technische universität
dortmund

Diploma thesis

Surface Reconstruction with Diversity-Preserving Multi-Objective Differential Evolution

Matthias Jung
April 14, 2009

Supervisors:
Prof. Dr. Günter Rudolph
Prof. Dr.-Ing. Dirk Biermann

Department of Computer Science
Algorithm Engineering (Ls11)
Technische Universität Dortmund
http://ls11-www.cs.uni-dortmund.de

Institute of Machining Technology
Technische Universität Dortmund
http://www.isf.de

Contents

List of symbols

Abbreviations

$\#$	number of
$\ll, \gg$	Much smaller, much bigger
$\approx$	Approximately
CD	Crowding-distance
HV	Hypervolume

General notations

$x[]$	Array notation
$x[i]$	Value at position i in array x

Evolutionary algorithms related

D	Number of decision variables
M	Number of objectives
G	Number of generations to compute
$\vec{x} = (x_1, ..., x_D)$	Vector of decision variables (individual)
$f(\vec{x}) = (f_1(\vec{x})), ..., f_M(\vec{x}))$	Fitness vector of individual $\vec{x}$
p	Vector of coordinates (point)
$S = \{p_1, ...\}$	Set of points
P	Population (set of individuals)
$\mathcal{F}_i$	ith front
$\mathcal{F} = \{\mathcal{F}_1, ...\}$	Partition of a set into fronts
μ	Population size
λ	Number of children created per generation
F	Step-size factor (differential evolution)
CR	Crossover probability (differential evolution)
L	Local search probability (cluster-based algorithms)
C	Number of clusters (HACM number criterion)
p_m	Mutation probability per decision variable (PM)
p_c	Crossover probability per decision variable (SBX)
η_m	Distribution index of the mutation operator (PM)
η_c	Distribution index of the crossover operator (SBX)

Chapter 1

Introduction

This work proposes a multi-objective evolutionary algorithm, that preserves diversity in decision space while also reaching good solutions, even if only few function evaluations can be calculated. To investigate its performance, the algorithm is evaluated on several test problems and a real-world mechanical engineering application.

Motivation

The Institute of Machining Technology (ISF) uses 3D models of real objects in both CAD programs and for simulation purposes. Such a 3D model is obtained using the following steps:

1. The object is digitized with a 3D scanner. As a result, a large number of points located on the surface of the object is determined.

2. An efficient mathematical description based on non-uniform rational B-splines [1] (NURBS) is calculated from those scan points.

The transformation of the scanned points into a mathematical description (i.e. CAD model) of the surface is called surface reconstruction [2]. The NURBS surface is to be optimized regarding several objectives, which represent desirable surface properties common to many work pieces in mechanical engineering. Because the desirable properties of a reconstructed surface like quality of the approximation and smoothness of the surface are (partially) not correlative, there are a lot of formally incomparable, Pareto optimal solutions, which represent surfaces with different control structure and appearance. Presently two approaches are implemented at the ISF, a single-objective deterministic algorithm and the multi-objective evolutionary algorithm SMS-EMOA [3]. Regarding the objectives above, both currently implemented algorithms only cover a small part of the Pareto front.

In order to achieve a greater variety of surfaces with diverse appearance, the solutions have to feature different control nets, so the diversity in decision space needs to be improved. Because there are many scan points required to appropriately describe the surfaces, the evaluation of the objective functions takes a lot of computation time. Thus, only about 20000 function evaluations can be calculated in a reasonable amount of time and additional performance enhancements are important to reduce the computation time.

Multi-objective differential evolution has proven to be very successful with other applications, because it combines desirable invariance features with fast convergence towards good solutions and low computing time. Differential evolution is also known to be a good variation operator for geometrical problems [4]. Applying multi-objective differential evolution to surface reconstruction therefore seems promising.

This diploma thesis aims to assess, analyze and enhance the GDE3 algorithm [5] (Generalized Differential Evolution version 3) regarding its application to surface reconstruction. Diversity in decision space on the one hand is increased by creating the initial population differently and by modifying the selection of GDE3, while performance on the other hand is improved by using the dominated hypervolume [3] as secondary selection criterion, employing a superior variation operator and exploring a good set of parameters for the algorithm designed. The improvements are assessed separately by means of representative test problems. To find out whether the determined performance on the test function is representative for surface reconstruction as well, the algorithms with

- best performance,
- best diversity and
- worst performance

are analyzed and compared to the starting point GDE3, the previously used SMS-EMOA and the well known NSGA-II [6] regarding performance on the surface reconstruction problem.

Structure

Chapter 2 describes the basic principles of evolutionary algorithms, multi-objective optimization and (experimental) performance assessment of such algorithms. Also current approaches for surface reconstruction and the data structures involved are presented. In the third chapter the proposed enhancements are evaluated on some test problems, while chapter 4 shows and analyzes the performance of selected algorithms on the problem of surface reconstruction. The last chapter provides a short summary and some ideas for further research.

Chapter 2

Related research

2.1 Evolutionary algorithms

2.1.1 Basic idea

Evolutionary algorithms are general, randomized search heuristics inspired by biological evolution (search heuristics are optimization algorithms, expected to work well on a specific problem in many cases). Three major branches of this kind of strategy have been developed in the 1960's and coexisted ever since:

- Evolutionary Programming (EP) [7]
- Evolutionary Strategies (ES) [8, 9]
- Genetic Algorithms (GA) [10]

In all three concepts the optimization process consists of three main episodes/phases.

- Variation (reproduction) derives a new solution based upon already known solutions.
- Evaluation assesses the newly created solution regarding an objective.
- Selection decides whether to keep the solution or delete it.

An initial set of solutions is required for the algorithm to work, these initial solutions can be e.g. determined at random or consist of already known solutions. Analog to biological evolution, the optimization of an evolutionary algorithm consists of small, progressive enhancements and basically implements a learning trial-and-error process. Since evolutionary algorithms are general search heuristics, they can be employed for numerous optimization tasks as a black box optimizer, hence with only slight modification effort such an algorithm can be used to solve numerous problems. There is a drawback - as the algorithm itself does not make use of any special knowledge on the problem it is trying to solve, there is no guarantee, that it will create decent results in any case. In contrast to the black-box

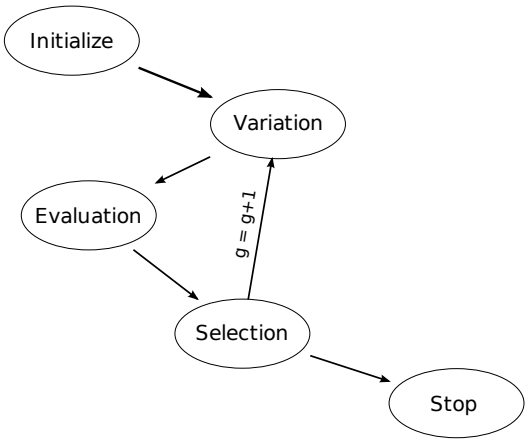


Figure 2.1: Workflow of an evolutionary algorithm

optimization scenario, algorithms tailored for a specific problem perform better, because special knowledge about the problem is incorporated. Evolutionary algorithms should therefore only be used for tasks, where the problem to solve is not very well understood and therefore only few knowledge about the structure of the problem is available, which could lead to an algorithm specifically designed for the task at hand or in cases where the design of a new algorithm is not possible for other reasons (e.g. the developer has no time to design an entirely new algorithm). For many real-world problems, black-box optimizers are a reasonable choice according to the given criteria. The way the recombination phase proceeds is determined by variation operators. A variation operator is the sub-routine telling the evolutionary algorithm in which way a new, possibly better solution is created from the set of already known solutions. To optimize an evolutionary algorithm for a specific problem, the variation operator can be (re-)designed using knowledge about the problem. For problems, where the decision variables are encoded into real numbers, a variation operating on the manipulation of the bit strings is not feasible. Commonly used variation operators for these cases are e.g. simulated binary crossover (SBX) [11], polynomial mutation (PM) [12] and differential evolution [13, 14].

2.2 Multi-objective optimization

If there is only one objective to optimize for, each solution has only one fitness property. Comparing different solutions means comparing two fitness values, one from each solution. Given an objective to be minimized, at any time in the optimization process, the best solution found so far, is always the one with the lowest fitness value. However many real world problems have (partially) conflicting objectives, and thus no single best, but only good compromising, Pareto optimal solutions can be found. While any of these compromise solutions are formally incomparable, the user of the algorithm can take other, even subjective criteria into consideration and choose one or more solutions from the Pareto set based on the additional information. As motivated above, there can only exist a partial order to compare solutions for multi-objective optimization because some individuals are incomparable. A commonly employed partial order is the dominance relation, which compares all corresponding fitness values of two individuals pairwise for each dimension in objective function space. An individual $\vec{x}_1$ dominates another individual $\vec{x}_2$ iff for all objective functions the value of the first individual is better ($<$) than the one from the second individual [5].

Strict dominance relation. If all objectives $\vec{f}(\vec{x}) = (f_1, ..., f_M)$ are to be minimized:

$$\vec{x}_1 \prec \vec{x}_2 \Leftrightarrow f_i(\vec{x}_1) < f_i(\vec{x}_2) \,\forall \text{ objectives } f_i. \text{ [5]}$$
(read: individual $\vec{x}_1$ dominates individual $\vec{x}_2$)

Since individuals that share at least one equal corresponding objective function value can never be compared regarding the dominance order, the weak dominance relation can be used instead. Weak dominance has the same meaning as the dominance order, with the $\leq$ relation instead of the $<$ relation. If individual $\vec{x}_1$ does not dominate individual $\vec{x}_2$ the $\not\prec$ sign is used to describe the relation between the individuals ($\vec{x}_1 \not\prec \vec{x}_2$).

Weak dominance relation. If all objectives $\vec{f}(\vec{x}) = (f_1, ..., f_M)$ are to be minimized:

$$\vec{x}_1 \preceq \vec{x}_2 \Leftrightarrow f_i(\vec{x}_1) \leqslant f_i(\vec{x}_2) \,\forall \text{ objectives } f_i. \text{ [5]}$$
(read: individual $\vec{x}_1$ weakly dominates individual $\vec{x}_2$)

An individual $\vec{x}_1$ from population P which is not dominated by any other individual $\vec{x}_2 \in P$ is classified as non-dominated in P. A set of individuals which are all non-dominated in P is called a non-dominated set. If such a set contains all non-dominated individuals in P it forms the first front $\mathcal{F}$ of that population.

Non-dominance, non-dominated set, first Front. A non-dominated individual $\vec{x}_{nd}$ is an individual from population P:

$$\forall \vec{x}_i \in P : \vec{x}_i \not\prec \vec{x}_{nd}.$$

$\mathcal{F}_{nd} \subseteq P$ is called a non-dominated set, if:

$$\forall \vec{x}_i \in \mathcal{F}_{nd} : \forall \vec{x}_j \in P : \vec{x}_j \not\prec \vec{x}_i.$$

Fronts, non-dominated sorting. A partition $\mathcal{F}$ of a set S into subsets $\mathcal{F}_i$ (fronts)

$$\mathcal{F} = \{\mathcal{F}_1, \mathcal{F}_2, ...\}$$

is called non-dominated sorting, with the first front $\mathcal{F}_1 \subseteq S$:

$$\mathcal{F}_1 = \{\vec{x}_i \in S \mid \forall \vec{x}_j \in S : \vec{x}_j \not\prec \vec{x}_i \wedge \forall \vec{x}_k \in S \cap \mathcal{F}_1 : \exists \vec{x}_i : \vec{x}_i \prec \vec{x}_k\}$$

and all fronts $\mathcal{F}_i \, \forall i > 1$:

$$\mathcal{F}_i = \mathcal{F}_1 \subseteq S' \text{ where } S' = S \setminus (\mathcal{F}_1 \cup ... \cup \mathcal{F}_{i-1})$$

The first front contains all non-dominated individuals and is also called Pareto front [15], all individuals in that front are Pareto optimal [15]. Individuals $\vec{x}_i \in \mathcal{F}_i$ are considered "better" than $\vec{x}_j \in \mathcal{F}_{i+1}$, because there is at least one $\vec{x}_i$ which dominates any $\vec{x}_j$, so removing an $\vec{x}_j$ from the population still leaves the dominating individual in the population and so the fitness of the population is still the same.

Many multi-objective evolutionary algorithms, like NSGA-II, SMS-EMOA and GDE3, use the (weak) dominance relation as the first criterion to compare solutions to one another and evaluate those individuals, which cannot be compared this way by employing a secondary criterion. With multi-objective evolutionary algorithms, the second criterion is usually a measure for the spreading of the solutions along the Pareto front to preserve the diversity of the population (and solutions). The dominance relation enforces convergence against the optimum, while the secondary criterion spreads the solutions along the Pareto front in that optimum. It should be noted, that this only preserves diversity along the currently found Pareto front. For applications which require a large amount of diversity in decision space, additional means may be necessary, as all the solutions are likely located in the same (local) optimum. On the other hand, this behavior can be a desirable feature. For instance if only very few function evaluations can be performed, hence fast convergence against an optimum is required.

While there are many stopping criteria possible for evolutionary algorithms, only the total number of function evaluations are considered in this work, because a function evaluation is the most expensive part in surface reconstruction and, thus, all other operations are negligible as far as computational complexity is concerned. Also with the limitation to 20000 function evaluations, only smaller population sizes are feasible - otherwise each individual in the population can only be improved too few times and no individual reaches a decent performance.

2.2.1 NSGA-II

The elitist Non-dominated Sorting Genetic Algorithm (NSGA-II) [6] is a very popular multi-objective evolutionary algorithm proposed by Deb, Agrawal, Pratap and Meyarivan in 2000. It has since been successfully applied to various problems [16, 17, 18] and has also served as a basis for other algorithms like GDE3 and SMS-EMOA. It is therefore employed in this work for comparison purposes as well.

In each generation parents are chosen to breed $|P|$ new children ($\lambda = \mu$). The children are created as recombination of their parents, mutated afterwards and finally appended to the current population, regardless of their fitness. After all children have been created, the population size has increased by factor 2. The population size is then reduced to the original size again to reduce the complexity of the algorithm. If no such measure is taken, the size of the population doubles in each generation and thus the computational complexity of the entire algorithm grows exponentially with the number of generations. First non-dominated sorting partitions the population into fronts. The lower the front number, the better the solution, hence the fronts are included into the next generation beginning with the first front, until a front is reached that does not completely fit into the next population. This front can obviously only be included partially, so for any individual in this front, the crowding-distance [6] is calculated and those individuals with the highest crowding distance are added to the population until it reaches its normal size again. All other individuals are deleted.

The crowding-distance value of a solution $CD(\vec{f}(\vec{x}_i))$ is half the perimeter of the cuboid including $\vec{f}(\vec{x}_i)$ limited by the neighbor solutions from the same front. To calculate the crowding-distance efficiently, all individuals in the front are sorted by every objective. Regarding each objective, the minimal and maximal objective value attained by any member of the front is determined. The difference between these two values is used to normalize the distances between each individual and its neighbors to consider all objectives equally. The crowding-distance value for an individual $\vec{x}_i$ is then computed as the sum of the normalized distances between left ($f_m(\vec{x}_{ln})$) and right ($f_m(\vec{x}_{rn})$) neighbor regarding every objective m.
Though NSGA-II is not limited to specific variation operators, the most commonly ones used together with NSGA-II are simulated binary crossover (SBX) [11] and polynomial mutation (PM) [12]. The only parameter the NSGA-II itself uses is the population size μ. Depending on the variation operator(s), additional parameters may be available, e.g. two for SBX and PM respectively. The runtime of one generation of this algorithm is governed by the complexity of non-dominated sorting, hence the NSGA-II can be implemented in $\mathcal{O}(\mu \log^{M-1} \mu)$ [19].

