An Analysis of Differential Evolution Parameters on Rotated Bi-objective Optimization Functions

Martin Drozdik[1], Kiyoshi Tanaka[1], Hernan Aguirre[1], Sebastien Verel[2], Arnaud Liefooghe[3], and Bilel Derbel[3]

[1] Interdisciplinary Graduate School of Science and Technology, Shinshu University, Nagano, Japan (e-mail: martin@iplab.shinshu-u.ac.jp)
[2] LISIC, Université du Littoral Côte d'Opale, France
[3] Inria Lille-Nord Europe, Villeneuve d'Ascq, France

Abstract. Differential evolution (DE) is a very powerful and simple algorithm for single- and multi-objective continuous optimization problems. However, its success is highly affected by the right choice of parameters. Authors of successful multi-objective DE algorithms usually use parameters which *do not* render the algorithm *invariant* with respect to *rotation* of the coordinate axes in the decision space. In this work we try to see if such a choice can bring consistently good performance under various *rotations* of the problem. We do this by testing a DE algorithm with many combinations of parameters on a testbed of bi-objective problems with different modality and separability characteristics. Then, we explore how the performance changes when we rotate the axes in a controlled manner. We find out that our results are consistent with the single-objective theory but *only for unimodal problems*. On multi-modal problems, unexpectedly, parameter settings which do *not* render the algorithm rotationally invariant have a consistently good performance for all studied rotations.

Keywords: differential evolution, rotational invariance, multi-objective optimization, parameter analysis

1 Introduction

Differential evolution [6] started as a simple single-objective continuous optimization heuristic. The need for a versatile multi-objective optimizer has motivated researchers to generalize the basic algorithm for multi-objective problems. Now we have a great number of multi-objective DE variants. Many of them use the same mechanism to generate new individuals. In a problem with n variables a new individual is created using a crossover variation operator which randomly selects $k; k \leq n$ variables which are perturbed. The magnitude of the mutation is generated by scaling a *difference* of randomly chosen individuals.

Many research papers on DE such as [1] or [7] provide little insight into how the authors chose the parameters for their benchmarking. We find this striking since many authors choose their parameters such that the crossover operator perturbs only a small number of variables in an existing individual. In other words, the search for the Pareto optimal set proceeds *along the coordinate axes*. Since these algorithms perform very well [1][7], we have a suspicion that this may be due to some characteristic of the problem, such as separability, that makes it easy to optimize along the axes. This would mean that if the axes are transformed, the algorithm should lose some performance.

Very strict warning against the practice of perturbing a small number of variables at a time has been raised as soon as 1996 by Salomon [9]. Salomon empirically demonstrated that the stellar performance of many popular single-objective genetic algorithms owes to the fact that most of the benchmark functions were *separable* and that the low mutation rate caused them to be optimized *one component at a time*. Once Salomon stripped the separability by *rotating* the principal axes of the benchmark functions, many algorithms were significantly slowed down, while some failed to converge completely. Salomon's theoretical results state that, in some cases, the probability of finding the global optimum can drop below that of random search. We are concerned that the same is true for the multi-objective realm since many authors perform their experiments with separable test functions.

In DE the number of variables that are perturbed is controled by a parameter. If all variables are perturbed, the algorithm has the same performance *regardless of rotation*. Let us have a parameter setting, that perturbs only a *small proportion* of the variables, which outperforms a setting that perturbs *all* variables. In this work we attempt to answer this question: Is this *exceptionally good* performance on a problem with a particular alignment of the coordinate axes balanced by *exceptionally bad* performance on a different alignment?

We do this *empirically* by observing the performance of a simple multi-objective algorithm DEMO (Differential evolution for multiobjective optimization) [7] on a bi-objective subset of the WFG (Walking Fish Group) test suite [3]. We run all our experiments with a fixed population size and a fixed number of variables, while varying the *parameters*. Then we gradually *rotate* the problems in a controlled manner and observe the new behavior.

The answer to our question is, unexpectedly, *negative*. We find a statistically significant difference between the performance on the rotated problems and the original ones. Closer inspection reveals that a systematic performance *loss* happens when we rotate the *separable* problems, but the performance is still *significantly better* than for a rotationally invariant algorithm. We find that this happens for *multi-modal* problems, while *single-modal* problems exhibit the behavior we would expect from the work of Salomon.

