Solving Rotated Multi-objective Optimization Problems Using Differential Evolution

Antony W. Iorio and Xiaodong Li

School of Computer Science and Information Technology,
Royal Melbourne Institute of Technology University,
Melbourne, Vic. 3001, Australia
{iantony, xiaodong}@cs.rmit.edu.au
http://goanna.cs.rmit.edu.au/~xiaodong/ecml/

Abstract. This paper demonstrates that the self-adaptive technique of Differential Evolution (DE) [1] can be simply used for solving epistatic multi-objective optimization problems. The real-coded crossover and mutation rates within the NSGA-II [2] have been replaced with a simple Differential Evolution scheme and results are reported on a rotated problem which has presented difficulties using existing Multi-objective Evolutionary Algorithms. The Differential Evolution variant of NSGA-II has demonstrated rotational invariance and superior performance over the NSGA-II on this problem.

1 Introduction

Traditional genetic algorithms that employ low mutation rates and fixed step sizes have significant trouble with problems exhibiting *epistatic* behaviour, where parameters are interdependent. Although the NSGA-II is a very robust multi-objective optimization algorithm it suffers from the same limitations as traditional Genetic Algorithms on these problems.

Previous work has reported on the poor performance of a number of MOEAs, including NSGA-II, on a rotated problem which exhibits *epistatic* behaviour [2]. The primary reason for this poor performance is the crossover and mutation operators employed within the algorithm. Epistatic problems of this type require correlated self-adapting mutation step sizes in order to make timely progress in optimization. Traditional GAs are unsuited to this type of problem [3]. In contrast, Differential Evolution has previously demonstrated rotationally invariant behaviour in the single objective domain [4]. This provides motivation to further demonstrate its worth as a technique for addressing epistatic multi-objective optimization problems. After surveying existing works in the literature, it is apparent that no work has explicitly demonstrated rotationally invariant behaviour in multi-objective problems. Therefore, a simple alteration to the NSGA-II has been proposed, which replaces the mutation and crossover operators with a Differential Evolution algorithm for generating candidate solutions. Differential Evolution has all the desired properties necessary to handle complex *epistatic* problems without the implementation complexity and computation cost of some self-adaptive Evolutionary Strategies [4]. A number of experiments have been conducted on a uni-modal rotated problem from the literature [2], and found that integrating Differential Evolution within the NSGA-II achieves rotational invariance on this problem.

In the following section a brief introduction to the important concepts of Multi-objective Optimization, Differential Evolution, and Rotated Problems is provided. Section 3 will discuss the proposed model which integrates Differential Evolution with the NSGA-II. Section 4 outlines the performance metrics employed in this study. Section 5 describes the experiments that were conducted, followed by the parameter settings and discussion of results in Section 6 and 7. The outcomes of this work and some possible future directions are outlined in Section 8.

2 Background

2.1 Multi-objective Optimization

Multi-objective optimization deals with the optimization of problems which are formulated with some or possibly all of the objectives in conflict with each other. Such problems can be described as a vector of objectives $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), ..., f_n(\mathbf{x}))$ subject to a vector of parameters $\mathbf{x} = (x_1, x_2, ..., x_m) \in \mathbf{X}$, where $\mathbf{x}$ is an input parameter vector from the parameter vector space $\mathbf{X}$, n is the number of objectives, and m is the number of parameters. A solution $\mathbf{x} = (x_1, x_2, ..., x_n)$ dominates a solution $\mathbf{y} = (y_1, y_2, ..., y_n)$ if objective function $f_i(\mathbf{x})$ is no worse than objective function $f_i(\mathbf{y})$ for all n objectives and there exists some objective j where $f_j(\mathbf{x})$ is better than $f_j(\mathbf{y})$. The non-dominated solutions in a population are those solutions which are not dominated by any other individual in the population. Multiobjective evolutionary optimization is typically concerned with finding a diverse range of solutions close to the Pareto-optimal front, which is the globally non-dominated region of the objective space.

A number of evolutionary multiobjective algorithms have been developed since the late 80s, and NSGA-II [2], amongst others, is typically regarded as the current state of the art.

