

Capabilities of EMOA to Detect and Preserve Equivalent Pareto Subsets

Günter Rudolph, Boris Naujoks, and Mike Preuss

Universität Dortmund, Lehrstuhl für Algorithm Engineering
44221 Dortmund, Germany

{mike.preuss,boris.naujoks,guenter.rudolph}@uni-dortmund.de,
<http://ls11-www.cs.uni-dortmund.de>

Abstract. Recent works in evolutionary multiobjective optimization suggest to shift the focus from solely evaluating optimization success in the objective space to also taking the decision space into account. They indicate that this may be a) necessary to express the users requirements of obtaining distinct solutions (distinct Pareto set parts or subsets) of similar quality (comparable locations on the Pareto front) in real-world applications, and b) a demanding task for the currently most commonly used algorithms. We investigate if standard EMOA are able to detect and preserve equivalent Pareto subsets and develop an own special purpose EMOA that meets these requirements reliably.

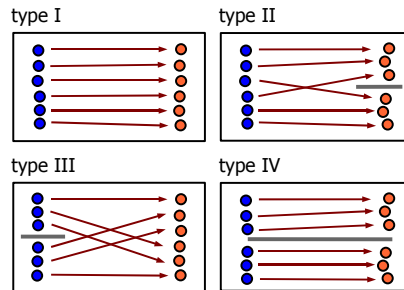
1 Introduction

Almost all publications about evolutionary multiobjective algorithms (EMOA) put their emphasis on approximating the Pareto front in the objective space whereas the relevance of an appropriate approximation of the Pareto set is widely neglected. The knowledge about the Pareto front is important for the product designer. But as soon as a solution in objective space has been selected it is important to know for the product engineer if there are alternative solutions in the decision space that lead to the same objective vector. Such Pareto-optimal solutions in decision space exist if there are symmetries in the objective function. This phenomenon occurs for example in the test problems considered by Chan and Ray [1] or Preuss et al. [2]. Basically, the Pareto set could be partitioned into subsets where the images of each subset are identical, i.e., each Pareto subset of this partition represents the entire Pareto front. Figure 1 illustrates and distinguishes different cases that may occur in multiobjective problems.

Apart from artificial test problems, there are of course real-world problems that exhibit such symmetries. For example, consider the problem of designing a proper diet for people with special needs. Besides taking into account nutrient and non-nutrient requirements, there are also aesthetic standards regarding shape, colors and others (cf. Seljak [3]). Of course, there are numerous ways to compile alternative but equally valuable meals that differ only in the exchange of some vegetables.

Here, we are interested in the capabilities of standard EMOA of detecting and/or preserving Pareto subsets of equivalent quality. A more detailed view of our aims and methods is given in section 2. For our analysis, we construct an artificial problem class that exploits symmetries in the objective function in an extreme manner along with various geometric transformations. The same blue print can be used to construct further test classes in future. This approach is presented in section 3, which is enriched with an experimental investigation of the problem hardness via *design of experiment* (DOE) methods. Section 4 evaluates standard EMOA and a special purpose EMOA on this problem class which leads to the observation that standard EMOA and even the special purpose EMOA do not provide fully satisfying results. Therefore, we develop a new EMOA approach that is based on the multistart technique along with several scalarization methods. We can show empirically that this approach delivers a reliable and accurate approximation of all Pareto subsets with equivalent quality. We finish with our conclusions in section 6.

Fig. 1: Different Pareto set and Pareto front type combinations: One Pareto set and one Pareto front (type I), one Pareto set and multiple Pareto front parts (type II), multiple Pareto subsets and one Pareto front (type III), and multiple Pareto subsets and Pareto front parts (type IV). Type III problems are rarely investigated, although they potentially provide multiple preimages for every objective vector of interest.



2 Aims and Methods

To investigate the behavior of EMOA and their operators in presence of multiple Pareto set parts (type III problems), we concentrate on three main questions:

- Which properties make these problems especially hard or simple for standard EMOA?
- What are the mechanisms in EMOA that lead to better or worse performance in terms of Pareto set preservation and Pareto front approximation?
- How can Pareto set preservation in EMOA be improved?

Obviously, standard performance measures for multiobjective optimization algorithms disregard how Pareto sets are dealt with; they only refer to population distributions in the objective space. We therefore define two simple new measures which require knowledge about Pareto subset numbers and locations and are thus not applicable to real-world application problems.

The formal definitions refer to a population P of points (ind) in decision space and a set S of Pareto subsets (set). The boolean function $\mathbf{near}(\text{ind}, \text{set})$ becomes true if the tested individual reaches the vicinity of the tested set. For determining when exactly this is the case, the concrete problem must be taken into account. VAR stands for the sample variance s^2 , determined to $s^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2$.

Covered sets (cs): The number of covered Pareto subsets (which comprise at least one individual in their vicinity).

$$\text{cs}(P, S) := |\{\text{set} \in S : \exists \text{ind} \in P, \mathbf{near}(\text{ind}, \text{set})\}| \quad (1)$$

Set population spread (sps): The standard deviation of the Pareto subset population counts (the number of individuals found on a Pareto subset).

$$\text{sps}(P, S) := \sqrt{\text{VAR}(\{\forall \text{set} \in S : |\text{ind} \in P, \mathbf{near}(\text{ind}, \text{set})|\})} \quad (2)$$

For measuring the Pareto front approximation quality of a population, we utilize the common S-metric (hypervolume). Furthermore, standard experiment layout and visualization techniques from the *design of experiments* (DOE) field (see Montgomery [4]) are employed.