In the following section we provide background information on DE and on the previous related work on DE parameters. In Section 3 we introduce the experimental design, where we explain which problems are used and why were they chosen. In addition we introduce a new performance metric called the *relative hypervolume*, and explain the controlled manner in which the rotations are generated. In Section 4 we present our data along with a discussion. Finally, in Section 5, we present the conclusion.

Algorithm 1: Modified DEMO [7] algorithm

```
1  initialize $P = \{X_1, ..., X_N\}$ uniformly randomly in the decision space
2  for generation := 1 to $G_{max}$ do  Evolutionary loop
3      for target := 1 to $N$ do  Generational loop
4          randomly generate mutually distinct $r_1, r_2, r_3 \neq$ target
5          $X_{\text{mutant}} := X_{r_1} + \text{F}(X_{r_2} - X_{r_3})$
6          randomly generate inv $\in \{1, \ldots, n\}$
7          for $i := 1$ to $n$ do
8              if rand(0.0; 1.0) < Cr or $i =$ inv then
9                  $X_{\text{trial},i} := X_{\text{mutant},i}$
10             else
11                 $X_{\text{trial},i} := X_{\text{target},i}$
12             end
13         end
14         project $X_{\text{trial}}$ to decision space
15         if $X_{\text{target}}$ dominates $X_{\text{trial}}$ then
16             discard $X_{\text{trial}}$
17         else if $X_{\text{trial}}$ dominates $X_{\text{target}}$ then
18             replace $X_{\text{target}}$ with $X_{\text{trial}}$
19         else if $X_{\text{target}}$ and $X_{\text{trial}}$ are mutually non-dominated then
20             add $X_{\text{trial}}$ to the end of the population
21         end
22     end
23     Trim the $P$ to size $N$ using non-dominated sorting[7] and MNN diversity[4]
24 end
```

2 Background

2.1 Differential Evolution

In this section, we describe the variant of DE which we use in this work. It is a slightly modified version of the DEMO algorithm [7] described in Algorithm 1. The modified parts are highlighted with yellow color in lines 14 and 23. Let us look at the algorithm in detail. First, the population of the algorithm is randomly initialized (line 1). Then, the algorithm runs for a fixed number of generations (evolutionary loop). In each generation, DE iterates through the entire population generating a *trial* individual, which is compared to an existing *target* individual. The trial is generated utilizing the traditional method in lines 4 to 13. Here we introduce the parameters of DE.

To generate a new individual, three distinct individuals are selected from the population. By forming a *difference* between two of them, scaling it by a fixed parameter F, and adding to a third individual we obtain a so called mutant individual (line 5). The trial individual is created by crossover between the mutant and the target. First, a randomly chosen variable from the mutant is inherited (line 6). Next, each other variable is inherited from the mutant with a fixed crossover probability of Cr. Otherwise it comes from the target.

After the individual is generated we project it to the decision space. This is a modification of the original algorithm which did not explicitly deal with domain issues. The purpose is to keep the algorithm as simple as possible while being able to optimize problems with simple constraints. Next the trial is compared to the target. If one of them is dominated by the other, we discard the dominated one. If they are *mutually non-dominated*, we keep them both. At the end of the generation loop, the population is trimmed to size N using non-dominated sorting and the M nearest neighbor diversity estimation procedure [4]. We chose this procedure because it achieves a better distribution along the Pareto front than the original crowding distance computation.

Note that $Cr = 1$ is the *only* value of Cr for which the DE algorithm is rotationally invariant with probability 1. Rotational invariance does not by itself imply good performance. Its merit is that it allows us to *generalize* a single observation to an entire invariance class [2].

2.2 Crossover Probability and Separability

In this section we summarize what we know about the relationship between the separability of the test functions and the good choice of Cr parameter.

There are many different types of separability. One of the simplest is *additive* separability. A function $f : D \subseteq \mathbb{R}^n \to \mathbb{R}$ is called *additively* separable if:

$$\exists f_1, \dots, f_n \text{ such that } f(x_1, \dots, x_n) = \sum_{i=1}^{n} f_i(x_i)$$

The most important consequence of additive separability is that the n-dimensional problem can be optimized sequentially *one variable at a time.* Therefore separable problems are not subject to the curse of dimensionality [9].