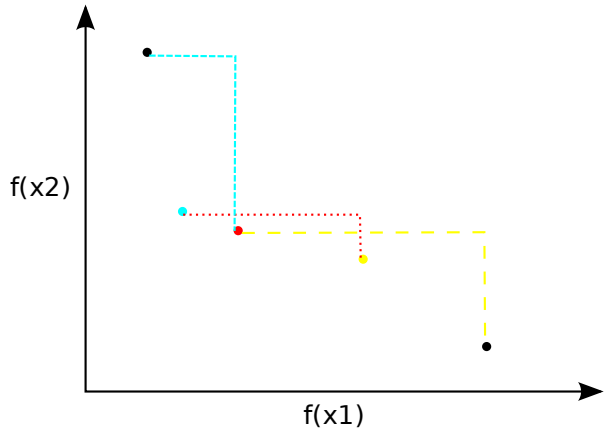


Figure 2.2: Graphical perception of the crowding-distance for two objectives

Algorithm 1 Calculating the crowding-distance for front $\mathcal{F}_i$

1: $n \leftarrow |\mathcal{F}_i|$ // number of individuals in front
2: **for all** $\vec{x}_j \in \mathcal{F}_i$ **do** // all individuals in front
3: $dist[j] \leftarrow 0$ // initialize distance
4: **end for**
5: **for** $m = 0$ to $M - 1$ **do** // all objectives
6: $F_t \leftarrow \text{sort}(f_m(\vec{x}), \leq, \mathcal{F}_i)$ // sort ascending by objective m
7: $dist[0] \leftarrow \infty$, $dist[n-1] \leftarrow \infty$ // boundary points
8: **for** $j = 1$ to $n - 2$ **do** // all other points
9: $\vec{x}_{ln} \leftarrow F_t[j-1]$, $\vec{x}_{rn} \leftarrow F_t[j+1]$ // left and right neighbor
10: $\vec{x}_{max} \leftarrow F_t[n-1]$, $\vec{x}_{min} \leftarrow F_t[0]$ // min and max value regarding f_m
11: $dist[j] \leftarrow dist[j] + (f_m(\vec{x}_{rn}) - f_m(\vec{x}_{ln}))/(f_m(\vec{x}_{max}) - f_m(\vec{x}_{min}))$ // see figure 2.2
12: **end for**
13: **end for**

Algorithm 2 NSGA-II

$g \leftarrow 0$ // initialize generation counter
$P_g \leftarrow$ initializePopulation()
$f \leftarrow \mu$ // number of function evaluations performed
while $f + \mu < f_{max}$ **do** // generation loop
  $P_{temp} \leftarrow P_g$
  for $i = 0$ to $\mu - 1$ **do**
    // breed
    $parents_i \leftarrow$ selectParents(P_g)
    $child_i \leftarrow$ recombine($parents_i$)
    $child_i \leftarrow$ mutate($child_i$)
    $P_{temp} \leftarrow P_{temp} \cup child_i$ // add child to intermediate population
    // evaluate
    evaluate($child_i$) // determine fitness
  end for
  // select
  $\mathcal{F} \leftarrow$ sort(fitness, $\prec$, P_{temp}) // non-dominated-sorting, Fronts $\mathcal{F} = \mathcal{F}_1, \mathcal{F}_2, ...$
  $P_{g+1} \leftarrow \emptyset$ // next generation
  $i \leftarrow 0$
  while $|P_{g+1} \cup \mathcal{F}_i| < \mu$ **do** // append all fronts that fit in completely
    $P_{g+1} \leftarrow P_{g+1} \cup \mathcal{F}_i$
    $i \leftarrow i + 1$
  end while
  crowdingDistance($\mathcal{F}_i$)
  $\mathcal{F}_i \leftarrow$ sort(crowdingDistance, $\geq$, $\mathcal{F}_i$) // sort descending by crowding distance
  $n \leftarrow \mu - |P_{g+1}|$
  $P_{g+1} \leftarrow P_{g+1} \cup \mathcal{F}_i[0 : n[$ // add individuals with largest crowding distance
end while

2.2.2 SMS-EMOA

SMS-EMOA [3] can be considered a modification of the NSGA-II and therefore offers the same parameters for tuning. NSGA-II uses the same pool of parents for as many function evaluations as there are individuals in the population ($\mu + \mu$ approach). Children created in the same generation, with potentially better fitness cannot serve as parents until the generation is complete, later in the same generation bred children can therefore not profit from already created children's (better) fitness, because these are denied parenthood. Also the crowding-distance is not a very accurate measure for the spreading of the population along the Pareto front [20].

To cover for these deficiencies of NSGA-II, two additional measures were introduced with SMS-EMOA:

1. a steady-state $(\mu + 1)$ approach in the breeding phase and
2. the hypervolume contribution as secondary selection criterion.

The hypervolume contribution of a point p from a set of points S regarding a reference point r is defined as the S-metric [21] value of S regarding r minus the S-metric value of $S \setminus \{p\}$ regarding r. As a quality measure the S-metric features several favorable properties [22]:

- The S-metric value of a set S_1 with $S_1 \prec S_2$ is always higher than the S-metric value of S_2.
- Additional non-dominated solutions do not decrease the S-metric value (monotony).
- Invariance against linear scaling of the objective space [23].

Since the secondary selection criterion needs to be computed once per generation and SMS-EMOA has to perform more generations because of the $\mu + 1$ approach, it is slower than NSGA-II. Also calculation of the S-metric is significantly more complex than computation of the crowding-distance for more than three objectives. However, since only the individual with the worst hypervolume contribution from the last front is removed during the selection phase, the S-metric value of the population can only increase, hence the quality of the solutions can only get better according to the S-metric (monotony). However the premise for the design of SMS-EMOA was to only run few function evaluations, and the function evaluations are presumed to be the most expensive operation in the overall computation. In these cases the slowdown by the measures is negligible. For example, if 1000 function evaluations require a total computation time of 2 days, the user will not care about an additional 18 minutes, required by the S-metric calculations.

Due to the additional measures SMS-EMOA has outperformed NSGA-II on a lot of practical applications. Calculating the S-metric for the hypervolume contribution is the most expensive operation of SMS-EMOA, if function evaluations are negligible. Efficient algorithms for calculating the hypervolume contribution are known for the special case of 2 and 3 objectives, for more objectives the complexity dramatically increases [24, 25]. Given an algorithm calculating the S-metric value, for a set S of n points, the hypervolume contribution can be determined by running the algorithm n times: each time without one different point. The hypervolume contribution of a point p equals the difference between the hypervolume covered by S and the hypervolume covered by $S \setminus \{p\}$. For the two-dimensional case an algorithm is known, which can compute all hypervolume contributions in $\mathcal{O}(n \log n)$ [23].

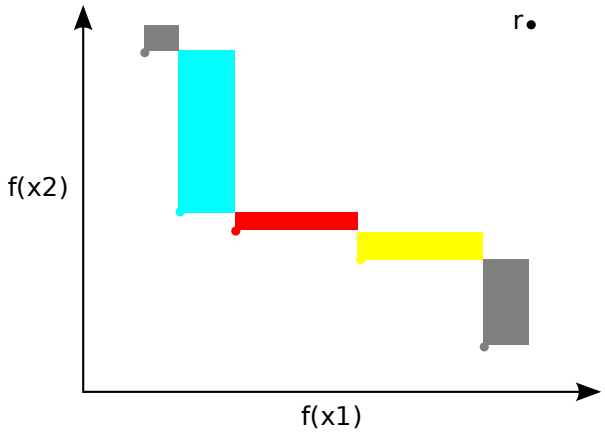


Figure 2.3: Graphical perception of the hypervolume contribution for two objectives

M	Hypervolume	Hypervolume contribution
2 [23]	$\mathcal{O}(n \log n)$	$\mathcal{O}(n \log n)$
3 [25]	$\mathcal{O}(n \log n)$	$\mathcal{O}(n^2 \log n)$
4+ [25]	$\mathcal{O}(n^{D-2} \log n)$	$\mathcal{O}(n^{D-1} \log n)$
4+ [24]	$\mathcal{O}(n^{\frac{D}{2}} \log n)$	$\mathcal{O}(n^{\frac{D}{2}+1} \log n)$

Table 2.1: Computational complexity: hypervolume calculation for n points in $\mathbb{R}^D$

2.2.3 GDE3

Differential evolution has been very successful in single-objective optimization [13, 26, 27, 14]. Unlike SBX and PM, the variation operator from differential evolution is invariant against the rotation of the search coordinate system and very easy to understand, so crossing that concept with the already available multi-objective algorithms seems promising and has been subject to several publications in the past [5, 28, 29, 30]. In 2005 Kukkonen and Lampinen published the third version of their approach called Generalized Differential Evolution [5]. This algorithm is much like NSGA-II, but uses differential evolution from algorithm 4 as variation operator and has a slightly modified selection. Also constraints are supported, but since all the applications used in this work do not require constraints, the GDE3 is presented here without supporting constraints. The breeding phase is the same as with DynDE, for each child the following is determined:

- a random dimension in decision space
- four mutually different parents, one parent is the predecessor, the other three p_1, p_2, p_3 are chosen at random.

Algorithm 3 SMS-EMOA

1: $g \leftarrow 0$ // initialize generation counter
2: $P_g \leftarrow$ initializePopulation()
3: $f \leftarrow \mu$ // number of function evaluations performed
4: **while** $f + 1 < f_{max}$ **do** // generation loop
5: $P_{temp} \leftarrow P_g$
6: // breed (steady-state)
7: $parents \leftarrow$ selectParents(P_g)
8: $child \leftarrow$ recombine($parents$)
9: $child \leftarrow$ mutate($child$)
10: $P_{temp} \leftarrow P_{temp} \cup child$ // add child to intermediate population
11: // evaluate
12: evaluate($child$) // determine fitness
13: // select
14: $\mathcal{F} \leftarrow$ sort(fitness, $\prec$, P_{temp}) // non-dominated-sorting, Fronts $\mathcal{F} = \mathcal{F}_1, \mathcal{F}_2, ...$
15: $P_{g+1} \leftarrow \emptyset$ // next generation
16: $i \leftarrow 0$
17: **while** $|P_{g+1} \cup \mathcal{F}_i| < \mu$ **do** // append all fronts that fit in completely
18: $P_{g+1} \leftarrow P_{g+1} \cup \mathcal{F}_i$
19: $i \leftarrow i + 1$
20: **end while**
21: hypervolumeContribution($\mathcal{F}_i$)
22: $\mathcal{F}_i \leftarrow$ sort(hypervolumeContribution, $\geq$, $\mathcal{F}_i$) // sort descending by HV contribution
23: $n \leftarrow \mu - |P_{g+1}|$
24: $P_{g+1} \leftarrow P_{g+1} \cup \mathcal{F}_i[0 : n[$ // add individuals with largest HV contribution
25: **end while**

Predecessor. Given a $(\mu + \lambda)$ approach, the predecessor $pred$ from generation g of the ith child created in generation $g + 1$ is:

$$pred = \begin{cases} \vec{x}_{i,g}, & \lambda = \mu \\ \vec{x}_{rand(0,\mu-1),g}, & \text{otherwise} \end{cases} \tag{2.1}$$

For each dimension, the decision variables of the child are then allocated with the corresponding values from the predecessor, except for the dimension that was chosen at random and those dimensions where a randomly drawn number $rand \in [0, 1]$ does not exceed the parameter CR. For those dimensions the value is computed as crossover from the other parents according to the following formula: $child[d] = p_3[d] + F \cdot (p_1[d] - p_2[d])$. This kind of variation can be tuned in two different ways, by two parameters:

1. CR, the crossover rate, determines the variation probability of a variable in decision space, and

2. F, the step size factor, weights the distance of the modification performed.

If the child weakly dominates its predecessor, the child will replace the predecessor in the next generation. On the other hand, if the predecessor dominates the child, it is deleted. Otherwise, it is appended to the population. This way the population size can still increase, but in most cases it does not increase by factor 2, because children can be directly rejected and some predecessors immediately get replaced. To prune the population, the selection known from NSGA-II is performed.

Algorithm 4 differentialEvolution($parents$, $pred$, d_{rand}) (Storn, Price [13])

$p_1, p_2, p_3 \leftarrow parents$
$child \leftarrow pred$ // copy from predecessor (equation (2.1))
$d \leftarrow d_{rand}$ // start with dimension d_{rand}
repeat
  $child[d] \leftarrow p_3[d] + F \cdot (p_1[d] - p_2[d])$ // modify
  $d \leftarrow (d + 1) \mod D$ // do not stop at last dimension
until rand(0,1) $\geqslant CR$
return *child*

Algorithm 5 differentialEvolution($parents$, $pred$, d_{rand}) (Mendes, Mohais [14])

$p_1, p_2, p_3 \leftarrow parents$
for $d = 0$ to $D - 1$ **do** // all dimensions
  if rand(0,1) $< CR \vee d = d_{rand}$ **then** // modify
    $child[d] \leftarrow p_3[d] + F \cdot (p_1[d] - p_2[d])$
  else // copy from predecessor
    $child[d] \leftarrow pred[d]$
  end if
end for
return *child*

Algorithm 6 GDE3 (without constraint support)

1: $g \leftarrow 0$ // initialize generation counter
2: $P_g \leftarrow$ initializePopulation()
3: $f \leftarrow \mu$ // number of function evaluations performed
4: **while** $f + \mu < f_{max}$ **do** // generation loop
5: $P_{temp} \leftarrow P_g$
6: **for** $i = 0$ to $\mu - 1$ **do**
7: // breed
8: $parents_i \leftarrow$ selectParents(P_g)
9: $pred_i \leftarrow P_g[i]$ // predecessor
10: $d_{rand} \leftarrow$ rand(0,$D - 1$) // random number from $[0, D - 1]$
11: $child_i \leftarrow$ differentialEvolution($parents_i$, $pred_i$, d_{rand})
12: // evaluate
13: evaluate($child_i$) // determine fitness
14: // select part 1
15: **if** $child_i \preceq pred_i$ **then** // child weakly dominates its predecessor
16: $P_{temp}[i] \leftarrow child_i$ // replace predecessor with child
17: **else if** $pred_i \not\prec child_i$ **then** // predecessor does not dominate child
18: $P_{temp} \leftarrow P_{temp} \cup child_i$ // add child anyway
19: **end if**
20: **end for**
21: // select part 2, same as NSGA-II
22: $\mathcal{F} \leftarrow$ sort(fitness, $\prec$, P_{temp}) // non-dominated-sorting, Fronts $\mathcal{F} = \mathcal{F}_1, \mathcal{F}_2, ...$
23: $P_{g+1} \leftarrow \emptyset$ // next generation
24: $i \leftarrow 0$
25: **while** $|P_{g+1} \cup \mathcal{F}_i| < \mu$ **do** // append all fronts that fit in completely
26: $P_{g+1} \leftarrow P_{g+1} \cup \mathcal{F}_i$
27: $i \leftarrow i + 1$
28: **end while**
29: crowdingDistance($\mathcal{F}_i$) // calculate crowding distance
30: $\mathcal{F}_i \leftarrow$ sort(crowdingDistance, $\geq$, $\mathcal{F}_i$) // sort descending by crowding distance
31: $n \leftarrow \mu - |P_{g+1}|$
32: $P_{g+1} \leftarrow P_{g+1} \cup \mathcal{F}_i[0 : n[$ // add individuals with largest crowding distance
33: **end while**

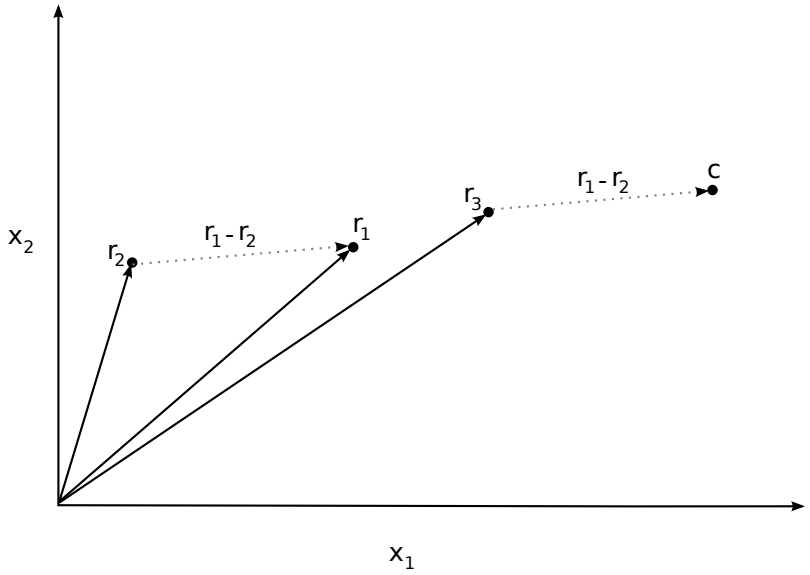


Figure 2.4: Differential evolution: location of the child in decision space ($F = 1$)

2.3 Test functions

Theoretical results in the area of evolutionary algorithms are still limited to very simple problems and algorithms [31, 32, 33]. In the area of multi-objective optimization, the situation is even more complex and, thus, experiments are the common way to analyze behavior and performance. For real-world applications, however, function evaluations are usually very expensive. Thus, running many experiments is often not feasible. Instead the behavior of an algorithm can be analyzed on a set of test problems, which are much faster to evaluate. If the fitness landscape of the test problem has similar properties as the real application, the knowledge obtained from the application of the algorithm to the test function can be transfered to the real application.

