DEMO: Differential Evolution for Multiobjective Optimization

Tea Robič and Bogdan Filipič

Department of Intelligent Systems, Jožef Stefan Institute,
Jamova 39, SI-1000 Ljubljana, Slovenia
tea.robic@ijs.si
bogdan.filipic@ijs.si

Abstract. Differential Evolution (DE) is a simple but powerful evolutionary optimization algorithm with many successful applications. In this paper we propose Differential Evolution for Multiobjective Optimization (DEMO) – a new approach to multiobjective optimization based on DE. DEMO combines the advantages of DE with the mechanisms of Pareto-based ranking and crowding distance sorting, used by state-of-the-art evolutionary algorithms for multiobjective optimization. DEMO is implemented in three variants that achieve competitive results on five ZDT test problems.

1 Introduction

Many real-world optimization problems involve optimization of several (conflicting) criteria. Since multiobjective optimization searches for an optimal vector, not just a single value, one solution often cannot be said to be better than another and there exists not only a single optimal solution, but a set of optimal solutions, called the *Pareto front*. Consequently, there are two goals in multiobjective optimization: (i) to discover solutions as close to the Pareto front as possible, and (ii) to find solutions as diverse as possible in the obtained nondominated front. Satisfying these two goals is a challenging task for any algorithm for multiobjective optimization.

In recent years, many algorithms for multiobjective optimization have been introduced. Most originate in the field of Evolutionary Algorithms (EAs) – the so-called Multiobjective Optimization EAs (MOEAs). Among these, the NSGA-II by Deb et al. [1] and SPEA2 by Zitzler et al. [2] are the most popular. MOEAs take the strong points of EAs and apply them to Multiobjective Optimization Problems (MOPs). A particular EA that has been used for multiobjective optimization is Differential Evolution (DE). DE is a simple yet powerful evolutionary algorithm by Price and Storn [3] that has been successfully used in solving single-objective optimization problems [4]. Hence, several researchers have tried to extend it to handle MOPs.

Abbass [5, 6] was the first to apply DE to MOPs in the so-called Pareto Differential Evolution (PDE) algorithm. This approach employs DE to create

new individuals and then keeps only the nondominated ones as the basis for the next generation. PDE was compared to SPEA [7] (the predecessor of SPEA2) on two test problems and found to outperform it.

Madavan [8] achieved good results with the Pareto Differential Evolution Approach (PDEA[1]). Like PDE, PDEA applies DE to create new individuals. It then combines both populations and calculates the nondominated rank (with Pareto-based ranking assignment) and diversity rank (with the crowding distance metric) for each individual. Two variants of PDEA were investigated. The first compares each child with its parent. The child replaced the parent if it had a higher nondominated rank or, if it had the same nondominated rank and a higher diversity rank. Otherwise the child is discarded. This variant was found inefficient – the diversity was good but the convergence slow. The other variant simply takes the best individuals according to the nondominated rank and diversity rank (like in NSGA-II). The latter variant has proved to be very efficient and was applied to several MOPs, where it produced favorable results.

Xue [9] introduced Multiobjective Differential Evolution (MODE). This algorithm also uses the Pareto-based ranking assignment and the crowding distance metric, but in a different manner than PDEA. In MODE the fitness of an individual is first calculated using Pareto-based ranking and then reduced with respect to the individual's crowding distance value. This single fitness value is then used to select the best individuals for the new population. MODE was tested on five benchmark problems where it produced better results than SPEA.

In this paper, we propose a new way of extending DE to be suitable for solving MOPs. We call it DEMO (Differential Evolution for Multiobjective Optimization). Although similar to the existing algorithms (especially PDEA), our implementation differs from others and represents a novel approach to multiobjective optimization. DEMO is implemented in three variants (DEMO/parent, DEMO/closest/dec and DEMO/closest/obj). Because of diverse recommendations for the crossover probability, three different values for this parameter are investigated. From the simulation results on five test problems we find that DEMO efficiently achieves the two goals of multiobjective optimization, i.e. the convergence to the true Pareto front and uniform spread of individuals along the front. Moreover, DEMO achieves very good results on the test problem ZDT4 that poses many difficulties to state-of-the-art algorithms for multiobjective optimization.

