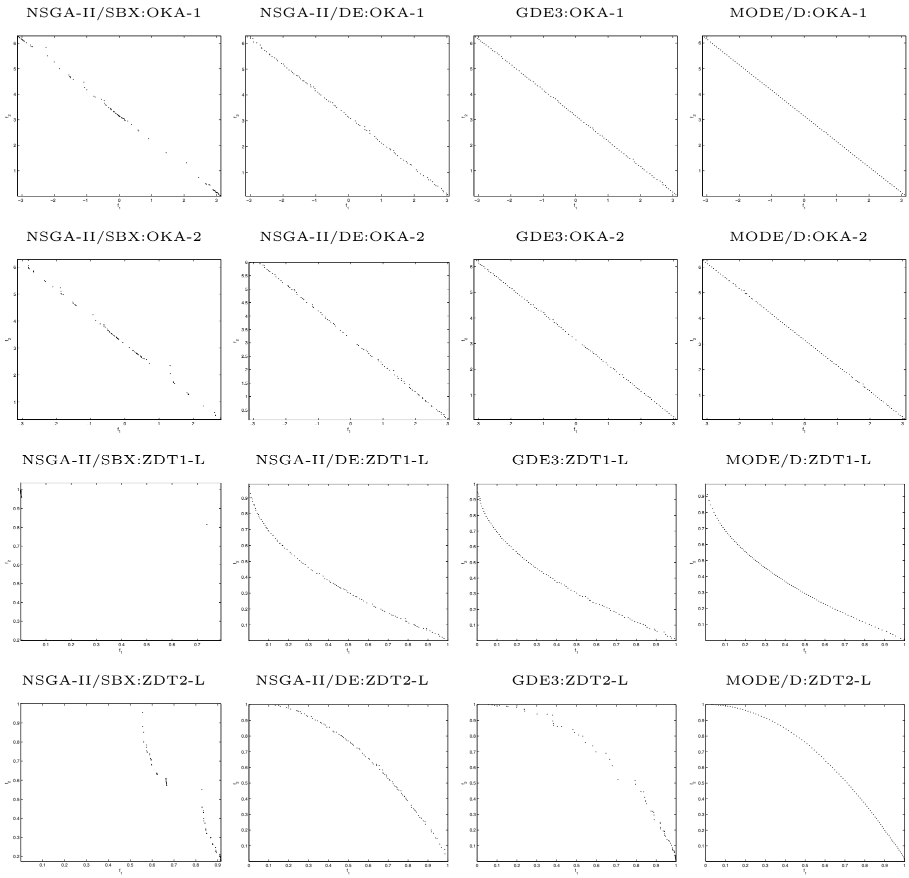

Fig. 1. Plots of the nondominated solutions in the run with the lowest D-values found by NSGA-II/SBX, NSGA-II/DE, GDE3, and MODE/D

OKA-1 and OKA-2. For ZDT1-L and ZDT2-L, MODE/D outperforms the other three MOEAs. In terms of $\mathcal{C}$-metric, NSGA-II/SBX is outperformed by the other three DE-based MOEAs. Therefore, a reproduction operator plays a key role in MOEAs for dealing with variable linkages.

6 Conclusions

In this paper, we proposed a multiobjective differential evolution approach based on decomposition (MODE/D) for MOPs with linkage. MODE/D is a MOEA/D with a DE operator. The experimental results show that, overall, MODE/D clearly outperforms NSGA-II/SBX, NSGA-II/DE, and GDE3. Our results suggest that MOEA/D is a promising method for solving MOPs. It is also clear that one needs to consider reproduction operators for tackling variable linkages