Evolutionary algorithms are general search heuristics expected to work on many different problems. Typical behavior of an algorithm can therefore be observed on different problems. Applying the algorithm on the real application can show whether the observations from the run on the test problems do also hold. The test problems used in this work are taken from the test suite developed for the CEC'07 MOEA contest [34].

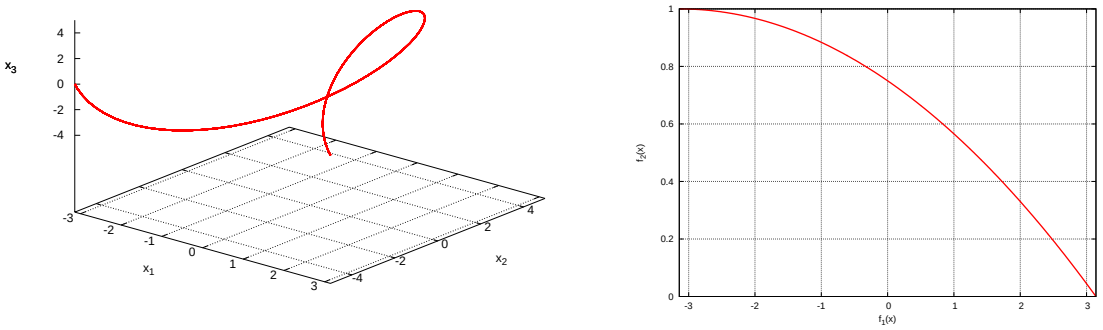
Test function	**M**	**D**	**Modality**
OKA2	2	3	uni
SYM-PART	2	30	multi
R_ZDT4	2	10	multi
S_DTLZ3	3	30	multi

Table 2.2: Properties of the test functions

OKA2 [35] (equation (2.2)) is a test function which is very easy to describe and implement, but has proven to be very difficult for evolutionary algorithms. As the decision space is limited to three dimensions, results can be plotted and the behavior of an algorithm on that function can be easily analyzed in decision space. For other test problems with a high dimensional decision space this is not the case. The Pareto set of OKA2 ($\vec{x} = (a, 5\cos a, 5\sin a)^T$, $a \in [-\pi, \pi]$) looks like a spiral.

$$\begin{aligned} & f_1(\vec{x}) = x_1 \\ & f_2(\vec{x}) = 1 - \frac{1}{4\pi^2}(x_1 + \pi)^2 + |x_2 - 5\cos x_1|^{\frac{1}{3}} + |x_3 - 5\sin x_1|^{\frac{1}{3}} \\ & x_1 \in [-\pi, \pi], x_2, x_3 \in [-5, 5] \end{aligned} \tag{2.2}$$

In 2007 Rudolph et al. proposed a test problem to directly measure the capability of multi-objective algorithms to preserve diversity in decision space called SYM-PART [36], most other test functions are designed to only help measure performance in objective space. Many evolutionary algorithms only take the diversity in the objective function space into account. The basic idea behind this test function is to take a simple, box constrained



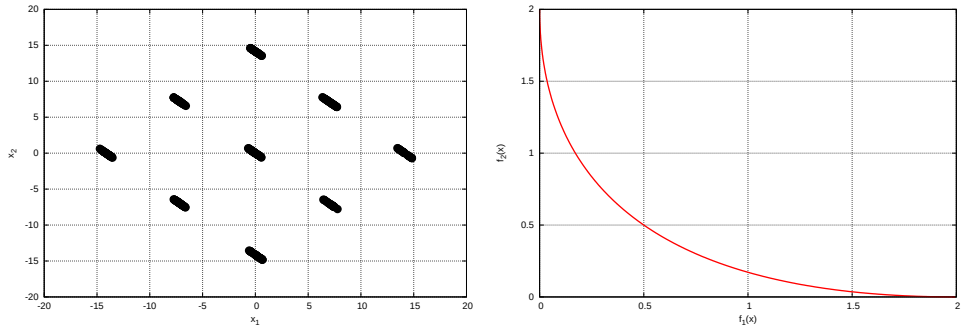
(a) Pareto set (b) Pareto front

Figure 2.5: OKA2: Pareto set and front

test problem, rotate it in order to make optimization of the problem more difficult and put multiple of these boxes together. As a result the decision space increases in size and multiple global optima are introduced, one in each box. Rudolph et al. call such a box tile. Because all global optima offer equal fitness, only one optimum needs to be found in order to cover the entire Pareto front. The diversity of high-quality solutions can be determined by calculating how many of these optima are found and maintained by an algorithm. To that end the covered sets indicator was developed (see section 2.4.4). Originally SYM-PART (equation (2.3)) was introduced with only two dimensions in decision space and two objective functions, but was later [34] extended to 30 and theoretically an unlimited, even number of dimensions in decision space. Many state-of-the-art algorithms manage to only cover one optimum [28] of SYM-PART(30D).

$$
\begin{aligned}
&f_1(\vec{x}) = (z_1 + a - t_1 c_2)^2 + (z_2 - t_2 b)^2 + ... + (z_{D-1} + a - t_1 c_2)^2 + (z_D - t_2 b)^2 \\
&f_2(\vec{x}) = (z_1 - a - t_1 c_2)^2 + (z_2 - t_2 b)^2 + ... + (z_{D-1} - a - t_1 c_2)^2 + (z_D - t_2 b)^2 \\
&\text{where:} \\
&t_1 = \text{sgn}(z_1) \cdot \left\lceil \frac{z_1 - \frac{1}{2}c_2}{c_2} \right\rceil, t_2 = \text{sgn}(z_2) \cdot \left\lceil \frac{z_2 - \frac{1}{2}b}{b} \right\rceil \\
&a = 1, b = 10, c = 8, c_2 = c + 2a = 10 \\
&\\
&\text{rotation: } \vec{z} = (z_1, \cdots, z_D)^T = M\vec{x}, \vec{x} \in [-20, 20]^D
\end{aligned}
\tag{2.3}
$$

R_ZDT4 is an extended and rotated version of the multi-modal test problem ZDT4. Because for the ZDT functions the global optimum is located at the bottom or in the center of the search space, the difficulty of these test functions can be significantly increased by extending and rotating the search space [34]. This extended and rotated version of ZDT4



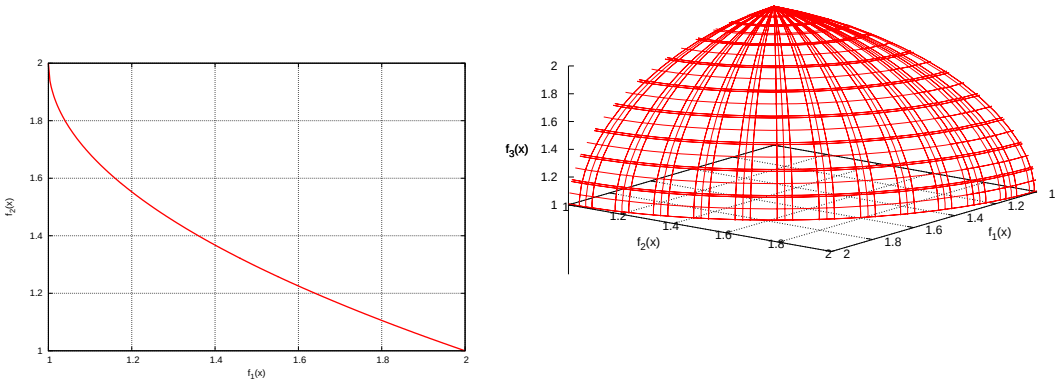
(a) Pareto sets (b) Pareto front

Figure 2.6: SYM-PART: Pareto sets and front

is called R_ZDT4 (equation (2.4)). The major difficulty of ZDT4 is, that it contains a total of 21^9 local Pareto optima, in which an algorithm can get stuck prior to finding the global optimum. Insufficient diversity in decision space will most likely cause an algorithm to stop in a local optimum, whereas, an algorithm performing well on this test function can handle multi-modality.

$$
\begin{aligned}
f_1(\vec{x}) &= \begin{cases} z_1' + 1, & z_1 \geq 0 \\ S(p_1)(z_1' + 1), & z_1 < 0 \end{cases} \\
f_2(\vec{x}) &= \begin{cases} g(x)(1 - \sqrt{z_1'/g(x)}) + 1, & \text{all } z_i \geq -5 \\ S\left(\sqrt{\sum_{i=1}^{D} p_i}\right)\left(g(x)(1 - \sqrt{z_1'/g(x)}) + 1\right), & \text{otherwise} \end{cases} \\
&\text{where:} \\
&g(x) = 1 + 10(D-1) + \textstyle\sum_{i=2}^{D}(z_i'^2 - 10\cos(4\pi z_i')) \\
z_1' &= \begin{cases} -\lambda_1 z_1, & z_1 < 0 \\ z_1, & 0 \leqslant z_1 \leqslant 1 \\ 1 - \lambda_1(z_1 - 1), & z_1 > 1 \end{cases} \quad p_1 = \begin{cases} -z_1, & z_1 < 0 \\ 0, & 0 \leqslant z_1 \leqslant 1 \\ z_1 - 1, & z_1 > 1 \end{cases} \\
z_i' &= \begin{cases} -5 - \lambda_i(z_i + 5), & z_i < -5 \\ z_i, & -5 \leqslant z_i \leqslant 5 \\ 5 - \lambda_i(z_i - 5), & z_i > 5 \end{cases} \quad p_i = \begin{cases} -5 - z_i, & z_i < -5 \\ 0, & -5 \leqslant z_i \leqslant 5 \\ z_i - 5, & z_i > 5 \end{cases}
\end{aligned}
\tag{2.4}
$$

rotation: $\vec{z} = (z_1, \cdots, z_D)^T = M\vec{x}$,
scale factor: $\vec{\lambda}$,
stretch function: $S(t) = \dfrac{2}{1 + e^{-t}}$



(a) R_ZDT4 (b) S_DTLZ3

Figure 2.7: Pareto fronts

$$f_1(\vec{x}) = \begin{cases} (1+g(x_M))\cos(z'_1\pi/2)...\cos(z'_{M-2}\pi/2)\cos(z'_{M-1}\pi/2), & z_i \geq 0 \\ S(psum_1)\left((1+g(x_M))\cos(z'_1\pi/2)...\cos(z'_{M-2}\pi/2)\cos(z'_{M-1}\pi/2)+1\right), & \text{otherwise} \end{cases}$$

$$f_2(\vec{x}) = \begin{cases} (1+g(x_M))\cos(z'_1\pi/2)...\cos(z'_{M-2}\pi/2)\sin(z'_{M-1}\pi/2), & z_i \geq 0 \\ S(psum_2)\left((1+g(x_M))\cos(z'_1\pi/2)...\cos(z'_{M-2}\pi/2)\sin(z'_{M-1}\pi/2)+1\right), & \text{otherwise} \end{cases}$$

$$f_3(\vec{x}) = \begin{cases} (1+g(x_M))\cos(z'_1\pi/2)...\cos(z'_{M-3}\pi/2)\sin(z'_{M-2}\pi/2), & z_i \geq 0 \\ S(psum_3)\left((1+g(x_M))\cos(z'_1\pi/2)...\cos(z'_{M-3}\pi/2)\sin(z'_{M-2}\pi/2)+1\right), & \text{otherwise} \end{cases}$$

where:

$$g(x_M) = 100\left(|x_M| + \sum_{x_i \in x_M}\left[(z'_i - \tfrac{1}{2})^2 - \cos(20\pi(z'_i - \tfrac{1}{2}))\right]\right)$$

$$z'_i = \begin{cases} z_i, & z_i \geq 0 \\ -\lambda_i z_i, & z_i < 0 \end{cases} \qquad p_i = \begin{cases} 0, & z_i \geq 0 \\ |z_i|/d_i, & z_i < 0 \end{cases}$$

shift: $\vec{z} = (z_1, \cdots, z_D)^T = \vec{x} - \vec{o}$,
scale factor: $\vec{\lambda}$,
stretch function: $S(t) = \dfrac{2}{1+e^{-t}}$ (2.5)

S_DTLZ3 (equation (2.5)) is an extended and shifted version of DTLZ3 [37]. It is multimodal and supports $M \geq 3$ objectives.

2.4 Performance metrics

Since experiments need to be evaluated, formal criteria are necessary to allow a fair performance assessment and comparison of solutions calculated by different algorithms. To that end, several unary indicators have been proposed. These metrics do not assess one individual, but the entire set of solutions compared to an approximation set. Implementations for all indicators described are available from the PISA [38] performance assessment toolkit.

2.4.1 Additive ε-indicator

Given an approximation set A and a set of individuals P, the ε-indicator [39] equals the minimum value I_ε the fitness regarding every objective has to be improved so that for every individual in P, $P \preceq A$ holds (equation (2.6)).

$$I_\varepsilon(A, P) = \inf_{\varepsilon \in \mathbb{R}} \{\forall \vec{x}_i \in A : \exists \vec{x}_j \in P : \vec{x}_j \preceq \vec{x}_i\} \tag{2.6}$$

Figuratively speaking, the ε indicator depicts how far P has to be moved towards A in objective space until no solution in A is better than all solutions in P [38].

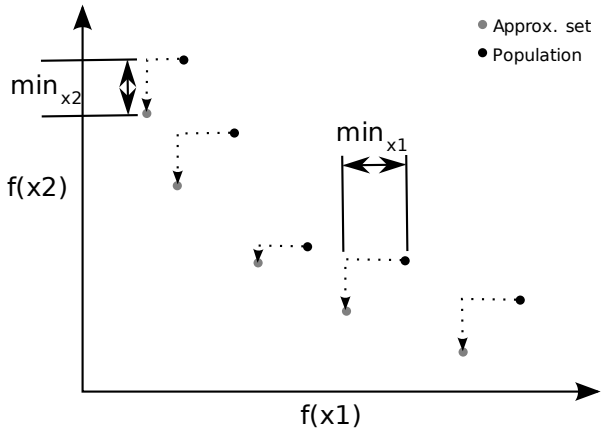


Figure 2.8: ε-indicator calculation (two objectives)

2.4.2 Hypervolume-Indicator

The hypervolume-indicator compares the S-metric [21] value of the solutions and a suitable reference set in objective space. The hypervolume of the reference set is subtracted from the hypervolume of the solutions. The closer the Pareto front found by an algorithm is located to the reference set, the smaller the difference between both S-metric values gets. Thus, the smaller the hypervolume indicator value I_H, the better the solution.