The rest of the paper is organized as follows. In Section 2 we describe the DE scheme that was used as a base for DEMO. Thereafter, in Section 3, we present DEMO in its three variants. Section 4 outlines the applied test problems and performance measures, and states the results. Further comparison and discussion of the results are provided in Section 5. The paper concludes with Section 6.

---

[1] This acronym was not used by Madavan. We introduce it to make clear distinction between his approach and other implementations of DE for multiobjective optimization.

Differential Evolution
1. Evaluate the initial population $\mathcal{P}$ of random individuals.
2. While stopping criterion not met, do:
  2.1. For each individual P_i ($i = 1, \ldots, popSize$) from $\mathcal{P}$ repeat:
    (a) Create candidate C from parent P_i.
    (b) Evaluate the candidate.
    (c) If the candidate is better than the parent, the candidate replaces
       the parent. Otherwise, the candidate is discarded.
  2.2. Randomly enumerate the individuals in $\mathcal{P}$.

Fig. 1. Outline of DE's main procedure

Candidate creation
Input: Parent P_i
1. Randomly select three individuals $P_{i_1}, P_{i_2}, P_{i_3}$ from $\mathcal{P}$,
  where i, i_1, i_2 and i_3 are pairwise different.
2. Calculate candidate C as $C = P_{i_1} + F \cdot (P_{i_2} - P_{i_3})$,
  where F is a scaling factor.
3. Modify the candidate by binary crossover with the parent
  using crossover probability *crossProb*.
Output: Candidate C

Fig. 2. Outline of the candidate creation in scheme *DE/rand/1/bin*

2   Differential Evolution

DE is a simple evolutionary algorithm that creates new candidate solutions by combining the parent individual and several other individuals of the same population. A candidate replaces the parent only if it has better fitness. This is a rather greedy selection scheme that often outperforms traditional EAs.

The DE algorithm in pseudo-code is shown in Fig. 1. Many variants of creation of a candidate are possible. We use the DE scheme DE/rand/1/bin described in Fig. 2 (more details on this and other DE schemes can be found in [10]). Sometimes, the newly created candidate falls out of bounds of the variable space. In such cases, many approaches of constraint handling are possible. We address this problem by simply replacing the candidate value violating the boundary constraints with the closest boundary value. In this way, the candidate becomes feasible by making as few alterations to it as possible. Moreover, this approach does not require the construction of a new candidate.

3   Differential Evolution for Multiobjective Optimization

When applying DE to MOPs, we face many difficulties. Besides preserving a uniformly spread front of nondominated solutions, which is a challenging task for any MOEA, we have to deal with another question, that is, when to replace the parent with the candidate solution. In single-objective optimization, the decision

Differential Evolution for Multiobjective Optimization

1. Evaluate the initial population $\mathcal{P}$ of random individuals.
2. While stopping criterion not met, do:
   2.1. For each individual P_i ($i = 1, \ldots, popSize$) from $\mathcal{P}$ repeat:
      (a) Create candidate C from parent P_i.
      (b) Evaluate the candidate.
      (c) If the candidate dominates the parent, the candidate replaces the parent.
         If the parent dominates the candidate, the candidate is discarded.
         Otherwise, the candidate is added in the population.
   2.2. If the population has more than $popSize$ individuals, truncate it.
   2.3. Randomly enumerate the individuals in $\mathcal{P}$.

Fig. 3. Outline of DEMO/parent

is easy: the candidate replaces the parent only when the candidate is better than the parent. In MOPs, on the other hand, the decision is not so straightforward. We could use the concept of dominance (the candidate replaces the parent only if it dominates it), but this would make the greedy selection scheme of DE even greedier. Therefore, DEMO applies the following principle (see Fig. 3). The candidate replaces the parent if it dominates it. If the parent dominates the candidate, the candidate is discarded. Otherwise (when the candidate and parent are nondominated with regard to each other), the candidate is added to the population. This step is repeated until $popSize$ number of candidates are created. After that, we get a population of the size between $popSize$ and $2 \cdot popSize$. If the population has enlarged, we have to truncate it to prepare it for the next step of the algorithm.