2.2 Differential Evolution

Differential Evolution is a population-based direct-search algorithm for global optimization. It has demonstrated its robustness and power from a variety of applications such as neural network learning [5], IIR-filter design [6], and the optimization of aerodynamic shapes [7]. It has a number of important characteristics which make it attractive as a global optimization technique, and the reader is referred to [4] for an excellent introduction to Differential Evolution which covers this in more detail. The primary property of Differential Evolution that will be the topic of study in this paper is rotational invariance.

Differential Evolution differs from other EAs in the mutation and recombination phase. Unlike stochastic techniques such as Genetic Algorithms and

Evolutionary Strategies, where perturbation occurs in accordance with a random quantity, Differential Evolution uses weighted differences between solution vectors to perturb the population. In single objective optimization, the new individual is compared with a current population member and if it evaluates better it replaces the member.

$$\begin{aligned} &\textit{randomly select } \ r_1, r_2, r_3 \in \{1, 2, ..., n\};\ r_1 \neq r_2 \neq r_3 \neq i \\ &\mathbf{u}_{i,G+1} = \mathbf{x}_{i,G} + K \cdot (\mathbf{x}_{r3,G} - \mathbf{x}_{i,G}) + F \cdot (\mathbf{x}_{r1,G} - \mathbf{x}_{r2,G}) \end{aligned} \tag{1}$$

If the new individual $\mathbf{u}_{i,G+1}$,evaluates better than the currently selected individual $\mathbf{x}_{i,G}$, then the current individual is replaced with the new one. The algorithm iterates from i to n, where n is the size of the population.

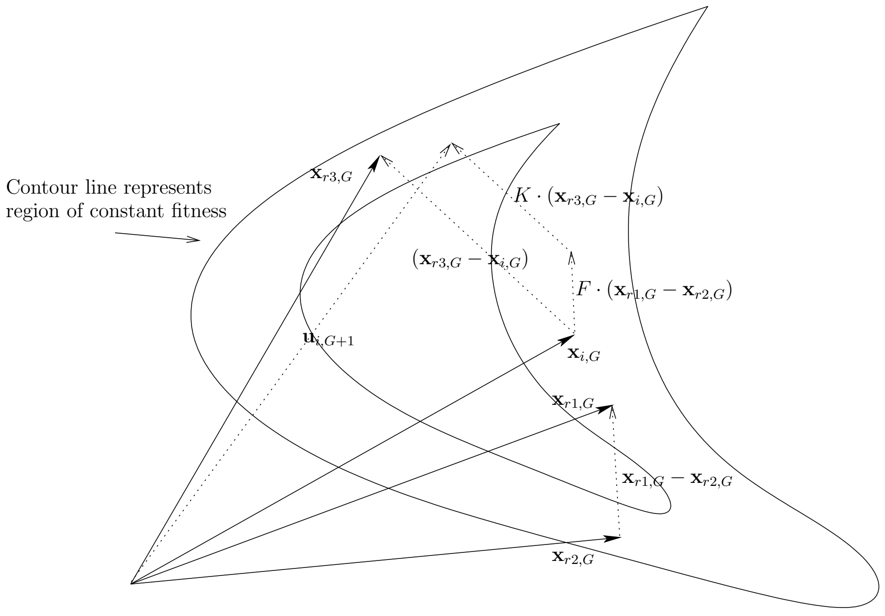
Contour line represents region of constant fitness
$\mathbf{x}_{r3,G}$
$K \cdot (\mathbf{x}_{r3,G} - \mathbf{x}_{i,G})$
$(\mathbf{x}_{r3,G} - \mathbf{x}_{i,G})$
$F \cdot (\mathbf{x}_{r1,G} - \mathbf{x}_{r2,G})$
$\mathbf{u}_{i,G+1}$
$\mathbf{x}_{i,G}$
$\mathbf{x}_{r1,G}$
$\mathbf{x}_{r1,G} - \mathbf{x}_{r2,G}$
$\mathbf{x}_{r2,G}$

Fig. 1. The above figure shows the vector addition and subtraction necessary to generate a new candidate solution in *DE/current-to-rand/1*.