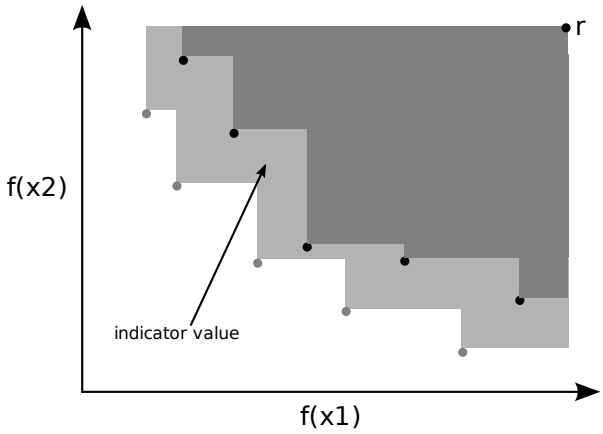


Figure 2.9: Hypervolume-indicator calculation (two objectives)

2.4.3 R2-Indicator

The main idea behind the R2-indicator [40] is to transform a multi-objective fitness vector into a single-objective value via a utility function $u(\lambda)$. First, for all individuals in A and P, the utility function values are calculated. The indicator value I_{R2} is computed as the weighted sum over the differences between the max value u^* of u from all individuals in A on the one side and all individual in P on the other side [38].

$$I_{R2}(A, P) = \frac{\sum_{\lambda \in \Lambda} u^*(\lambda, A) - u^*(\lambda, P)}{|\Lambda|} \tag{2.7}$$

2.4.4 Covered sets indicator

Calculating the number of covered sets is only supported on the SYM-PART test problem. As this function possesses several global optima with different location in decision space, the number of global optima covered by the final population can be used as a measure for the diversity of the solutions in decisions space [36]. An optimum is covered, if at least one individual is in proximity to the Pareto front in decision space. The more optima covered, the more diverse the final population is. The covered set indicator [36] also evaluates the performance of the solutions, because in order to cover an optimum, a solution needs to be almost optimal and also stay in the population until optimization is complete. GDE3 finds several of these optima, but drops individuals covering those sets in favor of slightly better individuals in one set [28]. When the algorithm is done, only one set remains covered. One major task of this work is to present measures to counteract this behavior.

2.5 Surface reconstruction

The transformation process of a surface represented by a set of scan points into a CAD model of the surface is called surface reconstruction [2]. While normally a CAD model of a work piece is designed before the work piece (prototype) itself, there are occasions where an accurate CAD model for an object is not available, e.g. due to modification during machining or because the work piece was designed and manufactured by a third party. A CAD model can therefore either be generated manually or automatically. Manual design of a CAD model for a given object is extremely difficult and time consuming if a high level of accuracy is required. Therefore, at the institute of machining technology the following work flow is employed to automatize the reconstruction:

1. A 3D scanner digitizes the work piece and obtains a highly accurate description of the surface through a large number of scan points. As this kind of description is very difficult to handle for computer systems, because both computational requirements and memory consumption of such a structure are enormous, a smaller and simpler mathematical description is preferable.

2. To reduce the workload for the surface reconstruction algorithm, the number of scan points is reduced. Several algorithms for this step can be found in the PhD thesis "Evolutionäre Flächenrekonstruktion" by Mehnen [2]. If only a part of the scanned object needs to be reconstructed, the amount of scan points can be reduced even further manually by cutting off all other parts.

3. With a deterministic algorithm a pre-optimized surface representation based upon NURBS surfaces [1] is computed afterwards as an intermediate solution. This algorithm only takes into account one objective, the proximity between the reconstructed surface and the filtered set of scan points.

4. Based upon the pre-optimized surface a multi-objective evolutionary algorithm obtains solutions satisfying multiple objectives, each representing desired features of reconstructed surfaces. Since multiple objectives are involved in the optimization process, a set of Pareto optimal solutions is determined for the user to choose from.

Surface reconstruction also is an interesting problem to evaluate and analyze multi-objective evolutionary algorithms. As in many real-world problems, the dimensionality of the decision space is very high. Usually it is not possible to visualize the decision space with $\gg 3$ dimensions. Since for surface reconstruction each dimension of the decision space represents a coordinate of a control point in a NURBS surface, the resulting surface and its properties can be easily visualized, allowing for an easier analysis of the behavior of the evolutionary algorithm.

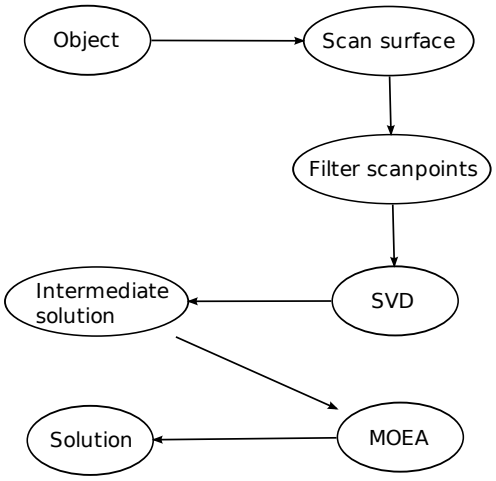


Figure 2.10: Hybrid approach for surface reconstruction

2.5.1 Data structures

The following sections of this chapter are based upon the papers "On the Design of Optimizers for Surface reconstruction" by Wagner, Michelitsch and Sacharow [41] and "On the Use of Problem-Specific Candidate Generators for the Hybrid Optimization of Multi-Objective Production-Engineering Problems" by Weinert, Zabel Kersting, Michelitsch and Wagner. A NURBS (Non-Uniform Rational B-Spline) surface is a mathematical description of a surface. The data structure involved consists of three parts,

- the order (p,q) (usually $p = q = 3$),
- a control net with $n \times m$ vertices and the corresponding $n \times m$ weight matrix and
- two knot vectors u and v.

Important features of NURBS surfaces are various smoothness properties, local support, compact data structures and numerical stability. [1] offers a detailed description on the construction and the features of NURBS surfaces and the underlying mathematical formulas. The shape of a NURBS surface is defined by the net of control points and the related weight matrix. Modifying the position of the control points can therefore change the appearance of the surface. Due to the local support property, moving a control point to another position only affects a locally limited area of the surface.

2.5.2 Algorithms

Before the actual optimizers can be understood, the objectives involved in the optimization need to be defined. These objectives mirror desirable features of the surfaces. Ideally the scan points should lie directly on the surface. Therefore the distance between the

reconstructed surface and the scan points and/or vice versa is to be minimized. As most real objects to be reconstructed have a smooth surfaces, an additional objective is the regularity of the control net. Other desired features, e.g. small distance between surface and scan points from the border of the original object can be formulated into further objectives.

To calculate the distance between the NURBS surface and a scan point p, p has to be projected onto the surface. The distance between p and the projected point p' equals the distance between p and the NURBS surface. Unfortunately the projection of a point onto a NURBS surface is a very expensive operation [42], which has to be computed for all scan points. Therefore a discrete approximation is used as first objective instead. Discrete, equally distributed points on the NURBS surface are sampled and for each sample point s_{ij} on the NURBS surface the scan point p with the minimal square distance to s_{ij} is determined. The first objective is then calculated as the average of all minimal distances [41]. As second objective the smoothness (regularity) of the NURBS surface is employed. For each sample point s_{ij}, the average squared distance d_{ij} between the normal of s_{ij} and the normal of each of the eight direct neighbors sample points around s_{ij} in the sample grid is calculated. The second objective equals the average of all d_{ij}.

The deterministic algorithm only optimizes the surface regarding the first objective. A system of linear equations is formulated [41] and solved with singular value decomposition (SVD) [43]. As there are a lot more scan points than control points, the system of linear equations is overdetermined and therefore, in most cases no exact solution exists. However, an implementation of SVD, which minimizes the sum of squared errors over all equations is available [44]. The solution obtained from the deterministic algorithm can be optimized further by an evolutionary algorithm (SMS-EMOA) to find Pareto optimal solutions regarding multiple objectives. The hybrid approach passes on the results computed by the deterministic to an evolutionary algorithm. As described above, the shape of a NURBS surface can be modified by changing the positions of one or more control points. The position of the control points (three coordinates per control point) is encoded into the allele, thus each individual represents a different control net. Each dimension in decision space represents a coordinate of one control point, the dimensionality of the problem is $D = 3 \cdot n \cdot m$. The population is initialized with the solution determined by the deterministic approach and randomly mutated versions of this solution. Given the expensive operations necessary to evaluate a NURBS surface regarding the described objectives, only approximately 20000 function evaluations can be computed in a reasonable amount of time.

Issues

Both the deterministic and the hybrid approach described above are only able to attain small parts of the Pareto front. The deterministic algorithm computes only one solution, the evolutionary algorithm extends this solution with many more Pareto optimal solutions. The final results are still very similar surfaces with good fitness values regarding the first objective, but regarding the regularity all surfaces offer similar fitness values. The main problem is lack of diversity in decision space among the solutions. In order to obtain a greater variety of solutions the diversity in decision space needs enhancement.

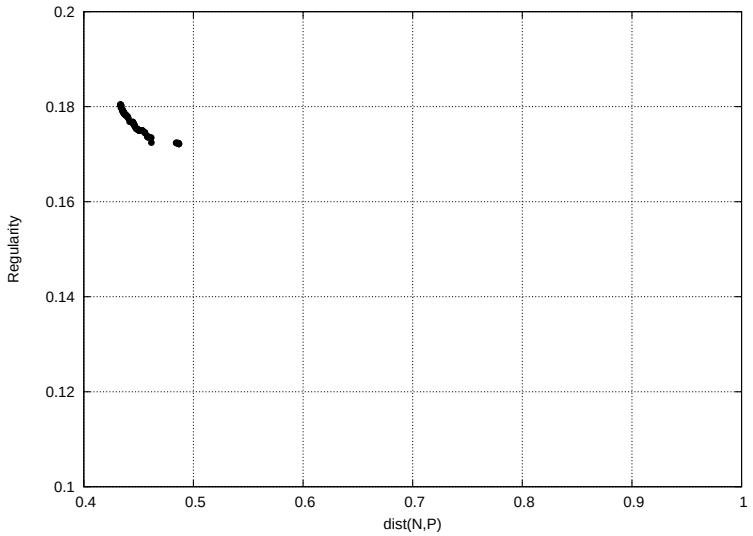


Figure 2.11: Pareto front attained by SMS-EMOA on surface reconstruction

Chapter 3

Approach

This chapter features the presentation of general enhancement ideas for GDE3. Improvements to both diversity in decision space and the quality of the solutions are incorporated into GDE3 forming several new algorithms whose behavior is then systematically analyzed on several test problems.

3.1 Enhancements to GDE3

Because preserving additional diversity in decision space is expected to decrease performance of the modified algorithm, it is necessary to develop some improvements to attenuate this performance impact. Except for the variation operator, GDE3 [5] is very similar to NSGA-II [6] so it is feasible to adapt improvements already proposed for NSGA-II to GDE3. The following enhancements originally employed by SMS-EMOA are explored for potential benefit to GDE3:

- hypervolume as secondary selection criterion,
- steady-state approach

Unfortunately, none of these measures aim at additional diversity. Further improvements are required. Since the major task is to improve diversity in decision space, an algorithm preserving that diversity needs additional information on the structure of the population, on which it works. Clustering algorithms [45] are made for exactly this purpose: to obtain information of the structure of the input. Based on a clustering both

- breeding and
- selection

of an evolutionary algorithm can be modified towards better performance and/or more diversity in decision space.

3.1.1 Modified selection

Preserving diversity in the decision space can be realized in either the breeding and/or selection phase of an evolutionary algorithm respectively. Thus, two approaches, which can, if desired, easily be combined, are explored in this work - one in the breeding, the other in the selection phase. Both approaches make use of a cluster analysis in decision space.

A cluster analysis or clustering [46, 47] is the creation of a group structure in a set of objects [48]. Such a group structure partitions the set of objects into groups of objects with similar properties in each group. In order to assign an object to a group and thus declare it similar to the other members of that group, a distance measure, e.g. the euclidean distance, can be used, but depending on which distance measure describes similarities between objects best, other distance measures, like city-block distance or even statistical operators can be appropriate as well [49]. The general idea behind clustering is, that if the similarity between the objects is low, those objects should not be put into the same cluster, while in the opposite case, when they are much alike, they ought to be. Many standard clustering algorithms are available. Testing all these algorithms would dramatically increase the number of experiments to conduct so the hierarchical agglomerative clustering method (HACM) [49] is used because it is very popular, easy to understand/demonstrate and yet fast and hence an efficient approach. The HACM algorithm is a heuristic bottom-up approach, so initially all objects are located in their own cluster. Until a stopping criterion is reached, the two clusters C_1 and C_2 with the minimal inter-cluster distance are united. Such stopping criterion can either be:

- the number criterion, meaning the number of clusters equals a fixed value C, or
- the distance criterion, stopping the algorithm in case the minimal inter-cluster distance exceeds a fixed value.