The truncation consists of sorting the individuals with nondominated sorting and then evaluating the individuals of the same front with the crowding distance metric. The truncation procedure keeps in the population only the best $popSize$ individuals (with regard to these two metrics). The described truncation is derived from NSGA-II and is also used in PDEA's second variant.

DEMO incorporates two crucial mechanisms. The immediate replacement of the parent individual with the candidate that dominates it, is the core of DEMO. The newly created candidates that enter the population (either by replacement or by addition) instantly take part in the creation of the following candidates. This emphasizes elitism within reproduction, which helps achieving the first goal of multiobjective optimization – convergence to the true Pareto front. The second mechanism is the use of nondominated sorting and crowding distance metric in truncation of the extended population. Besides preserving elitism, this mechanism stimulates the uniform spread of solutions. This is needed to achieve the second goal – finding as diverse nondominated solutions as possible. DEMO's selection scheme thus efficiently pursues both goals of multiobjective optimization.

The described DEMO's procedure (outlined in Fig. 3) is the most elementary of the three variants presented in this paper. It is called DEMO/parent.

The other two variants were inspired by the concept of Crowding DE, recently introduced by Thomsen in [11].

Thomsen applied crowding-based DE in optimization of multimodal functions. When optimizing functions with many optima, we would sometimes like not only to find one optimal point, but also discover and maintain multiple optima in a single algorithm run. For this purpose, Crowding DE can be used. Crowding DE is basically conventional DE with one important difference. Usually, the candidate is compared to its parent. In Crowding DE, the candidate is compared to the most similar individual in the population. The applied similarity measure is the Euclidean distance between two solutions.

Crowding DE was tested on numerous benchmark problems and its performance was impressive. Because the goal of maintaining multiple optima is similar to the second goal of multiobjective optimization (maintaining diverse solutions in the front), we implemented the idea of Crowding DE in two additional variants of the DEMO algorithm. The second, DEMO/closest/dec, works in the same way as DEMO/parent, with the exception that the candidate solution is compared to the most similar individual in decision space. If it dominates it, the candidate replaces this individual, otherwise it is treated in the same way as in DEMO/parent. The applied similarity measure is the Euclidean distance between the two solutions in decision space. In the third variant, DEMO/closest/obj, the candidate is compared to the most similar individual in objective space.

DEMO/closest/dec and DEMO/closest/obj need more time for one step of the procedure than DEMO/parent. This is because at every step they have to search for the most similar individual in the decision and objective space, respectively. Although the additional computational complexity is notable when operating on high-dimensional spaces, it is negligible in real-world problems where the solution evaluation is the most time consuming task.

4   Evaluation and Results

4.1   Test Problems

We analyze the performance of the three variants of DEMO on five ZDT test problems (introduced in [12]) that were frequently used as benchmark problems in the literature [1, 8, 9]. These problems are described in detail in Tables 1 and 2.

4.2   Performance Measures

Different performance measures for evaluating the efficiency of MOEAs have been suggested in the literature. Because we wanted to compare DEMO to other MOEAs (especially the ones that use DE) on their published results, we use three metrics that have been used in these studies.

For all 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

Table 1. Description of the test problems ZDT1, ZDT2 and ZDT3

ZDT1	
Decision space	$\mathbf{x} \in [0, 1]^{30}$
Objective functions	$f_1(\mathbf{x}) = x_1$
	$f_2(\mathbf{x}) = g(\mathbf{x}) \left(1 - \sqrt{x_1/g(\mathbf{x})} \right)$
	$g(\mathbf{x}) = 1 + \frac{9}{n-1} \sum_{i=2}^{n} x_i$
Optimal solutions	$0 \le \mathbf{x_1^*} \le 1$ and $\mathbf{x_i^*} = 0$ for $i = 2, \ldots, 30$
Characteristics	convex Pareto front