Salomon [9] illustrates the problems of algorithms which vary the individuals *one variable at a time* on a quadratic function of two variables in Figure 1. The ellipses in the left part are contours of a *separable* quadratic function. We can see two individuals on one of the contours. The blue individual represents an individual in a randomized algorithm. If we mutate one variable of this individual, the probability to get an improvement in the objective function is relatively high, since the improvement intervals d_1, d_2 are long. If we rotate the coordinate axes, thus rendering the function non-separable, the improvement intervals shrink.

One more illustration of problems which arise is using a sequential deterministic algorithm which finds the optimum with respect to *one variable at a time.* The red individual illustrates the path of one such an algorithm. When the function is aligned with the axes, this algorithm achieves optimum in just two iterations, while in the rotated case the algorithm not only progresses slower, but never actually reaches the optimum.

Huband et al. from the Walking Fish Group (WFG) define separability from the optimizational standpoint[3]. A variable x_i is separable if the set of global optima of a problem:

$$\operatorname*{argmin}_{x_i} f(x_1, \dots, x_n)$$

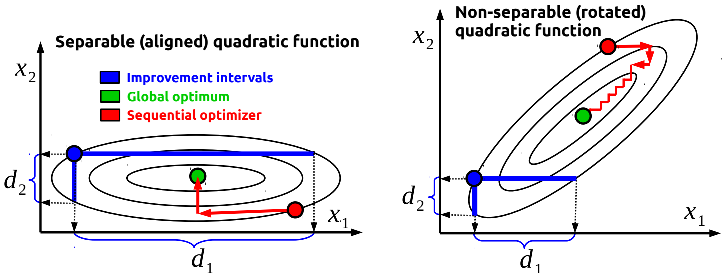


Fig. 1: Illustration of variable-wise optimization on a rotated quadratic function

is the same for any choice of the other variables $x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n$. For example, an additively separable function is WFG-separable, hence WFG-separability is a *generalization* of additive separability. The authors define a separable multi-objective problem as one where *each* objective is separable. The majority of the frequently used DTLZ and ZDT problems are WFG-separable [3], while their objective functions are *not* additively separable.

The multi-objective model is fundamentally different from the single-objective model because all objectives are being optimized *simultaneously*. The global optima of each optimized function constitute only a relatively small subset of the Pareto optimal set. Therefore, it is appropriate to ask if the problems of sequential algorithms which are illustrated in Figure 1 persist in multi-objective optimization. Also, while additively separable *unimodal* functions are inherently similar to the quadratic function in Figure 1, it is not clear if the intuition holds for *multi-modal* functions or for functions which are WFG-separable but not additively separable.

2.3 Variance as a Common Currency

Probably the most significant work on the theoretical properties of DE has been written by Zaharie [10]. Let us collect all the trial vectors that are generated in the course of one generational loop of Algorithm 1 into a set P_{trial}. Then the relationship between the *variance in decision space* of P and P_{trial} is given by the simple equation $\mathbb{E}[Var(P_{\text{trial}})] = c\mathbb{E}[Var(P)]$ where:

$$c = 2\mathrm{F}^2\mathrm{Cr} + \frac{\mathrm{Cr}^2 - 2\mathrm{Cr}}{N} + 1 \tag{1}$$

Zaharie omits the fact that in most DE variants the individuals which generate the trial individual are chosen *distinct* from the target individual (Algorithm 1 line 4). However her results hold unchanged also after adding this assumption.

The work of Zaharie is important since it transforms the two parameters into a single number c (common currency) which has a very intuitive interpretation. If $c < 1$ we see that the algorithm tends to contract the population while if $c > 1$ it expands the population. Based on empirical data Kukkonen concluded in [5]

[8] that a good choice of parameters is one that satisfies $c \in [1.0; 1.5]$ with the upper bound not very strict.

3 Experimental Design

In this section, we describe which test problems we chose and why. We explain what we mean by *rotating the problem* and we propose a new performance metric which we use.