For evolutionary algorithms the number criterion has several advantages over the distance criterion, because it is very intuitive and mostly problem independent, just like evolutionary algorithms themselves are and it is therefore easy for the user to understand and set a desired value. If the number of clusters is constant, the evolutionary algorithm behaves similar from beginning to end making both design and research of the such an algorithm much easier. Partitioning into multiple clusters is used as a means to enforce preservation of diversity, the number of clusters gives the user control over how much diversity to preserve so if, due to the distance criterion, clusters are merged, this mechanism does no longer function properly. In the worst case, where only one cluster remains, all efforts towards preserving diversity have been nullified. On the other hand, having a cluster containing multiple individuals is required to perform local search (see section 3.1.2). Obviously, there is at least one cluster with a minimum of four individuals in it, if $\frac{\mu}{C} \leq 4$, because

otherwise the total number of individuals in the population would be $< \mu$. Guaranteeing the existence of a cluster with enough individuals for local search is not possible if the distance criterion is used. Given the limitations of the distance criterion and the benefits of the number criterion, only the later seems promising and will therefore be investigated for use in the proposed algorithms. For the inter-cluster distance also several approaches can be implemented:

1. single link (equation (3.1)),
2. complete link (equation (3.2)) and
3. average link (equation (3.3))

The single link distance $dist_{SL}$ between two clusters C_1 and C_2 is the minimal distance between two objects $a_i \in C_1$ and $b_j \in C_2$.

$$dist_{SL}(C_1, C_2) := min(dist(a_i, b_j)),\ \forall a_i \in C_1, \forall b_j \in C_2 \tag{3.1}$$

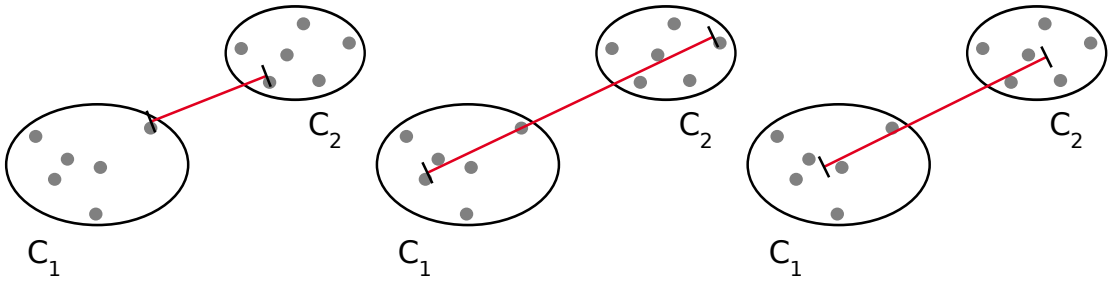
Analog to the single link distance $dist_{SL}$, the complete link distance $dist_{CL}$ uses the maximum distance between two objects $a_i \in C_1$ and $b_j \in C_2$.

$$dist_{CL}(C_1, C_2) := max(dist(a_i, b_j)),\ \forall a_i \in C_1, \forall b_j \in C_2 \tag{3.2}$$

Instead of using the distance between only two objects (one from each cluster), the average link distance $dist_{AL}$ uses the mean pairwise distance between all objects $a_i \in C_1$ and $b_j \in C_2$.

$$dist_{AL}(C_1, C_2) := \frac{1}{|C_1||C_2|} \sum_{a_i \in C_1} \sum_{b_j \in C_2} dist(a_i, b_j) \tag{3.3}$$

Given n objects for both $dist_{SL}$ and $dist_{CL}$ clustering algorithms are known with a computational complexity of $\mathcal{O}(n^2)$, for $dist_{AL}$ the same computational complexity can only be achieved under certain conditions [45]. Again because the number of experiments is limited, only one of the inter-cluster distance methods can be analyzed. This choice is made in favor of the optimally efficient and very fast SLINK [50] algorithm. The major disadvantage of SLINK is its tendency to build clusters shaped like a chain [45]. However, this effect is beneficial for the improvement intended, because during optimization evolutionary algorithms often form chains of individuals in decision space, with the best individuals in the front of the chain. Children, that are not deleted are located in the direction where



(a) Single link (b) Complete link (c) Average link

Figure 3.1: Inter cluster distance

better fitness or at least a better secondary criterion value can be found. This way these "chains" move forward towards better fitness values as better solutions are obtained and the worst individuals are deleted. Clustering via SLINK can thus help to build clusters that support the search of better solutions in different areas of decision space at the same time and in the same way the unmodified algorithm would do in only one region without the clustering, which is exactly what this enhancement is meant for. Future research could examine the benefit of other clustering algorithms than HACM, further inter-cluster distance methods than $dist_{SL}$ and more distance measures. The idea behind increasing

Algorithm 7 HACM (hierarchial agglomerative clustering)

1: $n \leftarrow |Objects|$
2: **for** $i = 1$ to n **do**
3: cluster($Object_i$) $\leftarrow C_i$ // each individual in its own cluster
4: **end for**
5: **repeat**
6: **for all** C_i **do** // for all cluster C_i
7: **for all** $C_j \neq C_i$ **do** // for all cluster $C_i \neq C_j$
8: $dist(C_i, C_j) \leftarrow$ interClusterDistance(C_i, C_j)
9: **end for**
10: **end for**
11: $C_i \leftarrow C_i \cup C_j$, where $dist(C_i, C_j) \leqslant dist(C_k, C_l)\ \forall$ cluster C_k, C_l
12: $C_j \leftarrow \emptyset$
13: **until** criterion satisfied

diversity in decision space is to limit the competition in the selection process to a subset of the population. In order to survive, a child must compete against individuals located in a similar area of decision space. Since individuals located in the same area in decision space have similar properties, a child can displace another (similar) solution without decreasing the diversity in decision space much.

Below different enhancements to both performance and diversity are discussed. First measures to improve diversity through a modified selection phase are described. These measures make use of a cluster analysis in decision space. If the cluster analysis is already computed, the breeding phase can also be modified to make use of this analysis. Additionally some standard concepts known from other MOEA are considered to further improve the performance of the GDE3 algorithm.

Replace worse predecessor

Due to the nature of differential evolution, a child is in many cases in proximity to its predecessor, as all unmodified dimensions are a copy of the corresponding dimensions of the predecessor. So if not many dimensions are modified or the modifications in each dimensions are not too big, the child resides in the same area as the predecessor and can therefore compete against it. Whoever survives, child or predecessor, the diversity is not changed by much. The major problem is that, the better the solutions get, the less likely an individual can dominate its already good predecessor and many children that could have dominated other individuals in the population are deleted and many function evaluations are wasted.

Replace worst in cluster

It is very intuitive that a cluster determined by cluster analysis can determine such an area in decision space to where competition is restricted. The algorithm only has to calculate into which cluster the child is bred and determine whether its fitness justifies its survival the same way as if the contents of the cluster were the entire population. Obviously good solutions that just reside in a cluster with better solutions can be deleted, while worse solutions survive in another area in decision space, this change alone is expected to decrease the performance of the entire population in favor of better diversity. To reduce the amount of clusterings performed, like in GDE3 the child first competes against the predecessor and only in case it fails the cluster analysis is performed.

3.1.2 Cluster analysis while breeding

NSGA-II, GDE3 and SMS-EMOA choose the parents uniformly distributed at random because no information on the individuals, that can indicate which parents to mate for a better child is available and thus the random choice is the only option. However, with the information from the clustering already available, a more systematic approach can be implemented to choose parents in order to influence the location of the child in decision space. Mating parents via differential evolution from the same area in decision space for instance results in a child also located in proximity to the parents. If all the parents are located in completely different regions of the decision space, the child lies somewhere

entirely different than most of the parents, possibly even in completely unexplored parts of the decision space. So the algorithm can then perform both a locally limited exploitation (local search) and an exploration of new areas on a global scale (global search). While local

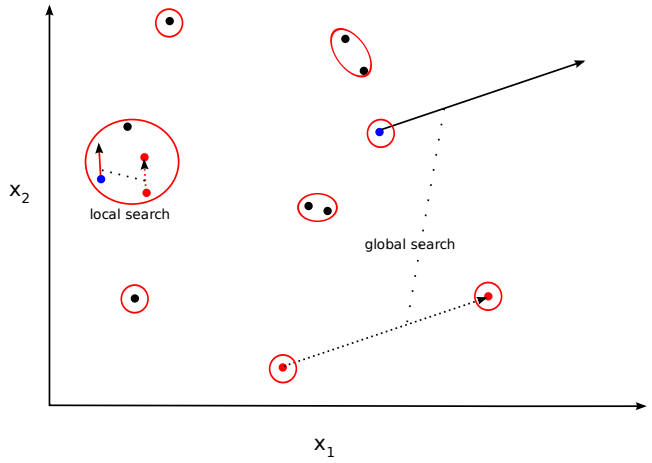


Figure 3.2: Local and global search (based on a clustering)

search offers fast convergence against a local optimum, global search ensures that leaving that optimum is still possible because not yet searched parts of decision space are also explored. Finding the right balance between local and global search is crucial to achieve good performance. An additional parameter L is introduced allowing the user to configure the probability for local and global search.
Differential evolution requires four parents. In some situations choosing all parents from the same cluster for local search may not be possible because there are less than four individuals in that cluster. The following cases can happen:

- Only one individual in the cluster. This individual is the predecessor, all other parents are chosen at random from the entire population.
- Two or three individuals are available. In this case one of them is predecessor, while another is chosen randomly as point of origin for the differential evolution (p_3). The difference between the other two parents p_1 and p_2 determines the distance the child is located away from p_3 in the dimensions modified. Since local search is desired, this difference needs to be small. As already explained above, the distance between two individuals in the same cluster is small, so both p_1 and p_2 are chosen at random from another also randomly chosen cluster that contains at least 2 individuals.

If there is only one cluster, this enhancement falls back the differential evolution from GDE3. Please note that differential evolution itself is left entirely unchanged, only the way parents are chosen differs in this approach. Thus, invariance properties of this variation operator remain unchanged.

3.1.3 Hypervolume vs Crowding-Distance

In NSGA-II and GDE3, the crowding-distance serves as a secondary selection criterion to evaluate individuals. In many cases several individuals are Pareto optimal and cannot be compared using the non-dominance relation. However, the target of a multi-criterion optimization algorithm is to find large parts of the Pareto front and ideally provide an equidistant sampling to not neglect or prefer parts of the front. As an evolutionary algorithm does not have any information on how the entire Pareto front looks like, it can only try to spread the individuals as fair as possible along the parts of the Pareto front found so far. However the crowding-distance (see chapter 1) is unable to properly approximate the crowding of the solutions for more than two objectives [51]. The hypervolume on the other hand does not suffer from this problem as severe as the crowding-distance does because, unlike the perimeter, the volume shrinks dramatically if one side of a cube is decreased and can therefore generate a much more fair distribution in objective function space by preferring other individuals with a bigger volume contribution in such a case. The downside is, that complexity of hypervolume calculation is much higher for more than three dimensions (see table 2.1). As all benchmarks are limited to three dimensions and efficient algorithms are known for those cases, the amount of additional workload on the improved algorithm is minimal, especially for small populations.

3.1.4 Steady-state

Using a steady-state approach like SMS-EMOA is usually an improvement in many ways. First of all, each generation, only one child is created, so a child from generation g can be parent from $g+1$ onwards. If a child survives, it can immediately be chosen as a parent when the next child is bred. Since the fundamental principle of the success of an evolutionary algorithm is that better parents can breed even better children, the fitness of the parents is imperative towards improvements of the solutions. The steady-state approach provides the best parents available in the population. Also due to the monotony of the S-metric, the population can only get "better" (according to the S-metric) as only the individual with the worst hypervolume contribution is removed from the last front, thus the S-metric value of the overall population cannot decrease but improves with any selected child. A positive side-effect is, that the algorithm's behavior can be analyzed much easier, as with each generation only one child can (but not necessarily does) replace another individual in the population and no intermediate state of relevance is to be taken into account. Also for any other $\mu + \lambda$ approach it is difficult, sometimes even impossible to say for sure which child replaced which individual. For the same reasons as already pointed out for SMS-EMOA this measure does not have a significant downside for the problems considered.

3.2 Created algorithms

To find out if and how each modification actually improves the algorithm, all enhancements must be tested separately. To limit the number of different algorithms to test, the steady-state approach is only used on algorithms where it is required for the algorithm to work and the crowding-distance is only employed as a secondary selection criterion for algorithms based directly on GDE3. All other algorithms make use of the hypervolume contribution instead. The assumption behind this is, that hypervolume contribution is by design a more fair approach for distribution of individuals along the Pareto front, while the crowding-distance approximates a fair distribution to reduce computational complexity. All algorithms proposed share a common basic structure (see algorithm 8). The changes are implemented in the functions selectParents(P_g) and addToCompetition($child_i$). In the unmodified algorithms runCompetitions() represents the unchanged selection.

Algorithm 8 Basic algorithm

1: $g \leftarrow 0$ // initialize generation counter
2: $P_g \leftarrow$ initializePopulation()
3: $f \leftarrow \mu$ // number of function evaluations performed
4: **while** $f + \mu < f_{max}$ **do** // generation loop
5: $P_{temp} \leftarrow P_g$
6: **for** $i = 0$ to $\mu - 1$ **do**
7: // breed
8: $parents_i \leftarrow$ selectParents(P_g)
9: $pred_i \leftarrow P_g[i]$ // predecessor
10: $d_{rand} \leftarrow$ rand(0,$D-1$) // random number from $[0, D-1]$
11: $child_i \leftarrow$ differentialEvolution($parents_i$, $pred_i$, d_{rand})
12: // evaluate
13: evaluate($child_i$) // determine fitness
14: // select part 1
15: **if** $child_i \preceq pred_i$ **then** // child weakly dominates its predecessor
16: $P_{temp}[i] \leftarrow child_i$ // replace predecessor with child
17: **else if** $pred_i \not\preceq child_i$ **then** // predecessor does not dominate child
18: addToCompetition($child_i$) // child must prove itself against a set of individuals
19: **end if**
20: **end for**
21: runCompetitions() // competitions
22: **end while**

3.2.1 Naming scheme

To keep the improved algorithms apart, a naming scheme is introduced, splitting the name into up to 3 parts separated by the "-" character.

- If, like in GDE3, the entire population is used in the selection phase, the name begins with GDE3, otherwise it starts with DE (for differential evolution).
- In case hypervolume contribution is the secondary selection criterion, the middle name is "SMS" (for S-metric selection), otherwise the middle name is omitted.
- Algorithms, in which in the selection phase an individual only competes with its predecessor, the last name is "P", where an individuals competes with those in the same cluster, the last name is "C".
- "D" is appended to algorithms that use clustering during breeding.

	Based on GDE3	**Otherwise**
Base name	GDE3	DE
	Hypervolume	**Crowding-distance**
Middle name	SMS	-
	Replace predecessor	**Replace worst in cluster**
Last name	P	C

Table 3.1: Naming scheme for proposed algorithms

3.2.2 Additional parameters

Along with the enhancements, two parameters are introduced to increase flexibility of the algorithm and allow the user to fine tune the behavior to the specific problem at hand. Parameter L (equation (3.4)) determines the probability of local search.

$$search\_type = \begin{cases} local, & rand(0,1) < L \\ global, & otherwise \end{cases} \tag{3.4}$$

During local search, if possible, all parents are chosen from the same cluster. For global search on the other hand all parents are picked from a mutually different cluster. The second additional parameter C is the number criterion for the clustering algorithm and determines into how many clusters the population is partitioned.

3.2.3 The algorithms

GDE3-D is the same as GDE3 but with the improved variation operator, which makes use of clustering. Replacing the crowding distance with hypervolume contribution, results in the algorithm with the name GDE3-SMS-D. Both algorithms feature the same parameters:

- population size μ
- step size factor F
- crossover probability CR
- local search probability L
- number of clusters C

The modified GDE3 using hypervolume contribution as the secondary selection criterion is called GDE3-SMS, since no additional parameters are introduced, they are the same as in GDE3. DE-SMS-P, DE-SMS-C, DE-SMS-PD and DE-SMS-CD integrate the diversity enhancements. While in DE-SMS-P and DE-SMS-PD a child competes only against its predecessor, in addition to that it can compete with all members from the same cluster it would reside in, if it was selected, in DE-SMS-C and DE-SMS-CD. Additionally, DE-SMS-PD and DE-SMS-CD also use the clustering during the breeding phase and thus combine both approaches to enhance diversity. Unfortunately in the worst case, the clustering has to be computed twice for DE-SMS-CD in each generation because if the child is temporarily added to the population in the selection phase, the output of the clustering can change. Both DE-SMS-C and DE-SMS-CD should only be implemented as a steady-state algorithm, because all n children created in one generation can get deleted if the clustering algorithm puts them together into one separate cluster when they are in close proximity to each other. The selection would then have to remove the worst n of n children from the cluster. In DE-SMS-CD this scenario is very likely to happen in small clusters, as during local search all parents are taken from the same cluster and the child is expected to be located near to its parents.

The distance matrix, which needs to be calculated by the HACM algorithm does not change, as only the distances from the child to all other individuals need to be appended. This can be done in $\mathcal{O}(\mu)$ instead of $O(\mu^2)$ to recalculate the entire distance matrix. Because each entry in the distance matrix still needs to be visited once, the computational complexity of the clustering algorithm does not decrease with this performance optimization. Analog, the same optimization can be performed if the child replaces another individual. In case the child is dropped, the population remains the same and the same clustering can be used in the next generation.

SMS-EMOA(DE) is an alias for SMS-EMOA with differential evolution as variation operator. The name is only introduced to tell the algorithm apart from SMS-EMOA using

SBX and PM in the tables and diagrams. As SMS-EMOA, this algorithm is investigated for comparison purposes only, as none of the diversity or performance enhancements proposed in this work are applied (except for the steady-state approach, which is integrated into SMS-EMOA by default and thus no improvement implemented by the author). To show a performance improvement of the basic algorithm aside from the variation operator, SMS-EMOA needs to be benchmarked with the same variation operator as the other algorithms.