ZDT2	
Decision space	$\mathbf{x} \in [0, 1]^{30}$
Objective functions	$f_1(\mathbf{x}) = x_1$
	$f_2(\mathbf{x}) = g(\mathbf{x}) \left(1 - (x_1/g(\mathbf{x}))^2 \right)$
	$g(\mathbf{x}) = 1 + \frac{9}{n-1} \sum_{i=2}^{n} x_i$
Optimal solutions	$0 \le \mathbf{x_1^*} \le 1$ and $\mathbf{x_i^*} = 0$ for $i = 2, \ldots, 30$
Characteristics	nonconvex Pareto front

ZDT3	
Decision space	$\mathbf{x} \in [0, 1]^{30}$
Objective functions	$f_1(\mathbf{x}) = x_1$
	$f_2(\mathbf{x}) = g(\mathbf{x}) \left(1 - \sqrt{x_1/g(\mathbf{x})} - \frac{x_1}{g(\mathbf{x})} \sin(10\pi x_1) \right)$
	$g(\mathbf{x}) = 1 + \frac{9}{n-1} \sum_{i=2}^{n} x_i$
Optimal solutions	$0 \le \mathbf{x_1^*} \le 1$ and $\mathbf{x_i^*} = 0$ for $i = 2, \ldots, 30$
Characteristics	discontinuous Pareto front

hard to obtain. In our experiments we use 500 uniformly spaced Pareto-optimal solutions as the approximation of the true Pareto front[2].

The first metric is *Convergence metric* Υ. It measures the distance between the obtained nondominated front Q and the set P^* of Pareto-optimal solutions:

$$\Upsilon = \frac{\sum_{i=1}^{|Q|} d_i}{|Q|},$$

where d_i is the Euclidean distance (in the objective space) between the solution $i \in Q$ and the nearest member of P^*.

Instead of the convergence metric, some researchers use a very similar metric, called *Generational Distance GD*. This metric measures the distance between

---

[2] These solutions are uniformly spread in the decision space. They were made available online by Simon Huband at http://www.scis.ecu.edu.au/research/wfg/datafiles.html.

Table 2. Description of the test problems ZDT4 and ZDT6

ZDT4	
Decision space	$\mathbf{x} \in [0,1] \times [-5,5]^9$
Objective functions	$f_1(\mathbf{x}) = x_1$
	$f_2(\mathbf{x}) = g(\mathbf{x})\left(1 - (x_i/g(\mathbf{x}))^2\right)$
	$g(\mathbf{x}) = 1 + 10(n-1) + \sum_{i=2}^{n}\left(x_i^2 - 10\cos(4\pi x_i)\right)$
Optimal solutions	$0 \le \mathbf{x_1^*} \le 1$ and $\mathbf{x_i^*} = 0$ for $i = 2, \ldots, 10$
Characteristics	many local Pareto fronts

ZDT6	
Decision space	$\mathbf{x} \in [0,1]^{10}$
Objective functions	$f_1(\mathbf{x}) = 1 - \exp^{-4x_1}\sin(6\pi x_1)^6$
	$f_2(\mathbf{x}) = g(\mathbf{x})\left(1 - (f_1(\mathbf{x})/g(\mathbf{x}))^2\right)$
	$g(\mathbf{x}) = 1 + \frac{9}{n-1}\sum_{i=2}^{n} x_i$
Optimal solutions	$0 \le \mathbf{x_1^*} \le 1$ and $\mathbf{x_i^*} = 0$ for $i = 2, \ldots, 10$
Characteristics	low density of solutions near Pareto front

the obtained nondominated front Q and the set P^* of Pareto-optimal solutions as

$$GD = \frac{\sqrt{\sum_{i=1}^{|Q|} d_i^2}}{|Q|},$$

where d_i is again the Euclidean distance (in the objective space) between the solution $i \in Q$ and the nearest member of P^*.

The third metric is *Diversity metric* Δ. This metric measures the extent of spread achieved among the nondominated solutions:

$$\Delta = \frac{d_f + d_l + \sum_{i=1}^{|Q|-1} |d_i - \overline{d}|}{d_f + d_l + (|Q|-1)\overline{d}},$$

where d_i is the Euclidean distance (in the objective space) between consecutive solutions in the obtained nondominated front Q, and $\overline{d}$ is the average of these distances. The parameters d_f and d_l represent the Euclidean distances between the extreme solutions of the Pareto front P^* and the boundary solutions of the obtained front Q.