The Differential Evolution variant described here is known as *DE/current-to-rand/1* (Equation 1) and it guarantees rotational invariance. The shorthand description of this model states that *DE/current-to-rand/1* generates 'vectors that are linear combinations of the *current* vector $\mathbf{x}_{i_G}$, and a *rand*omly chosen donor $\mathbf{x}_{r3,G}$' [4]. The crossover constant, *CR*, is not used and implicitly equals *1*. The Population of a Differential EA is typically randomly initialised according to the initial parameter bounds. At each generation G, the population undergoes perturbation. Three individuals, or solution vectors denoted by $\mathbf{x}$, are randomly selected from the population such that $\mathbf{x}_{r1,G} \neq \mathbf{x}_{r2,G} \neq \mathbf{x}_{r3,G}$. The coefficient

K is responsible for the level of combination that occurs between $\mathbf{x}_{r3,G}$ and the current individual $\mathbf{x}_{i,G}$. The coefficient F is responsible for scaling the step size resulting from the vector subtraction $\mathbf{x}_{r1,G} - \mathbf{x}_{r2,G}$. Figure 1 details the relationship between the vectors responsible for the generation of a new candidate solution.

2.3 Rotated Problems

A rotated problem is rotated on one or more planes in the decision space, where the number of planes is determined by the dimensionality of the problem. A problem with D dimensions in the decision space has $D(D-1)/2$ possible planes of rotation. A problem rotated on all possible decision space planes means that every decision space variable has some dependency on every other. The interdependence between parameters is known as *epistasis*.

In order to generate a rotated problem, each solution vector $\mathbf{x}$ is multiplied by the rotation matrix $\mathbf{M}$, and the result is assigned to $\mathbf{y}$ (Equation 2). The new vector is then evaluated on each of the objective functions.

Figure 2 demonstrates the effect of rotation on the 2-dimensional version of the multi-objective problem outlined below. The shapes of the functions stay the same, but their orientations change.

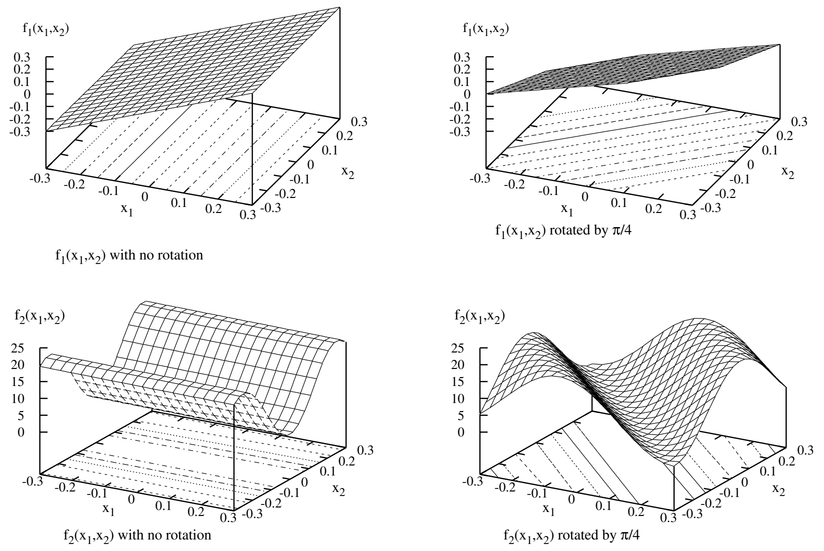

Fig. 2. The above figure shows the effect of a 45-degree rotation on the x_1x_2 plane on function f_1 and f_2. Before rotation, the functions are aligned with the coordinate system, and after rotation they are not.

$$\begin{aligned} minimize \quad & f_1(\mathbf{y}) = y_1 \text{ } and \text{ } f_2(\mathbf{y}) = g(\mathbf{y})exp(-y_1/g(\mathbf{y})) \\ where \quad & g(\mathbf{y}) = 1 + 10(D-1) + \sum_{i=2}^{D} \left[y_i^2 - 10\cos(4\pi y_i)\right] \\ and \quad & \mathbf{y} = \mathbf{M}\mathbf{x},\ -0.3 \le x_i \le 0.3,\ for\ i = 1, 2, ..., D. \end{aligned} \tag{2}$$

It is apparent from the contour plots in figure 2 that before rotation the functions are aligned with the coordinate system. When the functions are aligned with the coordinate system, it is possible to make progress in the search by perturbing the parameters x_1 and x_2 independently. Unfortunately many interesting problems are not of this variety. With rotated problems, significant progress in the search can only proceed by making simultaneous progress across all parameters within a solution vector. On these types of problems, the small mutation rates frequently used in Genetic Algorithms are known to be even less efficient than a random search [3]. Self-adaptation has been relatively successful at solving this sort of problem using Evolutionary Strategies, but it requires the learning of appropriate correlated mutation step sizes and it can be rather computationally expensive when D becomes large [4]. Differential Evolution is an attractive solution to this problem because of its ability to self-adapt to the fitness landscape through the self-correlation of mutation step sizes by adding the difference between randomly selected solution vectors.