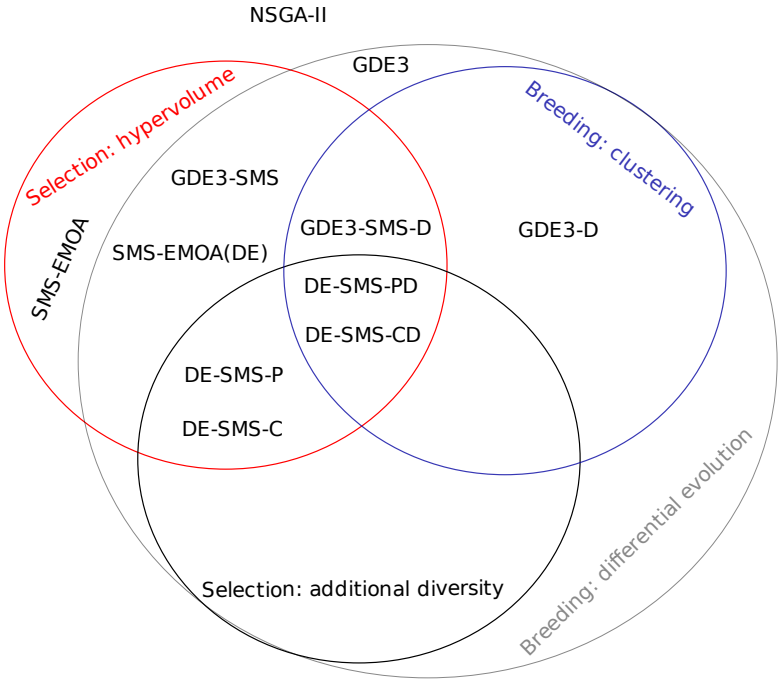


Figure 3.3: Names and features of the algorithms

Algorithm	λ	**Breeding**	**Competition**	**Secondary crit.**	**Parameters**
GDE3	μ	random	all	CD	μ, F, CR
GDE3-SMS	μ	random	all	HV	μ, F, CR
SMS-EMOA(DE)	1	random	all	HV	μ, F, CR
GDE3-D	μ	local/global	all	CD	μ, F, CR, L, C
GDE3-SMS-D	μ	local/global	all	HV	μ, F, CR, L, C
DE-SMS-P	μ	random	predecessor	HV	μ, F, CR
DE-SMS-PD	μ	local/global	predecessor	HV	μ, F, CR, L, C
DE-SMS-C	1	random	same cluster	HV	μ, F, CR, C
DE-SMS-CD	1	local/global	same cluster	HV	μ, F, CR, L, C

Table 3.2: Overview on the created algorithms

3.3 Parameter optimization

Because of the modifications proposed, the new algorithms are expected to behave in a different way than the original GDE3 or SMS-EMOA. No recommendation can be made towards what parameter combination is useful and shows good results for which algorithm on what test problem. This section will therefore explore several parameter sets via experiments on some test problems in a systematic manner to find:

1. Which modification or which combination of modifications attains the best performance on the test problems.

2. Which parameter set is suitable for which algorithm on which test function and which parameter combination is all right for all problems.

3. Which algorithm can preserve more diversity than SMS-EMOA and GDE3 while also producing solutions of acceptable quality [1].

3.3.1 Pre-experimental planning

To find a good parameter set for each algorithm, first the interval for the values of each parameter to be examined must be defined. In a parameter optimization process this is called the region of interest [52] for this parameter. Due to the different nature of each parameter, the intervals of course are different and largely depend on what is to optimize under which circumstances. Of course just guessing a parameter set for each new algorithm is possible, but neither is it systematic, nor does it give accurate and fair results on the performance and potential of the new algorithm, so no serious, profound result can be expected. On the other hand, trying every theoretically possible parameter combination is impossible, as there is an unlimited number of those combinations, because for most parameters the region of interest is an interval containing an infinite number of real values. Probing a limited number of multiple parameter sets from the space of entire region of interest from all parameter intervals in a systematic way is therefore the only available option that can satisfy both scientific procedure and quality of the results. In this case both performance and diversity in decision space are a target to be optimized. The effort is limited by the fact that no more than 20000 function evaluations are allowed. The region of interest of a parameter, which represents a similar feature several algorithms share, is the same for all these algorithms.

[1] acceptable means, the solutions shall be of similar quality to the results calculated by GDE3 and SMS-EMOA

Population size μ

Diversity of a population is limited on the lower bound by the population size, performance is limited on the upper bound. Because parents are chosen uniformly distributed from the population or a subset thereof, each individual in the population will only be improved a couple of times if the population size is large. Diversity of a small population is limited by the small amount of individuals - individuals that are not there cannot contribute to neither diversity nor performance, even though a small population can be good for performance, as each individual can be improved by the algorithm more often. A good balance for the μ parameter has to be a compromise between the described extremes. The minimum required population size for GDE3 is 4, but for the cluster analysis in the breeding phase to work there need to be at least four individuals in one cluster for local search and a minimum of three clusters for global search. To guarantee there are enough individuals in one cluster, the population size needs to be at least 10 (one cluster with four individuals and the other ones with three each in the worst case).

Remember: this does not guarantee, that at any time both local and global search are possible for any predecessor, it only proves there are at least four predecessors for which both local and global search can be performed.

Given diversity in decision space is the main goal of this work, the absolute minimum of individuals required is not a wise choice for the lower bound of parameter μ and therefore the double of the minimum value required by at least one of the algorithms is taken as minimum value. Large population sizes are also not feasible for a small amount of function evaluations (see above), the usual value recommended by the authors of GDE3, NSGA-II and SMS-EMOA is $\mu = 100$. Since with this value the diversity of at least SMS-EMOA and possibly NSGA-II and GDE3 is too small for surface reconstruction and those algorithms do not support any direct means of preserving diversity in decision space, a bigger upper limit is necessary. The triple recommended value ($\mu = 300$) seems a good choice for the upper bound, initial experiments show beyond $\mu = 300$ none of the algorithms was able to reach decent solutions on most of the test functions.

Step size factor F, crossover probability CR

For GDE3 the author recommends [53] a low value (e.g. $F = 0.2$) for multi-objective optimization, but also notes, that especially for ZDT4 $F = 0.5$ is suitable because the local optima are equally distant in decision space on that test problem [54]. Due to the different breeding in GDE3-D, GDE3-SMS-D, DE-SMS-PD and DE-SMS-CD the interval $F \in [0.1, 0.9]$ is investigated for all algorithms. Analog to F, CR is also taken from the same interval ($CR \in [0.1, 0.9]$).

Local search rate L

On adequate values for this entirely new parameter nothing is known, the entire possible interval between 0 (only global search) and 1 (only local search) is examined.

Number of clusters C

The number of clusters shall not exceed $\frac{\mu}{4}$ to guarantee that local search is possible, the more clusters there are, the more diversity in decision space is expected in the solution for DE-SMS-C(D) and too much diversity results most likely in bad quality of the solutions because for many problems there are only few regions in decision space, where good results can be found, thus covering large amounts of the rest of the decision space inside local optima is a bad idea. As for the lower bound, at least one cluster is required, for the minimal population size $\mu = 20$, C has to be $\frac{\mu}{20}$. The region of interest is then $C \in [\frac{\mu}{20}, \frac{\mu}{4}]$.

Parameters from SBX and PM

To also give NSGA-II and SMS-EMOA a fair chance in this comparison, their parameters are also optimized. The mutation probability p_m, normally recommended D^{-1} is examined between $p_m \in [D^{-2}, D^{-0.5}]$, the crossover probability p_c is taken from $p_c \in [0.5, 1]$ (usually used value is $p_m = 0.9$ or $p_m = 1$). As for the distribution indices η_m and η_c, the region of interest is chosen $\eta \in [5, 30]$. All parameter intervals were chosen to roughly cover an area where the authors recommended values or the usually used ones are located in the middle of that interval, so both increased and decreased values of each parameter can be evaluated where possible.

Parameter	Lower bound	Upper bound	Recommended
μ	20	300	100
F	0.1	0.9	0.2 or 0.5
CR	0.1	0.9	0.1 or 0.2
L	0	1	-
C	$\mu/20$	$\mu/4$	-
p_m	D^{-2}	$D^{-0.5}$	D^{-1}
η_m	5	30	20
p_c	0.5	1	1
η_c	5	30	15

Table 3.3: Region of interest for the parameters

Test functions

The benchmark is performed on test problems whose properties can show several desired features an algorithm has to possess in order to perform successfully on these test problems. R_ZDT4 is chosen because it is highly multimodal (multiple, in this case 21^9, local optima exist), so the ability of an algorithm to deal with this kind of situation can be observed. Test problem S_DTLZ3 is also multimodal, but uses three objective functions. It is included in the tests to show whether and how good an algorithm works with more than two objectives on a multimodal test. Both R_ZDT4 and S_DTLZ3 also leave the algorithms much room for improvement, because previously existing, evolutionary algorithms like GDE3 require 500000 or more function evaluations to find a global optimum, if they even find it at all [28]. While R_ZDT4 (10D) and S_DTLZ3 (30D) use many dimensions in decision space, OKA2 [35] only has three dimensions in decision space. This makes it easy to analyze the behavior of an algorithm on that function in decision space by simply plotting and observing the generations. Despite the low dimension decision space, OKA2 has been proven to be difficult for GDE3 [28].

With SYM-PART (see chapter 2.3), a test problem to measure the diversity of the population via the covered sets indicator is also employed. The covered set indicator however has a significant disadvantage, it only measures in which of the global optima at least one solution is approximated, it therefore shows the diversity on a global scale, but cannot meter the diversity in each local optimum. In the worst case scenario each global optimum is covered by only one individual, meaning there is a lot of diversity in decision space, but very few or even no diversity in objective function space. To continuatively examine the diversity in decision space three additional test functions and two indicators are introduced to also measure diversity of the solutions in decision space.

Additional test functions

The concept of the created test functions is very similar to that of SYM-PART - a box-constrained test problem is copied multiple times and the decision space of the copies is patched together in the first two dimensions to form a bigger decision space with multiple optima. Each part is an exact copy of the original decision space of the underlying test problem, it contains a global optimum and is called box from now on. Rudolph et al. name the parts tiles [36] because initially the SYM-PART function only had two dimensions in decision space. The difficulty of the underlying test problem remains the same, which is why also the performance of an algorithm on these test problems can be analyzed, even, or especially if only one optimum is found by an algorithm. For this benchmark S_DTLZ2 (equation (3.7)), S_ZDT1 (equation (3.5)) and S_ZDT2 (equation (3.6)) are chosen as underlying test problems, mostly because they are unimodal, from the very popular test suites ZDT [55] and DTLZ [37] and initial tests indicate that the global optimum

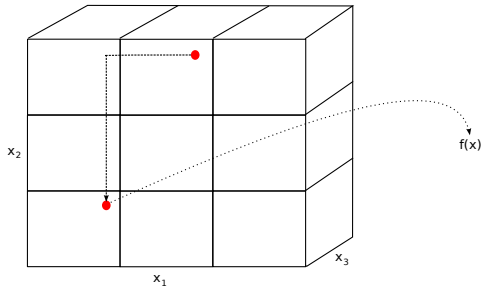


Figure 3.4: Multiple boxes forming the decision space

on S_DTLZ2 and S_ZDT1 can be found at least by GDE3 with only 20000 function evaluations. S_ZDT2 is also included because it is the concave version of S_ZDT2, the HVset indicator is expected to be more meaningful (see below). Even though the underlying test problem is unimodal in all three cases, the created test function is obviously multimodal because the optimum is copied multiple times. The resulting test functions are named M_underlying_problem (e.g. M_S_DTLZ2), the "M_" stands for "multiple boxes of". The fitness values regarding a test function with multiple test problem boxes are calculated in three steps. For the absolute position in decision space:

1. find the box the individual is located in,

2. determine relative position in the box and

3. calculate fitness value for that relative position regarding the underlying test problem.

The relative position in the box is the only important information for calculating the fitness. For the parameter optimization these test problems are used with a 3x3 box grid to have the same number of boxes that SYM-PART has tiles, but in general there is no limit to the total number of boxes as long as they are attached in a rectangle-shaped grid.

Algorithm 9 Calculate fitness value for test problems with multiple boxes

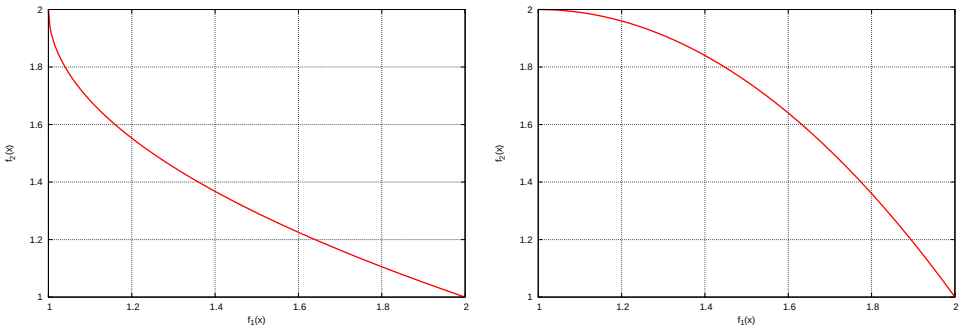
1: **for** $i = 1$ to 2 **do** // box coordinate 1 and 2
2: $b \leftarrow bounds_{hi,i} - bounds_{lo,i}$ // size of the box in dimension i
3: $m \leftarrow \left\lceil \frac{1}{b} \cdot (x_i - bounds_{lo,i}) \right\rceil$ // move left/down by m boxes
4: $x_i \leftarrow x_i - m \cdot b$ // relative coordinate in box
5: **end for**
6: **for** $j = 1$ to D **do** // all objectives
7: $f_{M\_,j} \leftarrow f_j(x_i)$ // assign fitness from underlying problem
8: **end for**

$$
\begin{aligned}
f_1(\vec{x}) &= \begin{cases} z_1' + 1, & z_1 \geq 0 \\ S(p_1)(z_1' + 1), & z_1 < 0 \end{cases} \\
f_2(\vec{x}) &= \begin{cases} g(x)(1 - \sqrt{z_1'/g(x)}) + 1, & \text{all } z_i \geq 0 \\ S\left(\sqrt{\sum_{i=1}^{D} p_i}\right)\left(g(x)(1 - \sqrt{z_1'/g(x)}) + 1\right), & \text{otherwise} \end{cases} \\
&\text{where:} \\
&g(x) = 1 + 9\left(\sum_{i=2}^{D} z_i'\right)/(D-1) \\
&z_i' = \begin{cases} z_i, & z_i \leqslant 0, \\ -\lambda_i z_i, & z_i < 0 \end{cases} \quad p_i = \begin{cases} 0, & z_i \geqslant 0 \\ |z_i|/d_i, & z_i < 0 \end{cases}
\end{aligned}
\tag{3.5}
$$

shift: $\vec{z} = \vec{x} - \vec{o}$,
scale factor: $\vec{\lambda}$

$$
\begin{aligned}
f_1(\vec{x}) &= \begin{cases} z_1' + 1, & z_1 \geq 0 \\ S(p_1)(z_1' + 1), & z_1 < 0 \end{cases} \\
f_2(\vec{x}) &= \begin{cases} g(x)(1 - (z_1'/g(x))^2) + 1, & \text{all } z_i \geq 0 \\ S\left(\sqrt{\sum_{i=1}^{D} p_i}\right)\left(g(x)(1 - (z_1'/g(x))^2) + 1\right), & \text{otherwise} \end{cases} \\
&\text{where:} \\
&g(x) = 1 + 9\left(\sum_{i=2}^{D} z_i'\right)/(D-1) \\
&z_i' = \begin{cases} z_i, & z_i \leqslant 0, \\ -\lambda_i z_i, & z_i < 0 \end{cases} \quad p_i = \begin{cases} 0, & z_i \geqslant 0 \\ |z_i|/d_i, & z_i < 0 \end{cases}
\end{aligned}
\tag{3.6}
$$