3.1 WFG Problems

In order to explore the relationship between the control parameters of DE and the characteristics of the problem, we chose 4 problems from the WFG test suite[3]. These problems have been chosen since they have the same Pareto front and contain all possible combinations of the *WFG-separability* and *modality* characteristics. They are summarized in Table 1. We chose the number of variables to be 10 of which one is a positional variable.

Table 1: Characteristics of the selected WFG problems

	WFG4	WFG7	WFG6	WFG9
separable	yes	yes	no	no
unimodal	no	yes	yes	no

3.2 Rotations in $\mathbb{R}^n$

As humans we have a very good intuitive understanding of rotation in 2 or 3 dimensional space. However in higher dimensions things are not as intuitive as they might seem. An elementary rotation by the angle ϕ is characterized by the matrix:

$$R^e = \begin{pmatrix} cos(\phi) & -sin(\phi) \\ sin(\phi) & cos(\phi) \end{pmatrix}$$

We can generalize this rotation to n-dimensional space by taking an n-dimensional identity matrix I and replacing $I_{i,i}, I_{i,j}, I_{j,i}, I_{j,j}$ by $R^e_{1,1}, R^e_{1,2}, R^e_{2,1}, R^e_{2,2}$ respectively. We can see that the rotation is not executed *around an axis* as we might intuitively feel, but *around an* $n-2$ *dimensional subspace* which is coincidentally a 1-dimensional axis in the intuitive 3-dimensional case. For our experiments, we generate the rotation matrix R by applying a rotation to *each* $n-2$ *dimensional subspace* in sequence, one rotation after the other.

We rotate the *entire decision space* (DS). This way the entire Pareto optimal set is always attainable since the entire decision space rotates along. In the case of WFG problems this means rotating a *n-dimensional hyper-box.* For example, in order to initialize the population in Algorithm 1 in the rotated DS (line 1), we first initialize the population in the original DS and then multiply by R^{-1}. Similar process is used to project the individual to the rotated DS on line 14. To evaluate the objective value of an individual we first multiply the decision vector by R and evaluate the original objective functions.

3.3 Relative Hypervolume

In our experiments, we use only one performance metric, the hypervolume (HV) [11], since it includes information on both convergence and spread of the individuals. With WFG problems, it is not easy to choose the reference point for the HV. Even if we choose the point as tight as possible, there are some individuals after the initialization of DE which dominate the reference point. Therefore the HV at the start is not zero and it is hard to say if a certain attained HV is good or bad. Moreover, it is hard to make quantitative comparisons based on HV. If some algorithm achieves HV of 100 and another one achieves a HV of 99.98, it may seem that the difference is not very big, but it all depends on the HV *at initialization*. If the algorithms started with HV = 0, the interpretation of the results would be quite different from one where HV = 99.99 at the start.

We attempt to mitigate this problem by subtracting the HV at initialization ($\mathrm{HV}_{\mathrm{init}}$) and normalizing the result using the *maximal attainable hypervolume* ($\mathrm{HV}_{\mathrm{max}}$). We define the relative hypervolume (RHV) in the following equation:

$$\mathrm{RHV} := \frac{\mathrm{HV} - \mathrm{HV}_{\mathrm{init}}}{\mathrm{HV}_{\mathrm{max}} - \mathrm{HV}_{\mathrm{init}}} \tag{2}$$

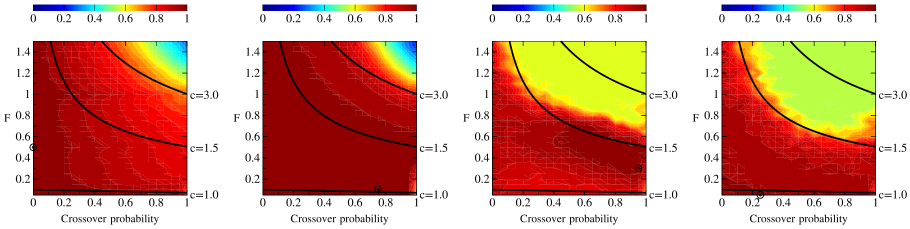
We compute $\mathrm{HV}_{\mathrm{max}}$ deterministically by integrating the space between the true Pareto front (PF) and the reference point. From (2), we have $\mathrm{RHV} \in [1; -\infty)$. $\mathrm{RHV} = 1$ implies convergence, RHV at initialization is 0 and $\mathrm{RHV} < 0$ indicates an algorithm which is receding from the Pareto front.