3 A test-problem class: SYM-PART

In a previous work [2], a configurable type III test problem with two distinct Pareto sets, overlapping only in the decision space origin, has been investigated. These distinct Pareto sets were caused by the point symmetry of the bi-modal objective function. It is easy to see that such property entails loss of surjectiveness by creating two or more preimages of the optima and search points in their vicinity. As soon as at least the global optimum of one objective function (which is by definition part of the Pareto front of the resulting multiobjective function) is affected, multiple, possibly connected Pareto subsets emerge. In the following, we use this reasoning to construct SYM-PART (symmetrical parts) test problems with a controllable number of Pareto subsets, heavily relying on symmetry properties of the underlying singleobjective functions.

3.1 Construction of the test problems

Starting point is a very simple and well known test problem with two objectives and two-dimensional search space, namely,

$$f(x_1, x_2) = \begin{pmatrix} (x_1 + a)^2 + x_2^2 \\ (x_1 - a)^2 + x_2^2 \end{pmatrix} \quad (3)$$

for some $a > 0$. The Pareto set $\mathcal{X}^* = \{x \in \mathbb{R}^2 : x = (x_1, 0)'$ with $x_1 \in [-a, a]\}$ maps to the Pareto front $\mathcal{F}^* = f(\mathcal{X}^*) = \{z \in \mathbb{R}^2 : z = (4a^2\nu^2, 4a^2(1 -$

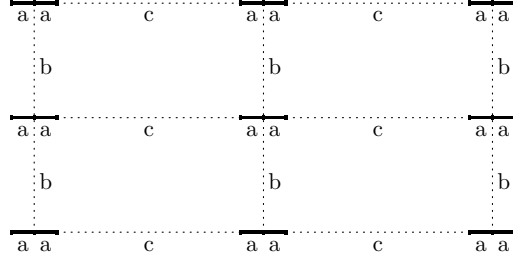


Fig. 2: Blue print of the initial test problem: Each subset of the Pareto set is a line of length $2a$. Parameter b specifies the vertical distance between neighboring Pareto subsets, whereas parameter c specifies the distance to the next Pareto subset on the horizontal line. Each Pareto subset maps to the same Pareto front.

$\nu)^2\}$ with $\nu \in (0, 1)$. Our idea is to translate the problem above to different regions in search space (see Fig. 2), such that each of these Pareto subsets are of equivalent quality since each Pareto subset maps to the same Pareto front.

For this purpose we define test problem (3) only in a certain neighborhood. Such a neighborhood will be called *tile* hereinafter (see Fig. 3).

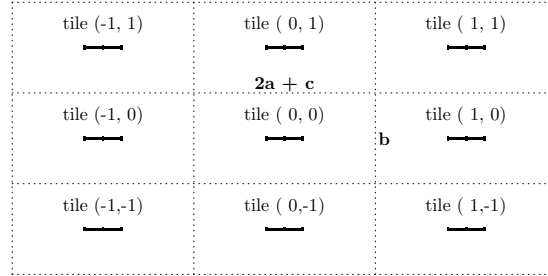


Fig. 3: Tile pattern for function (3) translated to tiles (i, j) that are defined by a rectangular region with width $2a + c$ and height b . Here, (i, j) denotes the tile identifier.

The tile identifiers are determined via

$$\hat{t}_1 = \text{sgn}(x_1) \times \left\lceil \frac{|x_1| - (a + \frac{c}{2})}{2a + c} \right\rceil \quad (4)$$

$$\hat{t}_2 = \text{sgn}(x_2) \times \left\lceil \frac{|x_2| - \frac{b}{2}}{b} \right\rceil \quad (5)$$

where a, b and c are the parameters for specifying the tile pattern. We restrict the problem to 9 tiles, i.e., the tile identifiers t_i only attain values in $\{-1, 0, 1\}$ by using the relation $t_i = \text{sgn}(\hat{t}_i) \times \min\{|\hat{t}_i|, 1\}$. Now we are in the position to

define the first test problem instance:

$$f^{(1)}(x_1, x_2) = f(x_1 - t_1(c + 2a), x_2 - t_2 b)$$

The second test problem instance requires that x is rotated by $\omega = 45^\circ$ via

$$r(x) = \begin{pmatrix} \cos \omega & -\sin \omega \\ \sin \omega & \cos \omega \end{pmatrix} x$$

before calculating the tile identifiers t_1, t_2 in (4) and (5). This leads to (see Fig. 4, left)

$$f^{(2)}(x_1, x_2) = f^{(1)}(r_1(x), r_2(x)).$$

Finally, we add a transformation that distorts the shape of the Pareto subsets:

$$d(x_1, x_2) = x_1 \times \left(\frac{x_2 - L + \varepsilon}{U - L} \right)^{-1}$$

for some small $\varepsilon > 0$ and where U and L denote the upper and lower bound of the search space, respectively. When transforming x_1 prior to calculating the tile identifiers, the third test problem instance is defined by (see Fig. 4, right)