shift: $\vec{z} = \vec{x} - \vec{o}$,
scale factor: $\vec{\lambda}$



(a) M_S_ZDT1 (b) M_S_ZDT2

Figure 3.5: Pareto front of M_S_ZDT1 and M_S_ZDT2

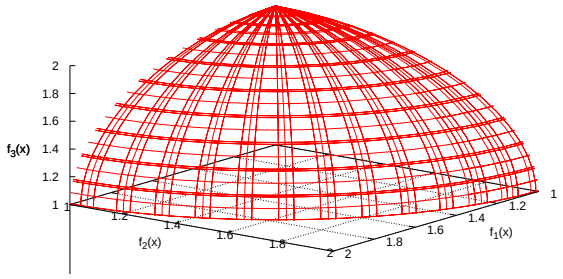


Figure 3.6: Pareto front of M_S_DTLZ2

$$f_1(\vec{x}) = \begin{cases} (1+g(x_M))\cos(z'_1\pi/2)...\cos(z'_{M-2}\pi/2)\cos(z'_{M-1}\pi/2), & z_i \geq 0 \\ S(psum_1)\left((1+g(x_M))\cos(z'_1\pi/2)...\cos(z'_{M-2}\pi/2)\cos(z'_{M-1}\pi/2)+1\right), & \text{otherwise} \end{cases}$$

$$f_2(\vec{x}) = \begin{cases} (1+g(x_M))\cos(z'_1\pi/2)...\cos(z'_{M-2}\pi/2)\sin(z'_{M-1}\pi/2), & z_i \geq 0 \\ S(psum_2)\left((1+g(x_M))\cos(z'_1\pi/2)...\cos(z'_{M-2}\pi/2)\sin(z'_{M-1}\pi/2)+1\right), & \text{otherwise} \end{cases}$$

$$f_3(\vec{x}) = \begin{cases} (1+g(x_M))\cos(z'_1\pi/2)...\cos(z'_{M-3}\pi/2)\sin(z'_{M-2}\pi/2), & z_i \geq 0 \\ S(psum_3)\left((1+g(x_M))\cos(z'_1\pi/2)...\cos(z'_{M-3}\pi/2)\sin(z'_{M-2}\pi/2)+1\right), & \text{otherwise} \end{cases}$$

where:

$g(x_M) = \sum_{x_i \in x_M}(z'_i - \frac{1}{2})^2$

$$z'_i = \begin{cases} z_i, & z_i \geq 0 \\ -\lambda_i z_i, & z_i < 0 \end{cases} \qquad p_i = \begin{cases} 0, & z_i \geq 0 \\ |z_i|/d_i, & z_i < 0 \end{cases}$$

shift: $\vec{z} = \vec{x} - \vec{o}$,

scale factor: $\vec{\lambda}$

(3.7)

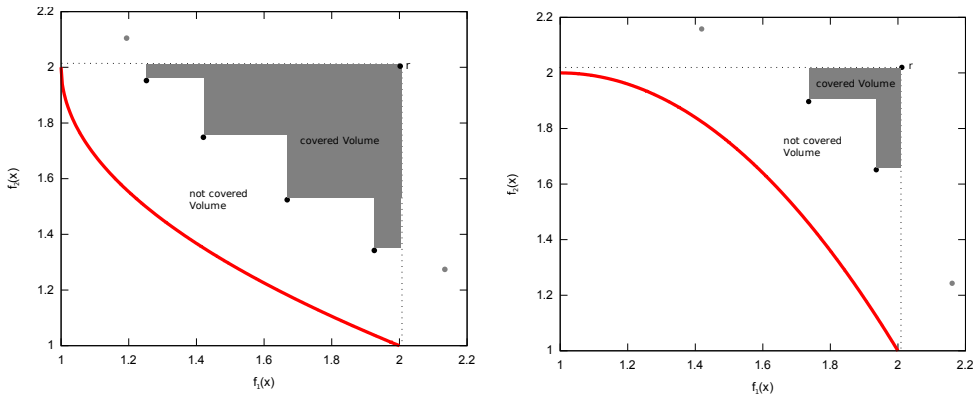
Evaluating the diversity on the proposed test functions could also be achieved by the covered sets indicator, however given the limitations discussed above, a similar, more powerful indicator is desirable, which can measure the diversity on both the global level, like the covered sets indicator does, while still taking the diversity inside one Pareto optimum into consideration.

The HVset indicator

Similar to the covered sets indicator, the HVset indicator first checks whether an optimum is covered. Towards that end, for each local optimum a bounding box is placed around the Pareto front and the hypervolume of the Pareto front in that bounding box is deter-

mined, all individuals outside a bounding box are removed. The sum of these values is the maximum coverable hypervolume on this test function in the bounding boxes. Then the indicator measures the diversity in every optimum separately by computing the hypervolume of the solutions covering the optimum. The sum of these values, divided by the maximum coverable hypervolume can then show how good and diverse the solutions are. An indicator value of 0 means no optimum is covered, the bigger the value gets, the more diverse the solutions are. While the indicator value does not give a total number of sets covered, it shows what fraction of the maximum coverable hypervolume the solution attains, information on how this is done is lost.

Given a HVset value of 0.4 for n boxes does not mean that in $0.4 \cdot n$ boxes the Pareto front is attained very dense, as e.g. 40% of the maximum coverable hypervolume can also be achieved by simply covering 40% of the volume in each of the boxes. But a considerably higher value shows that both diversity and performance of the algorithm are better, for the same test function the values for different algorithms can easily be compared as long as the difference is significant. If the Pareto front is concave, the hypervolume between Pareto front and the bounding box border becomes smaller, the space where not optimal solutions that still get counted can be located is smaller, solutions further from the Pareto front are discarded, which is why using an underlying test problem with a concave Pareto front is preferable.



(a) M_S_ZDT1 (b) M_S_ZDT2

Figure 3.7: HVset indicator calculation

The div+ indicator

To measure the diversity in decision space regardless of the performance of its individuals, no special test problem is required. Instead the amount of diversity can be approximated with the div+ indicator on any population. For each dimension in decision space the

Algorithm 10 Calculate the HVset value

```
1: n ← #boxes
2: max ← n· hypervolume(paretofront) // maximum attainable hypervolume
3: for i = 1 to n do // each box
4:    B_i ← filter(P, i) // only pick the individuals in box i
5:    A_i ← filter(B_i) // set of acceptable solutions in bounding box
6:    hvset ← hvset+ hypervolume(A_i)
7: end for
8: hvset ← hvset/max
9: return hvset
```

div+ indicator sums up the distance between the maximum and minimum value from any individual for this dimension. This indicator is very easy to calculate and understand and yet helpful to measure diversity in many cases. A disadvantage is that only two individuals are taken into consideration for each dimension, if the difference between maximum and minimum value is large, while the space in between them no other individual is located and the space is thus empty, the population seems diverse from the value of the indicator, while all individuals are concentrated in only two regions far apart from each other. If the indicator shows a very small value the diversity is obviously small, hence the indicator is accurately mirroring the diversity in decision space. Instead of the sum of the values for each dimension, one could also use the product, however this div* indicator is not pursued any further because depending on algorithm and test problem the indicator values tend to get either very small or big, sometimes beyond computational accuracy. Therefore indicator values cannot be compared properly if the difference between minimum and maximum is $\ll 1$ or $\gg 1$ for many dimensions in decision space. For many test problems only one value is optimal for at least one dimension in decision space, hence the described problem often occurs.

The All3 indicator

Evaluating solutions with a set of indicators comes with a severe problem: what if indicators contradict each other in the comparison of two algorithms or two parameter sets of the same algorithm? While ε-, hypervolume and R2-indicator evaluate similar properties of solutions, this can easily happen, if the quality of the solutions is not too far apart. Another problem is that the number of comparisions between algorithms increases linearly with the number of indicators used to evaluate the results. To not prefer any indicator in these cases, a helper indicator can be used, which equally combines the value of the three indicators (ε, hypervolume, R2), resulting in only one new indicator value to compare - the All3 indicator value. Sorting by the All3 indicator allows the user to find the best solution almost immediately. This indicator value is calculated as follows: Please note: this

Algorithm 11 Calculate the All3 indicator

$eps_{best} \leftarrow \infty; eps_{worst} \leftarrow -\infty$
$hyp_{best} \leftarrow \infty; hyp_{worst} \leftarrow -\infty$
$r2_{best} \leftarrow \infty; r2_{worst} \leftarrow -\infty$
for $i = 1$ to m **do** // each parameter combination
  for $j = 1$ to n **do** // each run
    $eps[i] \leftarrow$ median(epsilon_ind(P_{ij}))
    $hyp[i] \leftarrow$ median(hypervolume_ind(P_{ij}))
    $r2[i] \leftarrow$ median(r2_ind(P_{ij}))
  end for
end for
$eps_{best} \leftarrow \min(eps[]); eps_{worst} \leftarrow \max(eps[])$
$hyp_{best} \leftarrow \min(hyp[]); hyp_{worst} \leftarrow \max(hyp[])$
$r2_{best} \leftarrow \min(r2[]); r2_{worst} \leftarrow \max(r2[])$
for $i = 1$ to m **do** // each parameter combination
  $eps_{norm} \leftarrow (eps[i] - eps_{best})/(eps_{worst} - eps_{best})$
  $hyp_{norm} \leftarrow (hyp[i] - hyp_{best})/(hyp_{worst} - hyp_{best})$
  $r2_{norm} \leftarrow (r2[i] - r2_{best})/(r2_{worst} - r2_{best})$
  $all3[i] \leftarrow (eps_{norm} + hyp_{norm} + r2_{norm})/3$
end for

indicator is not intended to compare its absolute results in different sets of benchmarks because the value is a relative one, which has to be computed again if results are added, removed or changed. The All3 indicator was primarily developed to find the best parameter combination for one algorithm on a single test function. Analog the same concept can be used to find the best among several algorithms on a single test function. The value range of the All3 indicator is between 0 and 1, with 0 being the best and 1 the worst value.

3.3.2 Task

The task of this optimization process is to find a recommendable parameter combination for each algorithm. With this parameter combination the algorithm shall provide good results on all test problems. Once such a parameter combination is found, the algorithms can be compared with the optimized parameter combination and the reason for their performance regarding both diversity and quality of the solutions according to the metrics can be analyzed. Because only a general recommendation is made, specific optimizations may be required for real applications.

3.3.3 Setup

For each group of algorithms with exactly the same parameters 50 parameter sets are explored via the same Latin Hypercube Design [56] (to ensure the parameter combinations examined are equally distributed among the parameter space). The design file for each group can be found in appendix A. To have a reasonable statistical basis, each parameter combination from the design file is run 25 times on all test problems

- OKA2,
- R_ZDT4,
- S_DTLZ3 (30D),
- SYM-PART (30D),
- M_S_DTLZ2 (30D),
- M_S_ZDT1 (30D) and
- M_S_ZDT2 (30D).

The total number of runs is therefore:

$$runs = \#algorithms \cdot \#testproblems \cdot \#runs \cdot \#parametersets = 11 \cdot 7 \cdot 25 \cdot 50 = 96250 \tag{3.8}$$

For all parameter combinations of the same algorithm, the mean, median and standard deviation value of the

- ε,
- hypervolume and
- R2

indicators are computed and based on these values the All3 indicator value is determined on all test problems separately to easily compare parameter combinations on one test problem at a time. A lower All3 indicator value indicates better performance of the corresponding parameter combination, hence the combination with the lowest indicator value is designated the best combination for this test function found so far. In order to find a good overall combination with which a single algorithm works well on all test problems, for each parameter combination the sum of squares for all All3 indicator values $\sum_{i=1}^{7}(All3_i)^2$ (from each test function) is evaluated. Because all All3 indicator values are from the interval [0,1], the sum of squares prefers smaller values for each part of the sum, bad performance on a single test problem is penalized. The combination with the lowest sum of squares of the All3 indicator values is then declared as good overall parameter combination for the

algorithm under investigation. The performance of these algorithms with the found parameter combination is then analyzed further. Analog to finding the best overall parameter combination for a single algorithm, the sum of squares for the All3 indicator values can be used to determine a ranking among the algorithms (with their best overall parameter combination). To validate the results, a statistical hypothesis test (Kruskal-Wallis test[38] with parameter $\alpha = 0.01$) is employed for the best parameter-combinations on each test problem. In parallel to this analysis, the diversity is measured for all parameter-combination runs with the div+ indicator. In addition to that, the covered sets indicator on SYMPART and the hvset indicator values on M_S_ZDT1, M_S_ZDT2 and M_S_DTLZ2 are compared among the best algorithms to find out which of the algorithms can to what degree preserve diversity.

3.3.4 Results

For comparison purposes, additionally the recommended parameter-combination was evaluated for NSGA-II, SMS-EMOA and GDE3. In the following tables these parameter-combination are marked with a "*". In each table the best indicator value is printed bold.

	Parameters				
Algorithm	μ	F	CR	L	C
DE-SMS-P	28	0.38	0.22	-	-
DE-SMS-C	45	0.49	0.14	-	7
DE-SMS-PD	45	0.49	0.14	0.61	7
DE-SMS-CD	45	0.49	0.14	0.61	7
GDE3	101	0.18	0.18	-	-
GDE3*	100	0.5	0.1	-	-
GDE3-D	163	0.25	0.12	0.65	14
GDE3-SMS	28	0.38	0.22	-	-
GDE3-SMS-D	163	0.25	0.12	0.65	14
SMS-EMOA(DE)	34	0.43	0.56	-	-
	μ	p_m	p_c	η_m	η_c
SMS-EMOA	157	-1.33	0.7	21	5
SMS-EMOA*	100	-1	1	20	15
NSGA-II	157	-1.33	0.7	21	5
NSGA-II*	100	-1	1	20	15

Table 3.4: Best allround parameter set according to $\min(\sum_{i=1}^{7}(All3_i)^2)$

Algorithm	ε-indicator median	ε-indicator mean	ε-indicator std	hyp-indicator median	hyp-indicator mean	hyp-indicator std	R2-indicator med	R2-indicator mean	R2-indicator std	All3 indicator
DE-SMS-P	0.0621	0.0590	0.0330	0.0319	0.0299	0.0186	0.0147	0.0165	0.0127	0.7215
DE-SMS-C	**-0.0024**	0.0088	0.0309	**-0.0048**	0.0016	0.0176	**-0.0024**	0.0014	0.0107	**0.0000**
DE-SMS-PD	0.0466	0.0452	0.0068	0.0176	0.0193	0.0043	0.0077	0.0090	0.0040	0.4743
DE-SMS-CD	-0.0024	0.0290	0.0430	-0.0047	0.0113	0.0224	-0.0024	0.0062	0.0133	0.0002
GDE3	0.0750	0.0738	0.0077	0.0359	0.0359	0.0055	0.0157	0.0156	0.0047	0.8123
GDE3*	0.0806	0.0804	0.0098	0.0421	0.0447	0.0072	0.0189	0.0204	0.0061	0.9176
GDE3-D	0.0408	0.0431	0.0087	0.0196	0.0194	0.0034	0.0104	0.0105	0.0036	0.4991
GDE3-SMS	0.0602	0.0481	0.0341	0.0303	0.0245	0.0202	0.0126	0.0118	0.0105	0.6762
GDE3-SMS-D	0.0418	0.0413	0.0073	0.0181	0.0187	0.0037	0.0094	0.0100	0.0037	0.4800
SMS-EMOA(DE)	0.0466	0.0447	0.0485	0.0226	0.0231	0.0296	0.0075	0.0088	0.0124	0.5051
SMS-EMOA	0.0743	0.0728	0.0130	0.0380	0.0389	0.0069	0.0186	0.0189	0.0062	0.8607
SMS-EMOA*	0.0733	0.0790	0.0191	0.0446	0.0477	0.0130	0.0228	0.0249	0.0095	0.9546
NSGA-II	0.0675	0.0673	0.0083	0.0383	0.0384	0.0046	0.0192	0.0189	0.0040	0.8431
NSGA-II*	0.0721	0.0790	0.0215	0.0456	0.0494	0.0142	0.0235	0.0265	0.0100	0.9656