We use RHV since its normalized nature is more intuitive and it is more robust with respect to the selection of the reference point. It may be more meaningful to compare two algorithm runs in terms of RHV. If we have two algorithm runs starting from the same randomly initialized popluation then the ratio of their RHVs is independent of the choice of the reference point. [4] On the other hand, two independent runs which produce the same final population may yield different relative hypervolume.

4 Results and Discussion

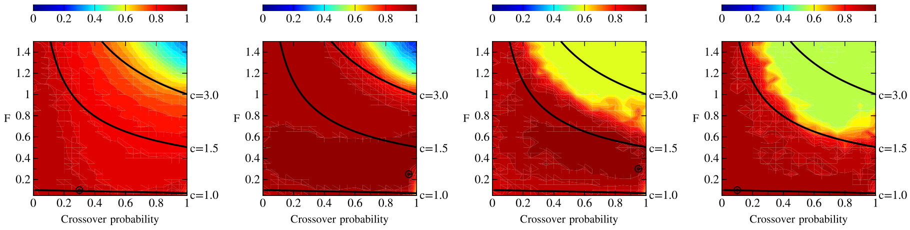
In our experiments we varied the parameters $\mathrm{F} \in [0.05; 1.5]$, $\mathrm{Cr} \in [0; 1]$ equidistantly with a resolution of 0.05. For each combination we performed 10 runs of Algorithm 1. To simplify the setup, the population size was kept constant at 100 individuals and the length of each run was 250 generations. We explored the rotations from 0 to 90 degrees with a resolution of 5 degrees. In the following we discuss our results on a subset of the experimental data. To simplify the analysis, in each section we keep either F, Cr or the rotation angle fixed.

[4] Given that the reference point is dominated by all individuals in the population.



(a) WFG4 (S-MM) (b) WFG7 (S-UM) (c) WFG6 (NS-UM) (d) WFG9 (NS-MM)

Fig. 2: Average RHV without rotation



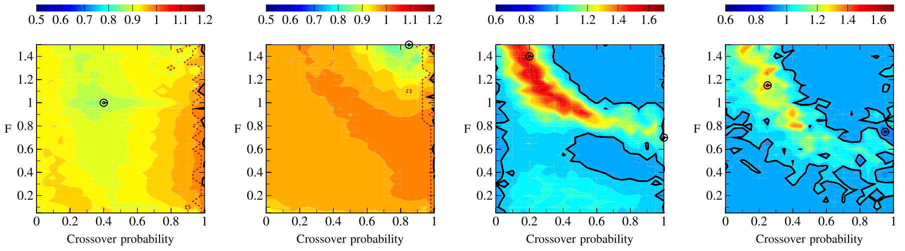
(a) WFG4 (S-MM) (b) WFG7 (S-UM) (c) WFG6 (NS-UM) (d) WFG9 (NS-MM)

Fig. 3: Average RHV with rotation angle of 5 degrees

4.1 Fixed Rotation Angle

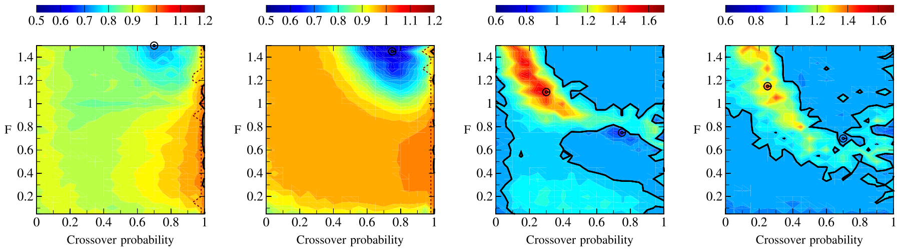
Figure 2 shows the average RHV on *non-rotated* problems. For illustration, we plot the combinations of F and Cr which result in $c = 1.0, 1.5$ and 3.0 according to (1). The circle marks the combination of parameters with the best RHV. The data in all our figures from now on is presented from left to right in the same order as in Table 1. The separable problems (S) are on the left, the non-separable (NS) on the right, while the unimodal (UM) are on the inside and multimodal (MM) ones are near the page margins. For each problem, an L-shaped favorable region containing RHV of 0.8 and higher, roughly corresponds to $c \in [1; 1.5]$. Low value of Cr is more robust, since it allows for a wider interval of F values. Unexpectedly, this holds also for non-separable problems WFG6 and WFG9.