3 NSDE: A simple modification to the NSGA-II

The NSGA-II algorithm uses elitism and a diversity preserving mechanism. N offspring are created from a parent population of size N. The combined population of size *2N* is sorted into separate non-domination levels. Individuals are selected from this combined population to be inserted into the new population, based on their non-domination level. If there are more individuals in the last front than there are slots remaining in the new population of size N, a diversity preserving mechanism is employed. Individuals from this last front are placed in the new population based on their contribution to diversity in the population. The algorithm then iterates until some termination condition is met. The NSGA-II uses a real-coded crossover and mutation operator but in the multi-objective implementation of *DE/current-to-rand/1*, NSDE (Non-dominated Sorting Differential Evolution), these mutation and recombination operators were not used, and were replaced with Differential Evolution. In the single objective implementation of the Differential Evolution, if the new candidate $\mathbf{u}_{i,G+1}$ evaluates better than the current individual $\mathbf{x}_{i,G}$, the current individual is replaced with the new individual. In the multi-objective implementation this is not possible because we don't know which individual is better until all candidates are sorted together and assigned to a non-domination level. Therefore, $\mathbf{u}_{i,G+1}$ is first added to the new candidate offspring population. New candidates are generated using *DE/current-to-rand/1* until the candidate offspring population is filled up to size

N. The new individuals are then evaluated on the objective functions, and then subjected to the combined non-dominated sorting described above. For further details regarding the implementation of the NSGA-II, the reader is referred to the following paper for a detailed description of the diversity preserving mechanisms and non-dominated sorting [2].

4 Performance Metrics

We use the following performance metrics introduced by Zitzler et al. [**?**]:

$$\mathcal{M}_1^*(Y') := \frac{1}{|Y'|} \sum_{p' \in Y'} \mathbf{min}\{||\mathbf{p}' - \bar{\mathbf{p}}||^*; \bar{\mathbf{p}} \in \bar{\mathbf{Y}}\} \tag{3}$$

$$\mathcal{M}_2^*(Y') := \frac{1}{|Y' - 1|} \sum_{p' \in Y'} |\{q' \in Y'; ||p' - q'||^* > \sigma^*\}| \tag{4}$$

$$\mathcal{M}_3^*(Y') := \sqrt{\sum_{i=1}^{n} \mathbf{max}\{||\mathbf{p}_\mathbf{i}' - \mathbf{q}_\mathbf{i}'||^*; \mathbf{p}', \mathbf{q}' \in \mathbf{Y}'\}} \tag{5}$$

where Y' is the set of objective vectors corresponding to the non-dominated solutions found, and $\bar{Y}$ is a set of uniform Pareto-optimal objective vectors. A niche neighbourhood size, $\sigma^* > 0$, is used in equation (4) to calculate the distribution of the non-dominated solutions. $\mathcal{M}_1^*(Y')$ gives the average distance from Y' to $\bar{Y}$. $\mathcal{M}_2^*(Y')$ describes how well the solutions in Y' are distributed. $\mathcal{M}_2^*(Y')$ should produce a value between [0, $|Y'|$] as it estimates the number of niches in Y' based on σ^*. The higher the value, the better the distribution is according to σ^*. $\mathcal{M}_3^*(Y')$ measures the spread of Y'.

5 Experiments

Experiments were conducted on the rotated problem in section 2.3. The dimensionality of the decision space was 10, resulting in 45 possible planes of rotation. Rotations were performed on each plane, introducing epistasis between all parameters. In order to demonstrate the rotational invariance of the NSDE on the problem, we performed experiments with 0 degrees of rotation (no parameter interactions) up to 45 degrees of rotation, at 5 degree intervals. Each experiment was run 30 times, for a total of 800 generations (80,000 evaluations) for each run. For comparative purposes the same experiments were performed with the NSGA-II as well. Results are presented in Figure 3, Figure 4, and Table 1.