4.3   Experiments

In addition to investigating the performance of the three DEMO variants, we were also interested in observing the effect of the crossover probability (*crossProb* in Fig. 2) on the efficiency of DEMO. Therefore, we made the following experiments: for every DEMO variant, we run the respective DEMO algorithm with crossover probabilities set to 30%, 60% and 90%. We repeated all tests 10 times

Table 3. Statistics of the results on test problems ZDT1, ZDT2 and ZDT3

Algorithm	ZDT1	
	Convergence metric	Diversity metric
NSGA-II (real-coded)	0.033482±0.004750	0.390307±0.001876
NSGA-II (binary-coded)	0.000894±0.000000	0.463292±0.041622
SPEA	0.001799±0.000001	0.784525±0.004440
PAES	0.082085±0.008679	1.229794±0.004839
PDEA	N/A	0.298567±0.000742
MODE	0.005800±0.000000	N/A
DEMO/parent	0.001083±0.000113	0.325237±0.030249
DEMO/closest/dec	0.001113±0.000134	0.319230±0.031350
DEMO/closest/obj	0.001132±0.000136	0.306770±0.025465

Algorithm	ZDT2	
	Convergence metric	Diversity metric
NSGA-II (real-coded)	0.072391±0.031689	0.430776±0.004721
NSGA-II (binary-coded)	0.000824±0.000000	0.435112±0.024607
SPEA	0.001339±0.000000	0.755148±0.004521
PAES	0.126276±0.036877	1.165942±0.007682
PDEA	N/A	0.317958±0.001389
MODE	0.005500±0.000000	N/A
DEMO/parent	0.000755±0.000045	0.329151±0.032408
DEMO/closest/dec	0.000820±0.000042	0.335178±0.016985
DEMO/closest/obj	0.000780±0.000035	0.326821±0.021083

Algorithm	ZDT3	
	Convergence metric	Diversity metric
NSGA-II (real-coded)	0.114500±0.007940	0.738540±0.019706
NSGA-II (binary-coded)	0.043411±0.000042	0.575606±0.005078
SPEA	0.047517±0.000047	0.672938±0.003587
PAES	0.023872±0.000010	0.789920±0.001653
PDEA	N/A	0.623812±0.000225
MODE	0.021560±0.000000	N/A
DEMO/parent	0.001178±0.000059	0.309436±0.018603
DEMO/closest/dec	0.001197±0.000091	0.324934±0.029648
DEMO/closest/obj	0.001236±0.000091	0.328873±0.019142

with different initial populations. The scaling factor F was set to 0.5 and was not tuned. To match the settings of the algorithms used for comparison, the population size was set 100 and the algorithm was run for 250 generations.

4.4   Results

Tables 3 and 4 present the mean and variance of the values of the convergence and diversity metric, averaged over 10 runs. We provide the results for all three DEMO variants. Results of other algorithms are taken from the literature (see

Table 4. Statistics of the results on test problems ZDT4 and ZDT6

Algorithm	ZDT4	
	Convergence metric	Diversity metric
NSGA-II (real-coded)	0.513053±0.118460	0.702612±0.064648
NSGA-II (binary-coded)	3.227636±7.307630	0.479475±0.009841
SPEA	7.340299±6.572516	0.798463±0.014616
PAES	0.854816±0.527238	0.870458±0.101399
PDEA	N/A	0.840852±0.035741
MODE	0.638950±0.500200	N/A
DEMO/parent	0.001037±0.000134	0.359905±0.037672
DEMO/closest/dec	0.001016±0.000091	0.359600±0.026977
DEMO/closest/obj	0.041012±0.063920	0.407225±0.094851

| Algorithm	ZDT6	
	Convergence metric	Diversity metric
NSGA-II (real-coded)	0.296564±0.013135	0.668025±0.009923
NSGA-II (binary-coded)	7.806798±0.001667	0.644477±0.035042
SPEA	0.221138±0.000449	0.849389±0.002713
PAES	0.085469±0.006664	1.153052±0.003916
PDEA	N/A	0.473074±0.021721
MODE	0.026230±0.000861	N/A
DEMO/parent	0.000629±0.000044	0.442308±0.039255
DEMO/closest/dec	0.000630±0.000021	0.461174±0.035289
DEMO/closest/obj	0.000642±0.000029	0.458641±0.031362

[1] for the results and parameter settings of both versions of NSGA-II, SPEA and PAES, [8] for PDEA, and [9] for MODE[3]).