$$f^{(3)}(x_1, x_2) = f^{(2)}(d(x_1, x_2), x_2).$$

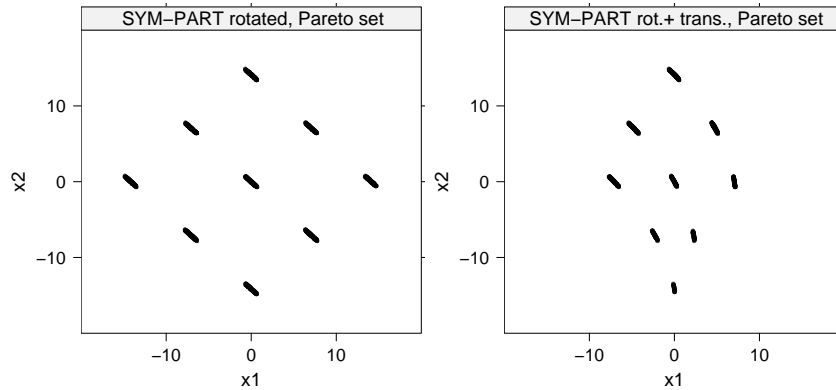


Fig. 4: Empirically detected (randomly enumerated) Pareto sets of SYM-PART test problems instances 2 and 3 (instance 1 refers to the original problem depicted in Fig. 3). Instance 2: 45° rotation, no transformation (left), instance 3: 45° rotation with transformation (right). Note that Pareto subset sizes differ here.

Needless to say, we are aware of the weaknesses of these test instances since they exploit only one type of symmetry and since they are defined only for two dimensions in search and objective space. But as can be seen shortly, these simple test problems can be used to demonstrate interesting phenomena occurring in standard EMOA and some special purpose EMOA presented below.

3.2 Experimental investigation of problem hardness

In the following sections, several EMOA are tested for their ability to reach and preserve many or all existing Pareto subsets. It is therefore necessary to establish differently difficult problem instances of the SYM-PART problem class. In particular, the three problem instances developed in §3.1 shall be assessed. Apart from the fact that we do not employ any evolutionary algorithm but simple, deterministic, grid-based search methods, exploring the effect of problem modifications onto optimization methods is related to the approach of Langdon and Poli [5].

Experiment 1 relies on the utilization of *design of experiments* (DOE) techniques as first introduced by Fisher [6]. The controllable input variables or factors—in this case problem properties—are varied systematically in discrete levels. Observing the resulting performance changes then enables estimating the impact of single properties (main effects) and combined properties (interaction effects). An experimental layout that requires to actually test all possible factor level combinations is called a fully factorial design. For larger numbers of factors, one often uses fractional factorial designs. These reduce the number of runs by ignoring certain factor level combinations at the expense of explanatory power regarding higher-order interaction effects. For a more thorough introduction into DOE methods we refer to standard textbooks (e.g. Montgomery [4]).

Experiment 1: Problem hardness of different SYM-PART configurations.

Pre-experimental planning: First experiments revealed that a standard operator/value NSGA2 (see Tab. 3) performs reasonably well in preserving Pareto sets over a long time (30,000 evaluations). Replacing search operators or parameter values seems to weaken this ability. The NSGA2 is therefore chosen as constant base algorithm when modifying the treated problem.

Task: Detect which SYM-PART modifications have a large impact on the ability of an EMOA to discover and preserve as many Pareto sets as possible. Recommend few considerably different SYM-PART instances for further use.

Setup: We apply a full factorial design: NSGA2 is run with 30 repeats on each factor level combination (16). Low and high factor levels are given in Tab. 1. Bounds refers to the rectangular search space bounds, shift stands for translation of the whole tile structure relative to the origin, rotation and transformation are as stated in §3.1.

Results/Visualization: The mean number of covered sets (cs) and the set population spread (sps) are used to compute main and interaction effects. These

Table 1: SYM-PART problem designs, made of combinations of 4 factors, each of which has a low (left) and high (right) level. Choosing all 4 low levels results in the original problem as described in §3.1.

parameter	bounds (L:U)	shift vs. origin	rotation angle	transformation
factor levels	−50:50/−20:20	(0, 0)/(2, 2)	0° / 45°	no / yes

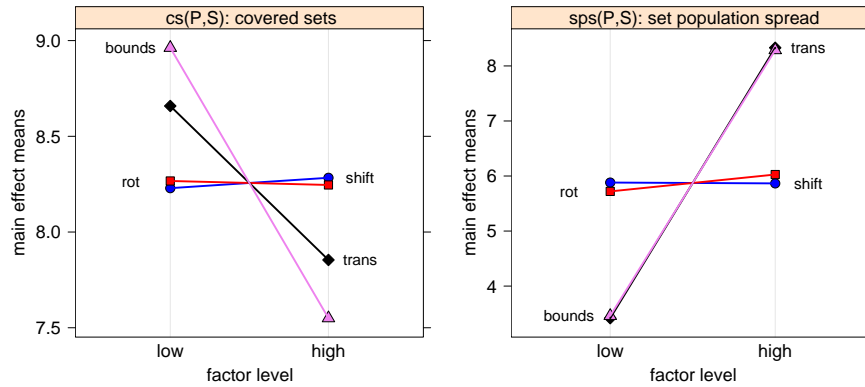


Fig. 5: DOE main effects, original mean values, without adjustment towards the average. As the standard deviations of the observed $cs(P,S)$ values are almost 1 (up to ≈ 5 for the largest values of $sps(P,S)$), all but the largest two effects are insignificant.

are depicted in Fig. 5 and Fig. 6, respectively. Due to space limitations and to enhance comparability, all effects are plotted into one diagram, thereby deviating from standard DOE practice. Higher-order interaction effects (of more than two factors) are disregarded.