The effect of introducing a rotation by 5 degrees is shown in Figure 3. The two figures seem identical, but the *ratio* of these averages in Figure 4 reveals a difference. A value of less than 1 indicates that the rotation caused the performance to decrease. We highlighted the contour at level 1 and marked the maximal and minimal value by circles. In order to make the results most readable we chose a color scale of $[0.5; 1.2]$ for separable problems and $[0.6; 1.7]$ for non-separable problems. The separable problems on the left half exhibit a performance loss consistent with Salomon's single-objective results. Performance dropped for almost all Cr smaller than 1. Non-separable WFG6 and WFG9 do not show such a systematic decrease. In some areas we even see an *increase* of performance.



(a) WFG4 (S-MM) (b) WFG7 (S-UM) (c) WFG6 (NS-UM) (d) WFG9 (NS-MM)

Fig. 4: $\frac{\text{Average RHV with a rotation of 5 degrees}}{\text{Average RHV without a rotation}}$

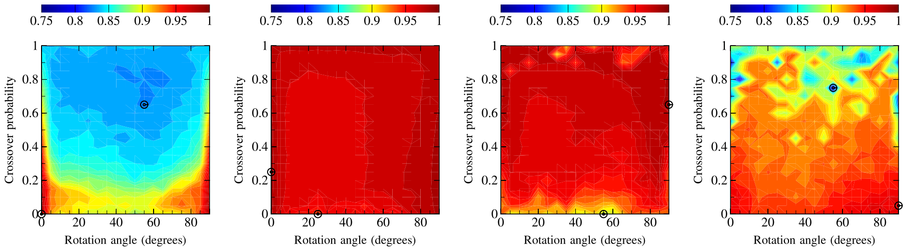


(a) WFG4 (S-MM) (b) WFG7 (S-UM) (c) WFG6 (NS-UM) (d) WFG9 (NS-MM)

Fig. 5: $\frac{\text{Average RHV with a rotation of 45 degrees}}{\text{Average RHV without a rotation}}$

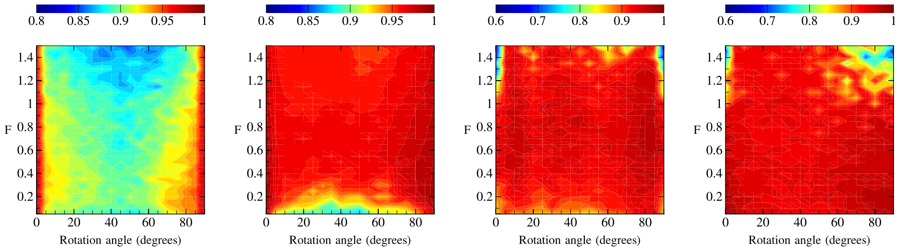
It seems that there is relatively little difference between the rotated and non-rotated data. These result may seem not as significant as Salomon's. However, there is an important methodological difference. When he mentions that the performance on the rotated benchmark is *six orders of magnitude* worse than the performance on the non-rotated benchmark ([9, p.273]), he means that *the minimal attained value* $2.65 \cdot 10^5$ is six orders of magnitude worse in absolute numbers. But the *value at initialization* was three orders of magnitude greater yet. This means that both algorithms started somewhere near $2.65 \cdot 10^8$ and the non-rotated one progressed to $2.65 \cdot 10^{-1}$ while the rotated one progressed to $2.65 \cdot 10^5$. In terms of relative hypervolume, this would be a very small difference [5]. In order to provide a scale-independent comparison, we compared all data using a two-tailed Wilcoxon signed rank test at a significance level of 0.05. For separable problems in Figures 4 and 5 we separate the parameter space with a dashed line into two areas. The area on the right is such that the rotated and non-rotated data is not significantly different, while on the left there is a *significant decrease in performance*. The data for non-separable problems contains areas of both significant decrease and significant decrease, as well as areas with no significant difference so in this case the separation cannot be plotted so comprendiously.