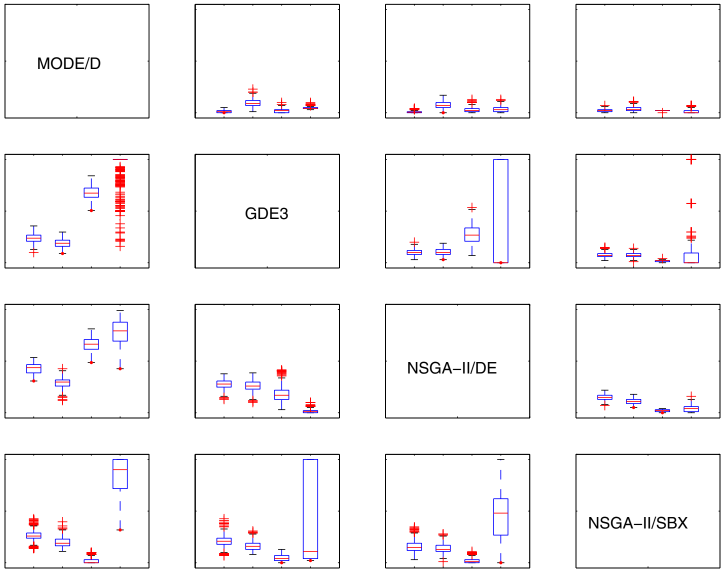

Fig. 2. The box plots of $\mathcal{C}$-metric of MODE/D, GDE3, NSGA-II/DE, and NSGA-II/SBX. Four box plots from left to right in each chart relate to OKA-1, OKA-2, ZDT1-L, and ZDT-L, respectively.

in MOPs. Future work includes study of the effect of parameters and schemes for adaptively adjusting weight vectors in MODE/D.

References

1. Zhang, Q., Li, H.: A multiobjective evolutionary algorithm based on decomposition. Technical Report CSM-450, Department of Computer Science, University of Essex (2006)
2. Deb, K., Kalyanmoy, D.: Multi-Objective Optimization Using Evolutionary Algorithms. John Wiley & Sons, Inc. (2001)
3. Coello, C.A.C.: Recent trends in evolutionary multiobjective optimization. In: Evolutionary Multiobjective Optimization: Theoretical Advances And Applications. Springer-Verlag, London (2005) 7–32
4. Bosman, P.A.N., Thierens, D.: The balance between proximity and diversity in multiobjective evolutionary algorithms. IEEE Trans. Evolutionary Computation **7**(2) (2003) 174–188
5. Knowles, J.D., Corne, D.: Properties of an adaptive archiving algorithm for storing nondominated vectors. IEEE Trans. Evolutionary Computation **7**(2) (2003) 100–116
6. Deb, K., Sinha, A., Kukkonen, S.: Multi-objective test problems, linkages, and evolutionary methodologies. Technical Report KanGAL Report No. 2006001, KanGAL, Indian Institute of Technology Kanpur (2006)

7. Miettinen, K.: Nonlinear Multiobjective Optimization. Kluwer Academic Publishers (1999)
8. Deb, K.: Multi-objective genetic algorithms: Problem difficulties and construction of test problems. Evolutionary Computation **7**(3) (1999) 205–230
9. Zitzler, E., Deb, K., Thiele, L.: Comparison of multiobjective evolutionary algorithms: Empirical results. Evolutionary Computation **8**(2) (2000) 173–195
10. Okabe, T., Jin, Y., Olhofer, M., Sendhoff, B.: On test functions for evolutionary multi-objective optimization. In: PPSN'2004. Lecture Notes in Computer Science, Springer (2004) 792–802
11. Deb, K., Agrawal, S., Pratap, A., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. Evolutionary Computation **6**(2) (2002) 182–197
12. Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C.M., da Fonseca, V.G.: Performance assessment of multiobjective optimizers: an analysis and review. IEEE Trans. Evolutionary Computation **7**(2) (2003) 117–132
13. Iorio, A., Li, X.: Solving rotated multi-objective optimization problems using differential evolution. In: Australian Conference on Artificial Intelligence. Lecture Notes in Computer Science, Cairns, Australia, Springer (2004) 861–872
14. Kukkonen, S., Lampinen, J.: GDE3: The third evolution step of generalized differential evolution. In: CEC'2005, Edinburgh, Scotland, IEEE Service Center (2005) 443–450