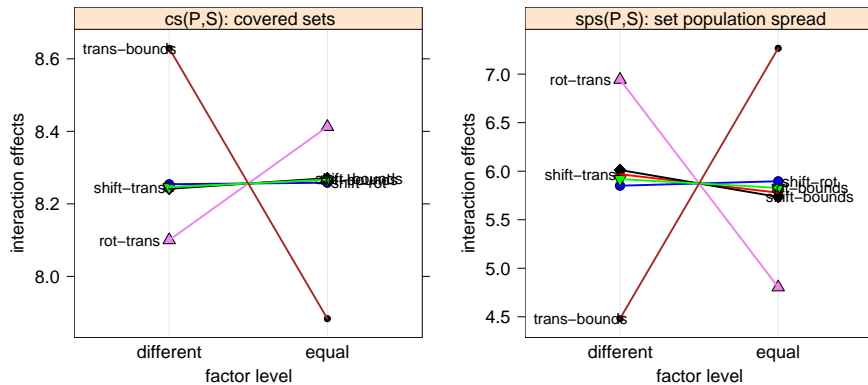


Fig. 6: DOE interaction effects, as Fig. 5 at the original location, but differently scaled, without adjustment. All except the trans-bound and rot-trans interactions are insignificant due to the high variance level.

Observations: The strongest main effects are caused by the transformation and the extent of decision space bounds (trans and bounds in Fig. 5). Measures $cs(P,S)$ and $sps(P,S)$ return consistent values: For smaller decision spaces, less Pareto subsets are kept, and the spread of set populations increases. The transformation has a similar effect and obviously makes the problem harder if switched

on. Shift and rotation apparently do not affect problem hardness. The interaction effect plots Fig. 6 document that only two interactions need to be considered: Trans-bound and rot-trans. Both interaction effects are much weaker than the important main effects. Whereas trans-bound signals harder problems if both factors are either set to their low or to their high levels, rot-trans points into the other direction. If only rotation or only transformation is switched on, the problem appears to be harder than if both are on or off.

Discussion: Surprisingly, changing decision space bounds has a large effect on performance in terms of $cs(P, S)$ and $sps(P, S)$. If the relative amount of search space that must be covered for placing individuals in all Pareto subsets approaches 1, the EMOA gets more and more difficulties. We attribute this behavior at least in part to the *polynomial mutation* (PM) operator which uses the upper and lower bounds for adjusting its step size distribution. We must however state that the PM operator works reasonably well even under very tight bounds around the Pareto subsets. As setting the bounds to the high factor level (-20/20) greatly increases problem difficulty, we consider only these in the following.

Dissecting the impact of the 4 possible combinations of rotation and transformation leads to an unexpected order of increasing hardness: $\neg\text{rot} \wedge \neg\text{trans}$ (mean/stddev(cs)=8.83/0.33) < $\text{rot} \wedge \neg\text{trans}$ (8.49/0.49) < $\text{rot} \wedge \text{trans}$ (8/0.60) < $\neg\text{rot} \wedge \text{trans}$ (7.71/0.76). To keep the number of problem instances for further testing as low as possible, we select only 3 of these, namely the simple one ($\neg\text{rot} \wedge \neg\text{trans}$), the rotated one ($\text{rot} \wedge \neg\text{trans}$), and the rotated and transformed one ($\text{rot} \wedge \text{trans}$). Instead of the latter, one could also chose the not rotated but transformed instance. However, we refrained from doing so because the difference between these two is rather small, and it is currently not clear why the instance without rotation may be more difficult.

4 Evaluation of Standard EMOA on SYM-PART

Compared to §3.2, we now follow the opposite approach and test several common EMOA on the three previously selected SYM-PART problem instances.

Experiment 2: Investigate convergence/diversity tradeoff for different EMOA.

Pre-experimental planning: First results confirmed the expected behavior: Standard techniques do not perform well even on the simplest instance of the SYM-PART problem. The algorithms only kept a very limited number of tiles ($cs(P, S)$).

Later, it was discovered that this unwanted behavior was seemingly caused by adaptive mutation featuring $n = 2$ step sizes [7]. After changing the variation operator to polynomial mutation [8], which became the standard mutation operator within EMOA in recent years, the quality of results increased significantly. This is indicated by the average number of tiles preserved by different EMOA, in turn using the two mentioned mutation operators. Mean values for $cs(P, S)$ are given next to the corresponding standard deviations (in brackets) in Tab. 2. As

polynomial mutation performed much better, this operator was applied in the investigation of different EMOA on all instances of the SYM-PART problem.