Results for PDEA in [8] were evaluated with generational distance instead of the convergence metric. Because PDEA is the approach that is the most similar to DEMO, we present the additional comparison of their results in Tables 5 and 6. Once more, we present the mean and variance of the values of generational distance, averaged over 10 runs.

Table 5. Generational distance achieved by PDEA and DEMO on the problems ZDT1, ZDT2 and ZDT3

Algorithm	Generational distance		
	ZDT1	ZDT2	ZDT3
PDEA	0.000615±0.000000	0.000652±0.000000	0.000563±0.000000
DEMO/parent	0.000230±0.000048	0.000091±0.000004	0.000156±0.000007
DEMO/closest/dec	0.000242±0.000028	0.000097±0.000004	0.000162±0.000013
DEMO/closest/obj	0.000243±0.000050	0.000092±0.000004	0.000169±0.000017

---

[3] The results for MODE are the average of 30 instead of 10 runs. In [9] no diversity metric was calculated.
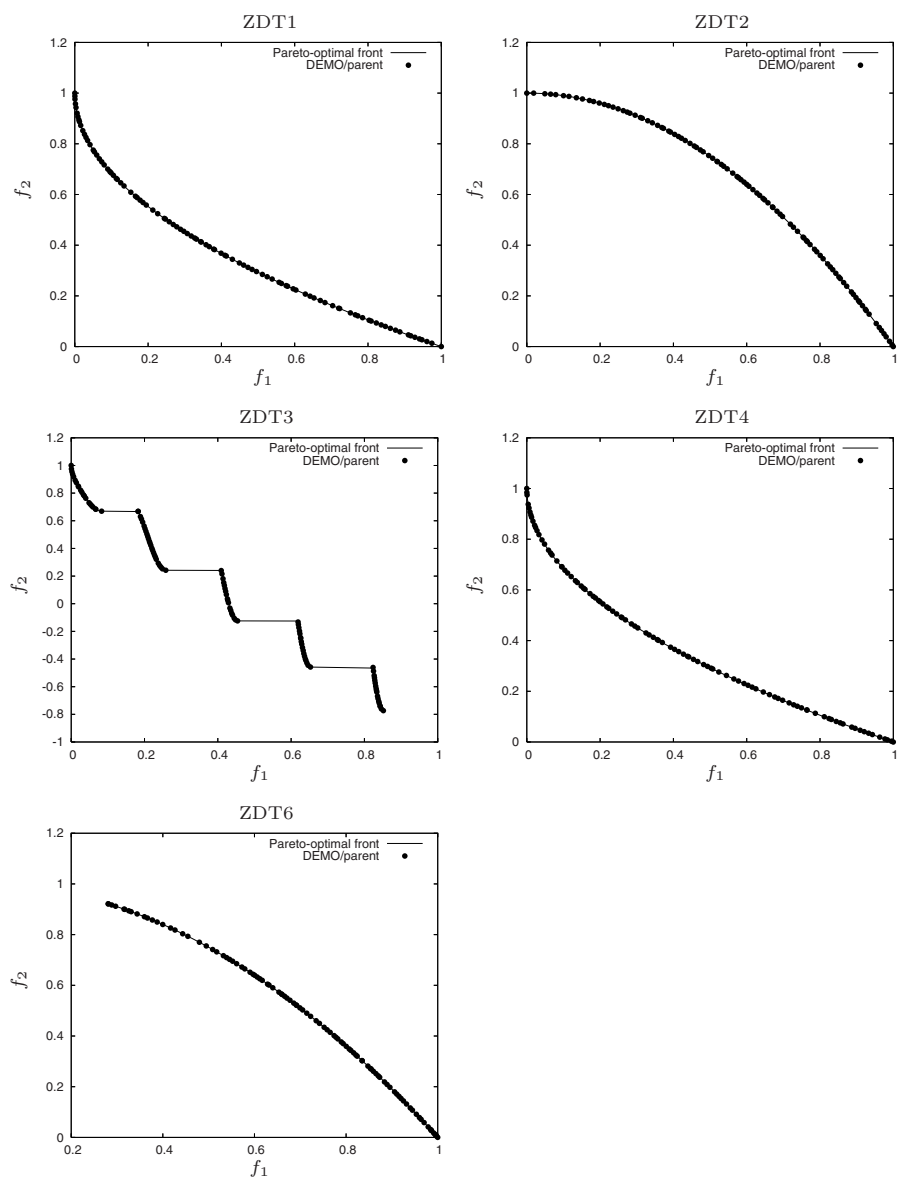
Fig. 4. Nondominated solutions of the final population obtained by DEMO on five ZDT test problems (see Table 7 for more details on these fronts). The presented fronts are the outcome of a single run of DEMO/parent