[5] Assuming that the minimum of the given function is 0, the difference would be on the order of 10^{-3}.



(a) WFG4 (S-MM) (b) WFG7 (S-UM) (c) WFG6 (NS-UM) (d) WFG9 (NS-MM)

Fig. 6: Average RHV for F = 0.5



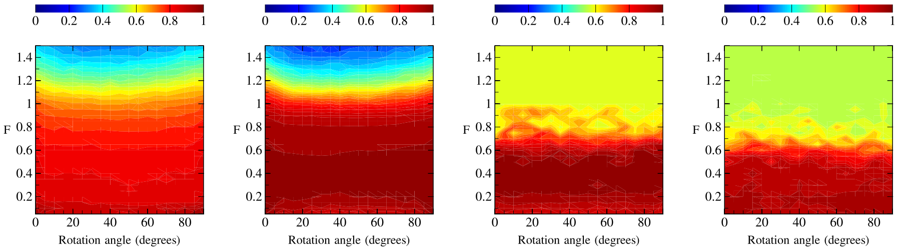
(a) WFG4 (S-MM) (b) WFG7 (S-UM) (c) WFG6 (NS-UM) (d) WFG9 (NS-MM)

Fig. 7: Average RHV for Cr = 0.1

The effects are more visible with 45 degree rotation in Figure 5. Again, there is a *systematic* decrease in performance for the separable problems for Cr < 1. However, this decrease does not imply that $Cr = 1$ is a good choice. Looking at Figures 2 and 3, we see that $Cr = 1$ is a consistently bad choice for the *multi-modal* problems WFG4 and WFG9.

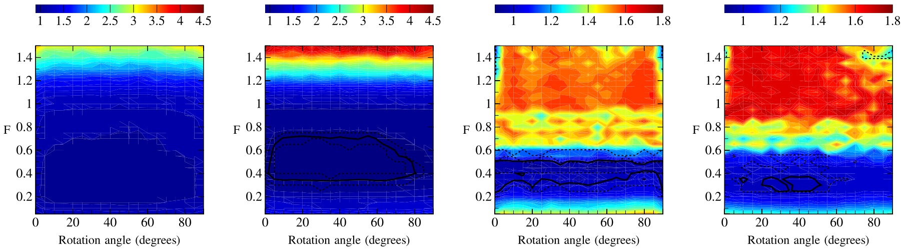
4.2 Fixed F

In Figures 2 and 3 we see that F = 0.5 is compatible with many different values of Cr and achieves consistently good performance. The average RHV for F = 0.5 is shown in Figure 6. For *multi-modal* problems WFG4 and WFG9, *very low* values of Cr are *consistently* good for all studied rotations, while for *uni-modal* problems WFG6 and WFG7 *big* values of Cr yield a *consistently* good performance. On the other hand, poor performance is achieved with *big* values of Cr for multi-modal problems and *small* values for uni-modal problems. The data for WFG4 and WFG9 suggests that the exceptionally good performance of a small Cr setting does *not* have to be balanced by an exceptionally bad performance after the problem is rotated. Based on the observation from Figure 6 we see that for each problem either Cr = 0.1 or Cr = 0.9 perform well through the observed spectrum of rotations.



(a) WFG4 (S-MM) (b) WFG7 (S-UM) (c) WFG6 (NS-UM) (d) WFG9 (NS-MM)

Fig. 8: Average RHV for Cr = 0.9



(a) WFG4 (S-MM) (b) WFG7 (S-UM) (c) WFG6 (NS-UM) (d) WFG9 (NS-MM)

Fig. 9: $\frac{\text{Average RHV with Cr = 0.1}}{\text{Average RHV with Cr = 1}}$

4.3 Fixed Cr

In Figures 7 and 8 we see data with a fixed value of Cr = 0.1 and Cr = 0.9 respectively. For Cr = 0.1 the regions with the best performance are for rotations which are either close to 0 or 90 degrees. This is true also for non-separable problems, but it is more visible for separable problems. The data for Cr = 0.9 seems different. The choice of Cr close to 1 means that the algorithm is nearly rotationally invariant. The gained robustness with respect to coordinate rotation is balanced by lost robustness in the choice of F. Almost in all cases the interval with favorable values of F became shorter.