Table 2: Test of standard EMOA with different mutation operators, namely polynomial mutation (PM) and adaptive mutation with two step sizes (AM). The values give average $cs(P,S)$ values of 18 runs with 10,000 evaluations each (standard deviations are given in brackets).

algorithm	AM	PM
NSGA-II	1.65 (0.745)	8.61 (0.608)
SPEA2	1.94 (0.873)	8.94 (0.236)

Task: The performance of EMOA is to be tested on all instances of the SYM-PART problem. More detailedly, we look for drawbacks of the standard techniques in contrast to an algorithm that is explicitly developed to keep diversity in solution space as well as in decision space. Are the algorithms able to discover new tiles and can they keep the new tiles for the rest of the optimization run?

Setup: We invoke two standard techniques next to a new development within the field. The Pisa framework¹ is used to conduct the referred optimization runs with the standard techniques. Here, all specifications of the SYM-PART problem have been implemented as a variator, which can be optimized with respect to different objectives and multiple selectors. Among the set of available selectors, NSGA-II and SPEA2 are chosen, because these appear to be the currently most well-known and commonly used algorithms in the field [8, 9]. Additionally, the more recent KP1 by Chan and Ray [1] is tested.

KP1 was designed for maintaining diversity in decision space as well as in objective space. Therefore, two criteria to measure the diversity of solutions in the corresponding spaces are defined and applied in each generation. These are dominated hypervolume of each individual for the objective space and a neighborhood counting approach for the decision space. Both are described in detail by Chan and Ray [1]. The OMNI-Optimizer by Deb et al. [10] considers only one of such measurements in the different space at a time and is not included in this study.

The parameters of the variation operators are set to standard values, i.e. SBX and PM with distribution indices $\eta_c = 15$ and $\eta_m = 20$, respectively. Crossover and mutation probability are set to one. Selection is performed using a (100 + 100) selection scheme for 300 generations in either cases, resulting in 30,000 fitness function evaluations per run (see Tab. 3).

The additional effort for a third algorithm in the study seems to be justified as the development aims of this algorithm directly address the difficulties of the chosen test problems.

¹ PISA - Platform and Programming Language Independent Interface for Search Algorithms, ETH Zurich, www.tik.ee.ethz.ch/pisa/

Table 3: Parameter setting for standard EMOA depicting mutation and crossover probabilities (mut.prop. and cross.prob.), distribution indices (η_m and η_c), and the selection scheme in use.

parameter	mut.prob.	η_m	cross.prob.	η_c	selection
value	1	20	1	15	(100 + 100)

Results/Visualization: Tab. 4 and 5 give average final results of the 30 runs performed for every algorithm on every instance of the SYM-PART problem. Tab. 4 more detailedly depicts the average number of tiles preserved by the indicated algorithm after 30,000 evaluations. The mean hypervolume received after the corresponding runs is contained in Tab. 5. But, these averaged values of the final results do not give evidence for the behavior of the different algorithms during the runs. This aspect is tackled in Fig. 7, where all repetitions of runs have been averaged generation by generation. For example, the upper left diagram of Fig. 7 depicts three curves, one for each instance of the SYM-PART problem. Each curve is generated averaging the results achieved after the first generation, the second one, up to the 300th one. The same holds for all other curves within all diagrams in Fig. 7. The middle row holds SPEA2 results while the lower one displays the results of KP1 by Chan and Ray. The upper row is dedicated to NSGA2 and the left column to the generation-wise averaged number of tiles kept as can be seen from the example above.

The right column gives the generation-wise average values of the dominated hypervolume. Here, the displayed area is shortened to the starting phase of the runs up to generation 50. This is done to highlight the interesting developments during the runs and implies that no major changes in the behavior take place after the depicted interval of the run. The final results of the averaged runs can be taken from Tab. 4 and 5 as described above.

Table 4: Test of different algorithms on all instances of the SYM-PART problem. The values give the average $cs(P, S)$ of 30 runs with 30,000 evaluations each (standard deviations are given in brackets).

algorithm	simple	rotated	rot.+trans.
NSGA-II	6.333 (1.446)	5.633 (1.450)	4.667 (1.124)
SPEA2	6.3 (1.022)	5.2 (1.157)	5 (1.364)
KP1	8.3 (1.290)	6.733 (1.818)	6.5 (0.9738)

Observations: With respect to the number of tiles kept, Tab. 4 shows the expected behavior of the algorithms within this study: The number of tiles kept decreasing with growing hardness of the considered instance of the SYM-PART problem. This means, most of the tiles are kept on SYM-PART 1 by all algorithms. Here, KP1 clearly outperforms the other algorithms keeping 8.3 of 9 tiles on average. This is the highest value achieved within all experiments. The lowest number of tiles is received for SYM-PART 3, the rotated and transformed instance and therefore the most difficult one. On this problem, NSGA2 receives

Table 5: Test of different algorithms on all instances of the SYM-PART problem. The values give the average dominated hypervolume after 30 runs with 30,000 evaluations each (standard deviations are given in brackets).

algorithm	simple	rotated	rot.+trans.
NSGA-II	22.254 (0.00353)	22.255 (0.00305)	22.254 (0.00358)
SPEA2	22.257 (0.00237)	22.257 (0.00243)	22.255 (0.00278)
KP1	22.241 (0.00712)	22.231 (0.00689)	22.220 (0.00781)

the lowest value achieved within all experiments (4.667). For all algorithms, the values for SYM-PART 2 are greater than the ones for SYM-PART 3 and smaller than the ones for SYM-PART 1. KP1 performs better than the other algorithms on all instances. Interestingly, NSGA2 is better than SPEA2 on SYM-PART 1 and SYM-PART 2, while SPEA2 performs better on SYM-PART 3.