In Tables 3, 4, 5 and 6, only the results of DEMO with crossover probability 30% are shown. The results obtained with crossover probabilities set to 60% and

Table 6. Generational distance achieved by PDEA and DEMO on the problems ZDT4 and ZDT6

Algorithm	Generational distance	
	ZDT4	ZDT6
PDEA	0.618258±0.826881	0.023886±0.003294
DEMO/parent	0.000202±0.000053	0.000074±0.000004
DEMO/closest/dec	0.000179±0.000048	0.000075±0.000002
DEMO/closest/obj	0.004262±0.006545	0.000076±0.000003

Table 7. Metric values for the nondominated fronts shown in Fig. 4

Problem	Convergence metric	Diversity metric
ZDT1	0.001220	0.297066
ZDT2	0.000772	0.281994
ZDT3	0.001220	0.274098
ZDT4	0.001294	0.318805
ZDT6	0.000648	0.385088

90% for all three variants were always worse than the ones given in the tables and are not presented for the sake of clarity[4].

Figure 4 shows the nondominated fronts obtained by a single run of DEMO/parent with crossover probability 30%. Table 7 summarizes the values of the convergence and diversity metrics for the nondominated fronts from Fig. 4.

5   Discussion

As mentioned in the previous section, DEMO with crossover probability of 30% achieved the best results. This is in contradiction with the recommendations by the authors of DE [10] and confirms Madavan's findings in [8]. However, we have to take these results with caution. Crossover probability is highly related to the dimensionality of the decision space. In our study, only high-dimensional functions were used. When operating on low-dimensional decision spaces, higher values for crossover probabilities should be used to preserve the diversity in the population.

The challenge for MOEAs in the first three test problems (ZDT1, ZDT2 and ZDT3) lies in the high-dimensionality of these problems. Many MOEAs have achieved very good results on these problems in both goals of multiobjective optimization (convergence to the true Pareto front and uniform spread of solutions along the front). The results for the problems ZDT1 and ZDT2 (Tables 3 and 5) show that DEMO achieves good results, which are comparable to the

---

[4] The interested reader may find all nondominated fronts obtained by the three versions of DEMO with the three different crossover probabilities on the internet site http://dis.ijs.si/tea/demo.htm.

results of the algorithms NSGA-II and PDEA. On ZDT3, the three DEMO variants outperform all other algorithms used in comparison. On the first three test problems we cannot see a meaningful difference in performance of the three DEMO variants.

ZDT4 is a hard optimization problem with many (21^9) local Pareto fronts that tend to mislead the optimization algorithm. In Tables 4 and 6 we can see that all algorithms (with the exception of DEMO) have difficulties in converging to the true Pareto front. Here, we can see for the first time that there is a notable difference between the DEMO variants. The third variant, DEMO/closest/obj, performs poorly compared to the first two variants, although still better than other algorithms. While the first two variants of DEMO converge to the true Pareto optimal front in all of the 10 runs, the third variant remains blocked in a local Pareto front 3 times out of 10. This might be caused by DEMO/closest/obj putting too much effort in finding well spaced solutions and thus falling behind in the goal of convergence to the true Pareto optimal front. In this problem, there is also a big difference in results produced by the DEMO variants with different crossover probabilities. When using crossover probability 60% or 90%, no variant of DEMO ever converged to the true Pareto optimal front.