6 Parameter Settings

A population size of 100 was used for both the NSDE and NSGA-II. A crossover rate of 0.9 and mutation rate of 0.1 were used with the NSGA-II. η_c and η_m

control the distribution of the crossover and mutation probabilities respectively and were assigned values of 10 and 50 respectively. The choice of the NSGA-II parameters is the same as the parameter values previously used on this rotated problem in other work. For the NSDE, F was set to 0.8 and K was set to 0.4. Suggestions from the literature helped guide our choice of parameter values for the NSDE [4]. The niche neighbourhood size, σ^* was set to 0.

7 Results and Discussion

From Table 1 it is apparent that the NSDE maintains a significantly better coverage (M_2^*) and spread (M_3^*) than the NSGA-II, independent of the degree of rotation on each plane. Although M_1^* suggests the NSGA-II apparently has reasonable convergence towards the pareto-optimal front, if one takes into account the other criteria such as coverage (M_2^*) and spread (M_3^*) of solutions, it is apparently quite poor. Figure 3 and 4 contains plots of 30 runs of the final non-dominated set after 80,000 evaluations. These figures further demonstrate that the NSDE was able to converge closely to the Pareto-optimal front rather consistently, and independent of the degree of rotation.

The only difference between the NSDE and the NSGA-II is in the method of generating new individuals. NSDE uses the self-correlating step sizes of Differential Evolution, and the NSGA-II uses real-coded crossover and mutation operators. It is obvious that the cause of the poor performance by the NSGA-II on the rotated problem is the mutation and crossover operators, which are not suited to this type of problem. We have demonstrated that Differential Evolution can provide rotationally invariant behaviour on multi-objective optimization problems.

8 Conclusion

Outside of Evolutionary Strategies, Differential Evolution is currently one of the few viable technique for optimizing multi-objective optimization problems with epistatic interactions between parameters. We can also conclude that traditional GAs are unsuitable for optimizing difficult problems of this kind. Future work in this area should consider even harder rotated problems with the NSDE, possibly incorporating some of the features of existing test functions, such as multi-modality, non-uniformity, and discontinuities.