The behavior of the algorithms changes when taking the dominated hypervolume into account (see Tab. 5). SPEA2 receives the best results on all instances, followed shortly by NSGA2. KP1 clearly achieves the worst values of dominated hypervolume on all instances. Furthermore, the values from this algorithm decrease with problem complexity. This behavior can not be observed for NSGA2 and SPEA2. Here, the largest dominated hypervolume is obtained on the rotated instance of SYM-PART, while the lowest values are achieved on the rotated and transformed SYM-PART 3.

More dramatic differences in the behavior of KP1 in contrast to NSGA2 and SPEA2 can be observed in the diagrams of Fig. 7 considering the average $cs(P, S)$ values per generation. In contrast to the behavior of KP1, NSGA2 and SPEA2 loose tiles during the averaged optimisation runs. KP1 first loses tiles as well but turns its behavior after about 20 generation on all three instances. Starting here, KP1 almost constantly captures tiles back. Interestingly, steps can be observed even in the averaged runs. This is due to increasing as well as decreasing $cs(P, S)$ values within single runs. As a consequence, also KP1 is not able to keep all newly discovered tiles for the rest of the run. Some are lost again after only a few generations. But, in contrast to NSGA2 and SPEA2, this algorithm is able to keep more tiles than get lost. This leads to the over all increasing number of tiles on average.

The curves depicting the hypervolume do not yield such interesting results. The values here increase rapidly to almost optimal values for all algorithms. More detailedly, NSGA2 and SPEA2 act almost comparable on SYM-PART 1 and SYM-PART 2. The dominated hypervolume increases a bit more slightly on SYM-PART 3. This also holds for KP1, where a more distinct difference can be observed between SYM-PART 1 and SYM-PART 2. Over all, the results for KP1 seem to converge to the almost optimal values for the run a bit faster. But, as can be seen from Tab. 5, these values are worse than the ones for SPEA2 and NSGA2.

Discussion: With respect to the course of the tiles kept, an important difference in the behavior of the algorithms is observed. While this course decreases for NSGA2 and SPEA2, it increases for KP1. The final conclusion that all but one

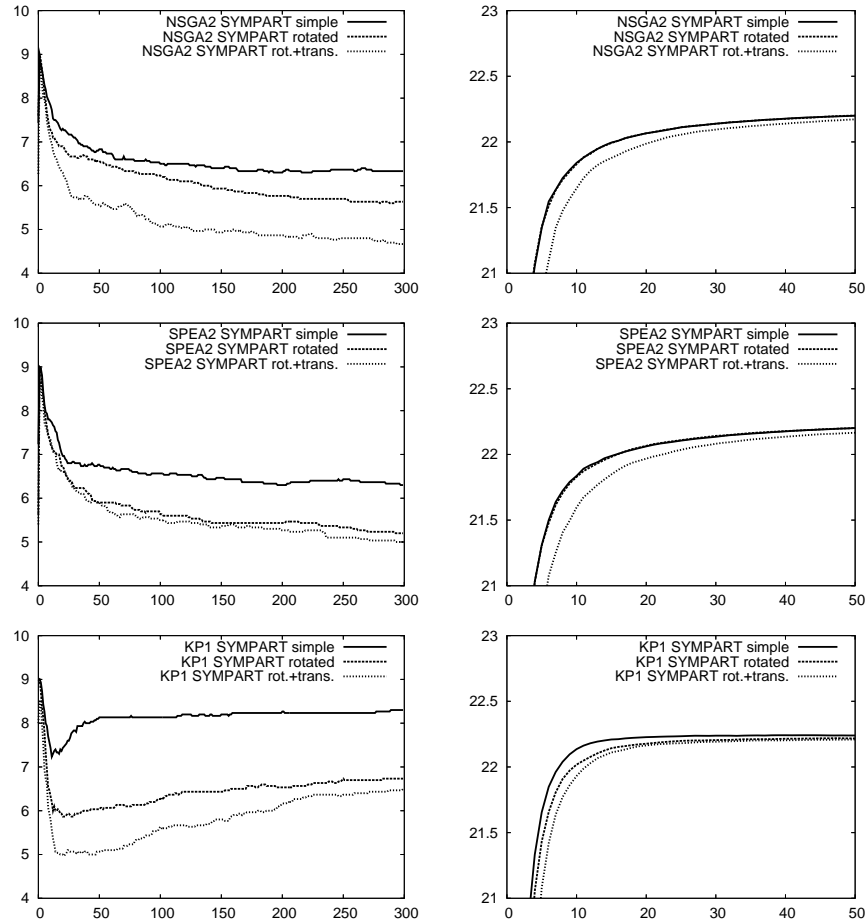


Fig. 7: Average runs of NSGA-II, SPEA2 and KP1 of Chan and Ray (labeled KP1) on all instances of the SYM-PART function. The left column presents the average $cs(P, S)$ values over the evaluations while the right one gives the average dominated hypervolume. The average runs have been received from 30 runs performed, 30,000 fitness function evaluations each. Only the first part up to 5,000 evaluations is presented in case of the hypervolume plot due to better observability of results.

tile are lost after more generations of NSGA2 and SPEA2 while all tiles are captured back using KP1 is not shown, but is an self-evident assumption.