With the test problem ZDT6, there are two major difficulties. The first one is thin density of solutions towards the Pareto front and the second one non-uniform spread of solutions along the front. On this problem, all three DEMO variants outperform all other algorithms (see Tables 4 and 6). The results of all DEMO variants are also very similar and almost no difference is noted when using different crossover probabilities. Here, we note that the diversity metric value is worse than on all other problems. This is because of the non-uniform spread of solutions that causes difficulties although the convergence is good.

6   Conclusion

DEMO is a new DE implementation dealing with multiple objectives. The biggest difference between DEMO and other MOEAs that also use DE for reproduction is that in DEMO the newly created good candidates immediately take part in the creation of the subsequent candidates. This enables fast convergence to the true Pareto front, while the use of nondominated sorting and crowding distance metric in truncation of the extended population promotes the uniform spread of solutions.

In this paper, three variants of DEMO were introduced. The detailed analysis of the results brings us to the following conclusions. The three DEMO variants are as effective as the algorithms NSGA-II and PDEA on the problems ZDT1 and ZDT2. On the problems ZDT3, ZDT4 and ZDT6, DEMO achieves better results than any other algorithm used for comparison. As for the DEMO variants, we have not found any variant to be significantly better than another. Crowding DE thus showed not to bring the expected advantage over standard DE. Because DEMO/closest/dec and DEMO/closest/obj are computationally more expensive

than DEMO/parent and do not bring any important advantage, we recommend the variant DEMO/parent be used in future experimentation.

In this study, we also investigated the influence of three different settings of the crossover probability. We found that DEMO in all three variants worked best when the crossover probability was low (30%). These findings, of course, cannot be generalized because our test problems were high-dimensional (10- or 30-dimensional). In low-dimensional problems, higher values of crossover probability should be used to preserve the diversity in the population. Seeing how crossover probability affects DEMO's performance, we are now interested if the other parameter used in candidate creation (the scaling factor F) also influences DEMO's performance. This investigation is left for further work. In the near future, we also plan to evaluate DEMO on additional test problems.

Acknowledgment

References

1. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA–II. 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 (TIK), Swiss Federal Institute of Technology (ETH) Zurich, Gloriastrasse 35, CH-8092 Zurich, Switzerland (2001)
3. Price, K.V., Storn, R.: Differential evolution – a simple evolution strategy for fast optimization. Dr. Dobb's Journal **22** (1997) 18–24
4. Lampinen, J.: (A bibliography of differential evolution algorithm) http://www2.lut.fi/~jlampine/debiblio.htm
5. 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 (CEC'2001). Volume 2, Piscataway, New Jersey, IEEE Service Center (2001) 971–978
6. Abbass, H.A.: The self-adaptive pareto differential evolution algorithm. In: Congress on Evolutionary Computation (CEC'2002). Volume 1, Piscataway, New Jersey, IEEE Service Center (2002) 831–836
7. Zitzler, E., Thiele, L.: Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach. IEEE Transactions on Evolutionary Computation **3** (1999) 257–271
8. Madavan, N.K.: Multiobjective optimization using a pareto differential evolution approach. In: Congress on Evolutionary Computation (CEC'2002). Volume 2, Piscataway, New Jersey, IEEE Service Center (2002) 1145–1150

9. Xue, F., Sanderson, A.C., Graves, R.J.: Pareto-based multi-objective differential evolution. In: Proceedings of the 2003 Congress on Evolutionary Computation (CEC'2003). Volume 2, Canberra, Australia, IEEE Press (2003) 862–869
10. Storn, R.: (Differential evolution homepage) http://www.icsi.berkeley.edu/~storn /code.html
11. Thomsen, R.: Multimodal optimization using crowding-based differential evolution. In: 2004 Congress on Evolutionary Computation (CEC'2004). Volume 1, Portland, Oregon, USA, IEEE Service Center (2004) 1382–1389
12. Zitzler, E., Deb, K., Thiele, L.: Comparison of multiobjective evolutionary algorithms: Empirical results. Evolutionary Computation **8** (2000) 173–195