References

1. Price, K. V.: 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

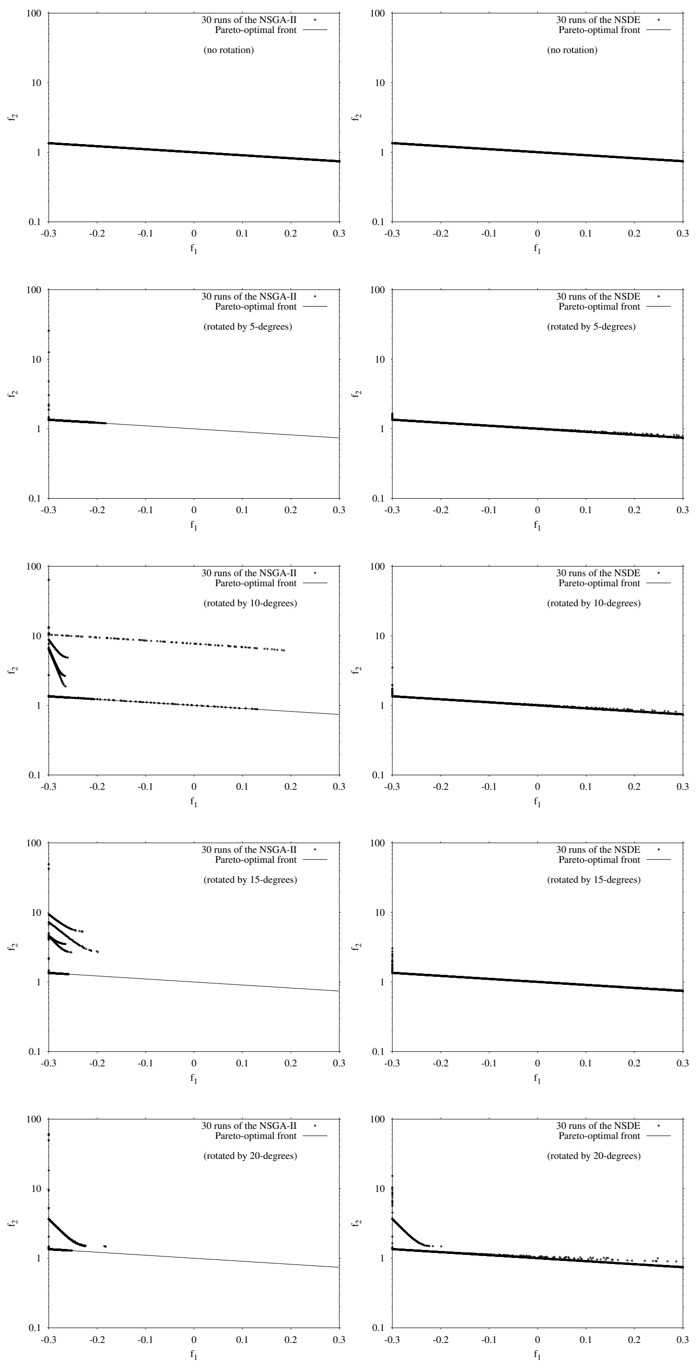

Fig. 3. The left and right plots respectively show 30 runs of the NSGA-II and the NSDE algorithm on the rotated problems.

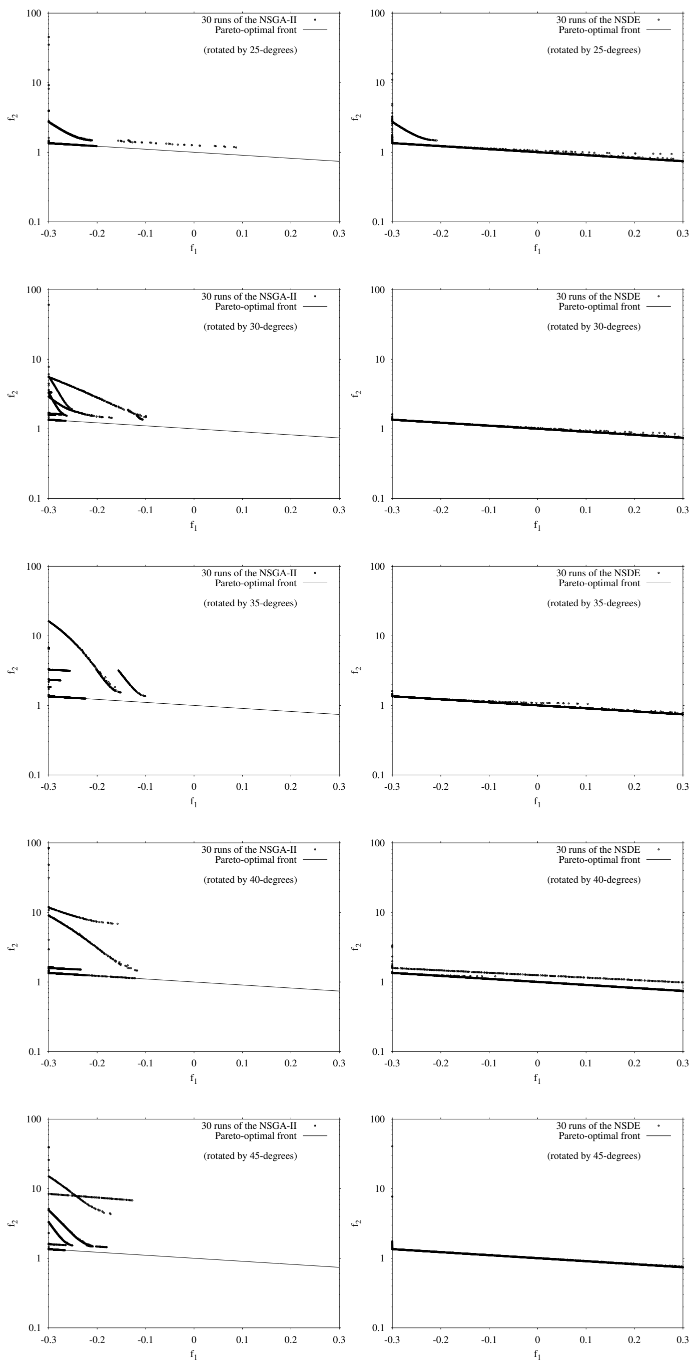

Fig. 4. The left and right plots respectively show 30 runs of the NSGA-II and the NSDE algorithm on the rotated problems.