The values for the dominated hypervolume reveal that the more tiles are kept, the less hypervolume is achieved. This leads to the assumption that both criteria are conflicting. The fact that no hypervolume is lost with increasing number of tiles in the KP1 runs contradicts this assumption. Therefore, KP1 can be stated to be the best algorithm within this study, although not dominating all the hypervolume the other algorithms do. This is due to KP1 preserving diversity

not only in the solution space, but also in the decision space. Considering both criteria, it would be better to stop the runs of the standard algorithms more early, i.e. after about 50 generations. At this point, they already dominate almost all possible hypervolume and occupy the highest number of tiles.

What is not tackled in this investigation is the distribution of individuals over tiles $sps(P, S)$. In the most comprehensive algorithms, the user would like the number of individuals to grow on newly occupied tiles. At the end of a run, a uniform distribution of individuals over all Pareto sets within tiles is aspired.

5 A Multistart Approach for Pareto Subset Detection

An alternative approach to detect and maintain several Pareto subsets of equivalent quality is provided by the multistart technique. The algorithm described here is still of experimental state but very promising. The main idea is as follows: We run a singleobjective optimizer for each objective function. Since the optimal solution of each objective function is Pareto-optimal we have a kind of anchor that can be used to approximate the associated Pareto subset successively by deploying some singleobjective optimizer repeatedly with different weights of the scalarized multiobjective function.

Let $f(x) = (f_1(x), \dots, f_d(x))$ be the objective function with $x \in \mathbb{R}^n$. At first, N runs with a standard $(1, \lambda)$ -ES are made for each of the d objectives. The ES stops if the standard deviation σ of the mutation operator is below some threshold $\delta > 0$. Each solution x^* is stored and annotated with the index of the objective function used: $(x^*, k) \in \mathbb{R}^n \times \{1, \dots, d\}$. Thus, we obtain $N \cdot d$ candidate solutions in this manner.

Suppose there are $s \in \mathbb{N}$ Pareto subsets with equivalent quality. If all Pareto subsets are hit by the multistart approach then we need only $s \cdot d$ anchor solutions as starting points of the singleobjective search with the scalarized multiobjective function to approximate all Pareto subsets. Since the number s of the equivalent Pareto subsets is unknown in general, we deploy an unsupervised clustering method to reduce the $N \cdot d$ candidate solutions to $s \cdot d$ anchor solutions required for the next step. Actually, it is possible to reduce the number of anchor solutions to s since we can apply the clustering method to the N solutions of each objective separately (recall that we have annotated each candidate solution with the index of the objective function used). Since the different objective functions may be of varying difficulty for the optimization, we can use the d outcomes of the clustering method as a consistency check. This idea, however, is currently not implemented. We simply cluster the candidate solutions of the objective function with index 1 and proceed with \hat{s} estimated anchor solutions.

The scalarization used in the sequel is known as the *weighted Tchebycheff method* [11]: The multiobjective function $f : \mathbb{R}^n \rightarrow \mathbb{R}^d$ is scalarized via

$$f^{<s>}(x) = \max_{i=1, \dots, d} \{w_i |f_i(x) - u_i^*|\}$$

where $u^* \in \mathbb{R}^d$ is the *utopian solution*. Since we have made N singleobjective optimizations of each objective $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$ in the first phase of our algo-

rithm, we have obtained an accurate estimator of the *ideal solution* z^* with $z_i^* = \min\{f_i(x) : x \in \mathbb{R}^n\}$ for $i = 1, \dots, d$. As a consequence, we may set $u_i^* = z_i^* - 1$ to get a valid utopian solution required for the weighted Tchebycheff method (WTM). We have chosen WTM because of its ability to find also solutions whose images are on a *concave* Pareto front. Needless to say, here we tacitly assume that the Pareto subsets are connected.

The user may choose how many representatives of each Pareto subset are desired. Suppose we like to obtain k representatives. Then for each of the \hat{s} anchor solutions x^* we start a standard $(1, \lambda)$ -ES with initial $\sigma_0 = 10\delta$, seeding point x^* , and weights that cover all possible weight assignments with maximal uniformity. In case of $d = 2$ objectives the weights are given by $w_1 = j/(k-1)$ and $w_2 = 1 - w_1$ for $j = 1, \dots, k-1$. Notice that the anchor solution x^* is used as initial parent of the ES for $j = 1$ only. The best solution found in this run serves as initial parent for $j = 2$. And so forth until $j = k-1$. In this vein, we finally arrive at an approximation of all Pareto subsets that were detected in the first phase of the algorithm.

For an assessment of this approach, we made some experiments for the three test instances introduced previously. The parametrization was as follows: $\lambda = 5$, $N = 50$, $\delta = 10^{-5}$, $k = 10$. The initialization of the ES in the first phase used $\sigma_0 = 20/6$ and the starting point was sampled uniformly from the region $[-20, 20]^2$.