We see that for values of Cr < 1 there is a performance loss when the coordinate axes are rotated, but does the performance drop bellow that of a rotationally invariant choice of Cr = 1? The data supporting a *negative* answer is presented in Figure 9. Here we divided the average RHV with Cr = 0.1 by the average RHV attained with a rotationally invariant Cr = 1. The interpretation of the dashed and full contour lines is the same as for Figures 4 and 5. For WFG4, the setting of Cr = 0.1 statistically significantly outperformed Cr = 1 for *all rotations* and *all* values of F. This means a definitive *negative* answer to our main question. The results are similar for the second multi-modal problem WFG9. Here we see a small region in which the data for Cr = 0.1 and Cr = 1 are not significantly different and Cr = 1 is significantly better in a few isolated cases. The unimodal problems on the other hand show that Cr = 1 is significantly better for most rotations and for the best performing values of F.

5 Conclusion

In this work we showed how the behavior of the differential evolution algorithm on bi-objective problems changes when the coordinate axes of the decision space are rotated. Our findings show that the change is significant even for small rotations. There is a consistent *drop* in performance on *separable* problems while the qualitative properties of the change for *non-separable* problems are much less predictable. Unexpectedly, for *multi-modal* problems, *low* values of crossover probability perform better through the observed spectrum of rotations. As a future work we propose to see if this holds for problems other than the ones we studied and if this is the case, to find the cause of this behavior.

References

1. R. Denysiuk, L. Costa, and I. Esprito Santo. Many-objective Optimization Using Differential Evolution with Variable-wise Mutation Restriction. In *Proceedings of the Conference on Genetic and Evolutionary Computation*, pages 591–598, New York, NY, 2013. ACM.
2. N. Hansen, R. Ros, N. Mauny, M. Schoenauer, and A. Auger. Impacts of Invariance in Search: When CMA-ES and PSO Face Ill-Conditioned and Non-Separable Problems. *Applied Soft Computing*, 11:5755–5769, 2011.
3. S. Huband, P. Hingston, L. Barone, and L. While. A Review of Multiobjective Test Problems and a Scalable Test Problem Toolkit. *IEEE Transactions on Evolutionary Computation*, 10(5):477–506, 2006.
4. S. Kukkonen and K. Deb. A Fast and Effective Method for Pruning of Non-dominated Solutions in Many-Objective Problems. In *Parallel Problem Solving from Nature - PPSN IX*, volume 4193, chapter Lecture Notes in Computer Science, pages 553–562. Springer Berlin Heidelberg, 2006.
5. S. Kukkonen and J. Lampinen. An Empirical Study of Control Parameters for The Third Version of Generalized Differential Evolution (GDE3). In *IEEE Congress on Evolutionary Computation*, pages 2002–2009, 2006.
6. K. Price, R. M. Storn, and J. A. Lampinen. *Differential Evolution: A Practical Approach to Global Optimization.* Springer, New York, NY, 2005.
7. T. Robič and B. Filipič. DEMO: Differential Evolution for Multiobjective Optimization. In *Evolutionary Multi-Criterion Optimization (EMO 2005)*, volume 3410, pages 520–533. Springer Berlin Heidelberg, 2005.
8. K. Saku. *Generalized Differential Evolution for Global Multi-Objective Optimization with Constraints.* PhD thesis, Lappeenranta Univ. of Technology, 2012.
9. R. Salomon. Re-evaluating Genetic Algorithm Performance Under Coordinate Rotation of Benchmark Functions. *Biosystems* , 39(3):263–278, 1996.
10. D. Zaharie. Critical Values for the Control Parameters of Differential Evolution Algorithm. In *Proceedings of MENDEL 2002*, 2002.
11. E. Zitzler. *Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications.* PhD thesis, Comput. Eng. Netw. Lab. Swiss Federal Instit. Technol. (ETH), Zurich, Switzerland, 1999.