Table 1. $\mathcal{M}_1^*$, $\mathcal{M}_2^*$, $\mathcal{M}_3^*$, and the number of evaluations (averaged over 30 runs). R_d represents the rotated problem where d is the degree of rotation on each plane.

Metric	Algorithm	R_0	R_5	R_{10}	R_{15}	R_{20}	R_{25}	R_{30}	R_{35}	R_{40}	R_{45}
$\mathcal{M}_1^*$	NSGA-II	6.26E-04	2.42E-02	1.49E+00	1.14E+00	3.49E-01	3.71E-01	5.18E-01	8.97E-01	8.17E-01	1.01E+00
		±7.55E-05	±9.31E-02	±3.07E+00	±1.62E+00	±6.85E-01	±5.47E-01	±7.91E-01	±2.08E+00	±1.79E+00	±1.87E+00
	NSDE	2.22E-03	3.76E-03	5.60E-03	2.95E-01	2.18E+00	2.36E-03	1.38E+00	3.86E-03	2.29E-02	5.82E-01
		±1.97E-04	±5.22E-03	±1.18E-02	±7.51E-01	±9.03E-16	±4.41E-19	±1.07E+00	±5.06E-03	±6.26E-02	±9.78E-01
$\mathcal{M}_2^*$	NSGA-II	9.86E+01	8.56E+01	6.87E+01	5.35E+01	6.41E+01	6.94E+01	5.22E+01	5.24E+01	4.60E+01	5.11E+01
		±7.00E-02	±2.46E+00	±2.42E+01	±2.35E+01	±1.98E+01	±2.66E+01	±3.61E+01	±3.42E+01	±3.39E+01	±3.75E+01
	NSDE	9.85E+01	9.85E+01	9.85E+01	9.86E+01	9.85E+01	9.86E+01	9.85E+01	9.85E+01	9.83E+01	9.86E+01
		±5.49E-02	±1.28E-01	±2.14E-01	±2.22E-01	±4.68E-01	±5.22E-01	±1.19E-01	±3.13E-01	±1.36E+00	±2.00E-01
$\mathcal{M}_3^*$	NSGA-II	1.10E+00	5.44E-01	6.47E-01	5.37E-01	8.66E-01	9.34E-01	8.39E-01	5.61E-01	7.95E-01	1.43E+00
		±5.48E-06	±3.33E-01	±7.94E-01	±6.27E-01	±1.52E+00	±1.22E+00	±1.47E+00	±9.51E-01	±1.75E+00	±2.32E+00
	NSDE	1.10E+00	1.10E+00	1.11E+00	1.12E+00	1.17E+00	1.32E+00	1.10E+00	1.09E+00	1.11E+00	1.28E+00
		±7.48E-04	±1.79E-02	±3.68E-02	±8.01E-02	±3.56E-01	±5.81E-01	±3.14E-02	±5.03E-02	±1.50E-01	±9.62E-01

2. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II, In: IEEE Trans. Evol. Comput., Vol. 6, No. 2. (2002) 182–197
3. Salomon, R.: Re-evaluating Genetic Algorithm Performance Under Coordinate Rotation of Benchmark Functions: A Survey of Some Theoretical and Practical Aspects of Genetic Algorithms. In: Bio Systems, Vol. 39, No. 3. (1996) 263–278
4. Price, K V.: An Introduction to Differential Evolution. In: Corne, D., Dorigo, M., and Glover, F. (eds.): New Ideas in Optimization. McGraw-Hill, London (UK) (1999) 79–108
5. Ilonen, J., Kamarainen, J.-K., Lampinen, J.: Differential Evolution Training Algorithm for Feed-Forward Neural Networks. In: Neural Processing Letters Vol. 7, No. 1 (2003) 93-105
6. Storn, R.: Differential evolution design of an IIR-filter. In: Proceedings of IEEE International Conference on Evolutionary Computation ICEC'96. IEEE Press, New York (1996) 268-273
7. Rogalsky, T., Derksen, R.W. and Kocabiyik, S.: Differential Evolution in Aerodynamic Optimization. In: Proceedings of the 46th Annual Conference of the Canadian Aeronautics and Space Institute. (1999) 29–36