Each run out of 30 in total detected the 9 Pareto subsets reliably and approximated the Pareto subset with high accuracy. In the first phase each run of the $(1, \lambda)$ -EA stops on average in less than 60 generations. Thus, we required less than $60 \times \lambda \times N \times d = 30,000$ function evaluations of the *single*-objective functions, which is equivalent to 15,000 function evaluations of the multiobjective function. The second phase (clustering) does not evaluate the objective function. The third phase required less than 5,000 function evaluations of the scalarized multiobjective function. Thus, this approach required less than the equivalent of 20,000 multiobjective function evaluations for a reliable and accurate approximation of all Pareto subsets for all test instances. Figure 8 shows typical results for the three test instances.

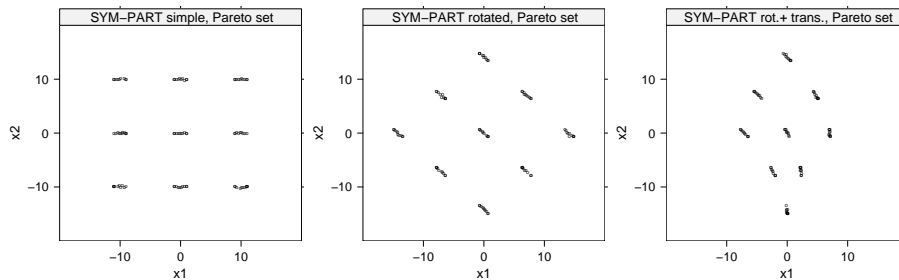


Fig. 8: Typical runs of the multistart approach on all instances of the SYM-PART problem (from left to right: instances 1, 2, and 3, as described in §3.1).

6 Conclusions and Future Work

We have shown that standard EMOA are not able to reliably detect and/or preserve all Pareto subsets of equivalent quality. This is not surprising as they have not been designed for this purpose. Moreover, this property is not required in some cases. But if we need this property we have to deploy special purpose EMOA. We have tested one such EMOA given in the literature and we have developed another EMOA that is based on a multistart approach which meets our requirements. It is imaginable that EMOA with niching can be successful in this case, too. But this analysis remains for future research, as well as the development of additional problem classes that exploit different types of symmetries and that are defined in higher-dimensional decision and objective spaces.

Acknowledgements

This work was supported by the *Deutsche Forschungsgemeinschaft (DFG)* as part of the *Collaborative Research Center 'Computational Intelligence' (SFB 531)*, the research project SCHW 361/15-1 "*Ein Verfahren zur Optimierung von aus Mehrkomponenten bestehenden Schiffsantrieben*", as well as project grant no. 252441, "*Mehrkriterielle Struktur- und Parameteroptimierung verfahrenstechnischer Prozesse mit evolutionären Algorithmen am Beispiel gewinnorientierter unscharfer destillativer Trennprozesse*".

References

1. Chan, K.P., Ray, T.: An Evolutionary Algorithm to Maintain Diversity in the Parametric and the Objective Space. In: Proceedings of IEEE International Conference on Computational Robotics and Autonomous Systems (CIRAS 2005), Centre for Intelligent Control, National University of Singapore (2005) ISSN: 0219-6131.
2. Preuss, M., Naujoks, B., Rudolph, G.: Pareto Set and EMOA Behavior for Simple Multimodal Multiobjective Functions. In Runarsson, T.P., et al., eds.: Parallel Problem Solving from Nature (PPSN IX). Volume 4193 of Lecture Notes in Computer Science., Springer, Berlin (2006) 513–522
3. Seljak, B.K.: Dietary Menu Planning by Evolutionary Computation. In Filipic, B., Silc, J., eds.: Bioinspired Optimization Methods and their Applications (BIOMA 2006), Jozef Stefan Institute, Ljubljana, Slovenia (2006) 87–98
4. Montgomery, D.C.: Design and Analysis of Experiments. 5th edn. Wiley, New York (2001)
5. Langdon, W.B., Poli, R.: Evolving problems to learn about particle swarm and other optimisers. In McKay, B., et al., eds.: Congress on Evolutionary Computation (CEC'05). Volume 1., IEEE Press, Piscataway NJ (2005) 81–88
6. Fisher, R.A.: The Design of Experiments. Oliver and Boyd, Edinburgh (1935)
7. Beyer, H.G., Schwefel, H.P.: Evolution strategies – A comprehensive introduction. *Natural Computing* **1** (2002) 3–52
8. Deb, K.: Multi-Objective Optimization using Evolutionary Algorithms. Wiley, Chichester, UK (2001)

9. Coello Coello, C.A., Van Veldhuizen, D.A., Lamont, G.B.: Evolutionary Algorithms for Solving Multi-Objective Problems. Kluwer, New York (2002)
10. Deb, K., Tiwari, S.: Omni-optimizer: A Procedure for Single and Multi-objective Optimization. In Coello Coello, C.A., et al., eds.: Evolutionary Multi-Criterion Optimization (EMO 2005). Volume 3410 of Lecture Notes in Computer Science., Springer, Berlin (2005) 47–61
11. Steuer, R.E.: Multiple Criteria Optimization: Theory, Computation, and Application. Wiley series in probability and mathematical statistics. Wiley, New York (1986)