Table 3.5: Performance with optimized parameter set on OKA2

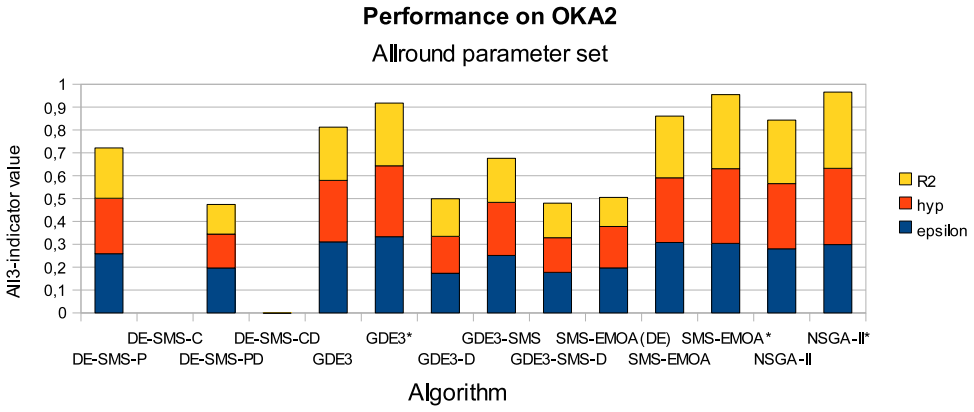


Figure 3.8: All3 indicator on OKA2

Algorithm	ε-indicator median	ε-indicator mean	ε-indicator std	hyp-indicator median	hyp-indicator mean	hyp-indicator std	R2-indicator med	R2-indicator mean	R2-indicator std	All3 indicator
DE-SMS-P	0.0272	0.0274	0.0053	0.0245	0.0245	0.0051	0.0124	0.0125	0.0022	0.2733
DE-SMS-C	0.0175	0.0177	0.0094	**0.0119**	0.0160	0.0095	0.0070	0.0077	0.0045	**0.0218**
DE-SMS-PD	0.0204	0.0224	0.0065	0.0136	0.0152	0.0049	0.0093	0.0094	0.0023	0.0941
DE-SMS-CD	0.0181	0.0210	0.0136	0.0154	0.0173	0.0101	0.0081	0.0090	0.0055	0.0681
GDE3	0.0246	0.0239	0.0094	0.0172	0.0208	0.0089	0.0104	0.0106	0.0042	0.1701
GDE3*	0.0339	0.0355	0.0170	0.0310	0.0317	0.0148	0.0154	0.0160	0.0070	0.4197
GDE3-D	0.0264	0.0292	0.0118	0.0227	0.0256	0.0104	0.0124	0.0129	0.0052	0.2543
GDE3-SMS	0.0171	0.0175	0.0055	0.0176	0.0183	0.0060	0.0079	0.0080	0.0027	0.0710
GDE3-SMS-D	0.0241	0.0246	0.0108	0.0186	0.0217	0.0113	0.0105	0.0110	0.0051	0.1762
SMS-EMOA(DE)	**0.0148**	0.0176	0.0107	0.0159	0.0180	0.0120	**0.0070**	0.0081	0.0053	0.0266
SMS-EMOA	0.0288	0.0315	0.0165	0.0315	0.0343	0.0182	0.0137	0.0151	0.0083	0.3552
SMS-EMOA*	0.0561	0.0580	0.0239	0.0615	0.0589	0.0238	0.0275	0.0277	0.0106	1.0000
NSGA-II	0.0260	0.0257	0.0087	0.0273	0.0276	0.0094	0.0124	0.0121	0.0043	0.2829
NSGA-II*	0.0436	0.0462	0.0172	0.0431	0.0478	0.0174	0.0212	0.0221	0.0083	0.6724

Table 3.6: Performance with optimized parameter set on R_ZDT4(10D)

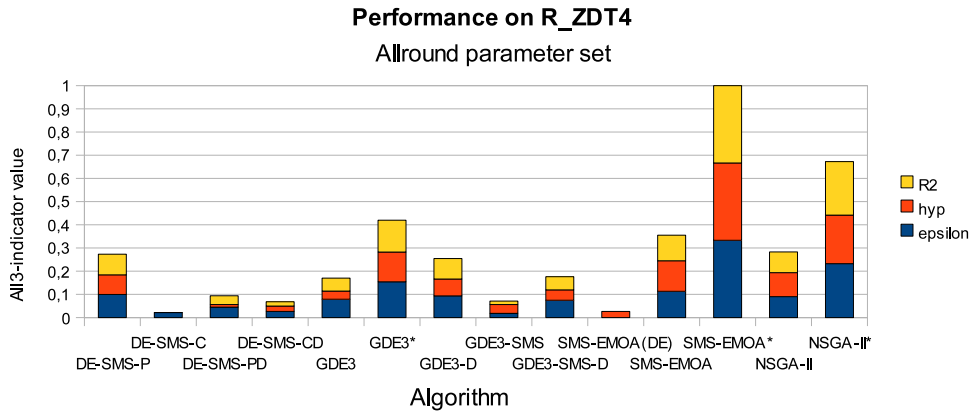


Figure 3.9: All3 indicator on R_ZDT4

	ε-indicator			hyp-indicator			R2-indicator			All3
Algorithm	median	mean	std	median	mean	std	med	mean	std	**indicator**
DE-SMS-P	1.05e-01	1.05e-01	9.87e-03	2.70e-03	5.35e-03	5.78e-03	1.35e-02	1.45e-02	2.74e-03	0.5746
DE-SMS-C	**1.32e-02**	1.43e-02	5.33e-03	**2.12e-05**	4.45e-05	6.71e-05	**1.93e-03**	2.04e-03	7.23e-04	**0.0000**
DE-SMS-PD	7.57e-02	7.47e-02	8.58e-03	1.44e-03	1.46e-03	2.42e-04	1.02e-02	1.02e-02	6.12e-04	0.3826
DE-SMS-CD	1.57e-02	1.69e-02	4.76e-03	2.59e-05	3.35e-05	2.47e-05	2.32e-03	2.40e-03	6.15e-04	0.0143
GDE3	6.33e-02	6.52e-02	9.83e-03	6.90e-04	7.20e-04	1.97e-04	8.71e-03	8.93e-03	8.79e-04	0.2912
GDE3*	4.84e-02	4.91e-02	7.92e-03	4.19e-04	4.34e-04	1.23e-04	7.26e-03	7.21e-03	7.43e-04	0.2122
GDE3-D	8.27e-02	8.23e-02	7.51e-03	1.37e-03	1.37e-03	1.89e-04	1.12e-02	1.11e-02	5.82e-04	0.4182
GDE3-SMS	1.73e-02	1.70e-02	5.24e-03	6.05e-03	5.68e-03	3.39e-03	3.47e-03	3.58e-03	8.94e-04	0.2823
GDE3-SMS-D	7.62e-02	7.49e-02	1.07e-02	1.17e-03	1.18e-03	2.65e-04	1.05e-02	1.05e-02	8.47e-04	0.3795
SMS-EMOA(DE)	9.90e-02	9.84e-02	1.47e-02	5.95e-03	7.52e-03	5.14e-03	1.53e-02	1.55e-02	2.53e-03	0.7233
SMS-EMOA	1.17e-01	1.18e-01	1.23e-02	5.53e-03	5.45e-03	8.08e-04	1.78e-02	1.77e-02	9.96e-04	0.8043
SMS-EMOA*	1.02e-01	9.31e-02	3.07e-02	2.36e-03	2.59e-03	1.47e-03	1.33e-02	1.26e-02	3.51e-03	0.5497
NSGA-II	1.29e-01	1.30e-01	1.40e-02	6.72e-03	6.87e-03	7.90e-04	1.91e-02	1.92e-02	7.91e-04	0.9090
NSGA-II*	1.34e-01	1.36e-01	2.14e-02	8.31e-03	8.17e-03	2.84e-03	1.98e-02	1.94e-02	2.59e-03	1.0000

Table 3.7: Performance with optimized parameter set on S_DTLZ3(30D)

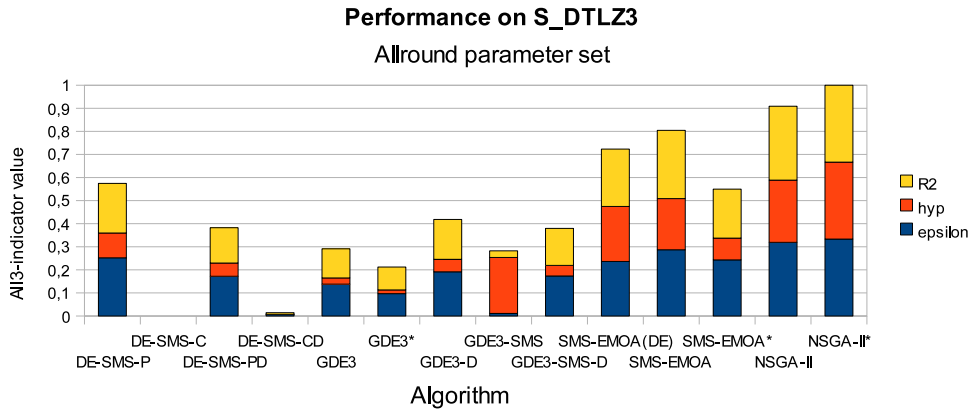


Figure 3.10: All3 indicator on S_DTLZ3

	ε-indicator			hyp-indicator			R2-indicator			All3
Algorithm	median	mean	std	median	mean	std	med	mean	std	**indicator**
DE-SMS-P	3.29e-04	4.67e-04	3.23e-04	6.19e-04	8.19e-04	5.13e-04	4.83e-05	1.11e-04	1.49e-04	0.0059
DE-SMS-C	6.43e-04	6.73e-04	1.80e-04	1.25e-03	1.23e-03	2.65e-04	1.39e-04	1.43e-04	5.87e-05	0.0135
DE-SMS-PD	2.95e-04	3.16e-04	6.74e-05	5.46e-04	5.50e-04	1.24e-04	8.04e-05	8.06e-05	1.67e-05	0.0056
DE-SMS-CD	5.62e-04	6.26e-04	2.61e-04	9.67e-04	1.15e-03	5.01e-04	9.08e-05	1.47e-04	1.27e-04	0.0106
GDE3	3.56e-04	3.83e-04	8.80e-05	7.17e-04	7.30e-04	1.19e-04	4.73e-05	5.22e-05	1.69e-05	0.0066
GDE3*	2.73e-04	2.73e-04	6.35e-05	4.79e-04	5.06e-04	1.04e-04	3.76e-05	3.72e-05	8.13e-06	0.0045
GDE3-D	4.61e-04	4.61e-04	9.36e-05	9.10e-04	9.10e-04	1.57e-04	1.26e-04	1.29e-04	3.22e-05	0.0097
GDE3-SMS	7.96e-04	1.33e-03	9.72e-04	1.37e-03	2.33e-03	1.71e-03	2.46e-04	5.92e-04	5.46e-04	0.0171
GDE3-SMS-D	4.37e-04	4.55e-04	9.08e-05	8.42e-04	8.76e-04	1.73e-04	1.21e-04	1.25e-04	3.32e-05	0.0091
SMS-EMOA(DE)	**7.22e-05**	7.35e-05	5.64e-06	**5.25e-05**	5.45e-05	1.34e-05	**1.21e-05**	1.22e-05	6.05e-07	**0.0000**
SMS-EMOA	1.83e-02	1.80e-02	8.67e-03	3.90e-02	3.76e-02	1.84e-02	1.32e-02	1.28e-02	6.49e-03	0.5613
SMS-EMOA*	3.28e-02	3.11e-02	1.60e-02	6.84e-02	6.43e-02	3.27e-02	2.37e-02	2.23e-02	1.16e-02	1.0000
NSGA-II	6.87e-03	7.57e-03	4.46e-03	1.46e-02	1.45e-02	8.55e-03	4.72e-03	4.73e-03	3.00e-03	0.2064
NSGA-II*	2.85e-02	3.09e-02	1.23e-02	5.61e-02	6.08e-02	2.70e-02	1.93e-02	2.11e-02	9.59e-03	0.8352

Table 3.8: Performance with optimized parameter set on SYM-PART(30D)

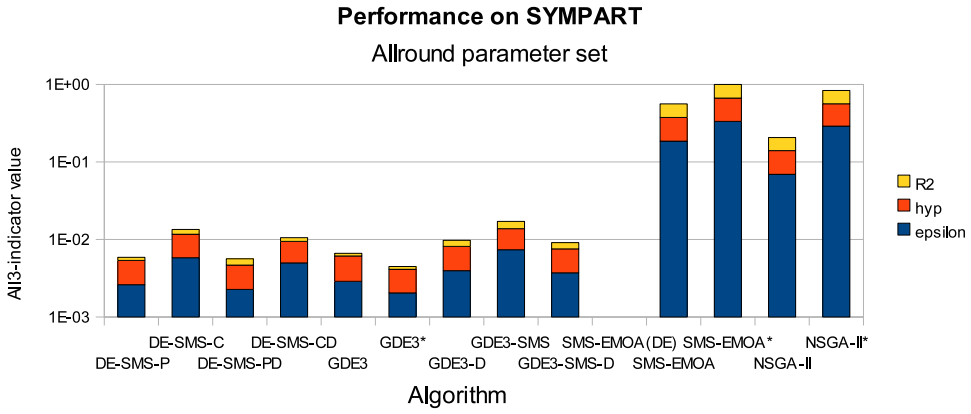


Figure 3.11: All3 indicator on SYM-PART

	ε-indicator			hyp-indicator			R2-indicator			All3
Algorithm	median	mean	std	median	mean	std	med	mean	std	**indicator**
DE-SMS-P	1.10e-02	1.11e-02	8.61e-04	1.73e-04	1.79e-04	5.48e-05	3.73e-04	3.94e-04	7.58e-05	0.0997
DE-SMS-C	9.76e-03	1.71e-02	1.06e-02	1.85e-04	6.78e-04	7.67e-04	2.70e-04	6.61e-04	5.80e-04	0.0782
DE-SMS-PD	8.86e-03	9.06e-03	6.60e-04	2.15e-04	2.25e-04	7.29e-05	2.97e-04	3.04e-04	3.73e-05	0.0792
DE-SMS-CD	9.82e-03	1.56e-02	1.01e-02	2.12e-04	5.79e-04	6.60e-04	2.78e-04	5.96e-04	5.28e-04	0.0862
GDE3	1.22e-02	1.23e-02	2.00e-03	4.48e-04	4.66e-04	2.13e-04	2.93e-04	3.02e-04	7.87e-05	0.1691
GDE3*	1.21e-02	1.26e-02	1.86e-03	7.23e-04	7.78e-04	3.25e-04	4.51e-04	4.81e-04	1.43e-04	0.2524
GDE3-D	1.03e-02	1.06e-02	1.31e-03	4.75e-04	5.07e-04	1.85e-04	3.73e-04	3.73e-04	6.66e-05	0.1648
GDE3-SMS	2.10e-02	2.88e-02	1.44e-02	1.46e-03	1.68e-03	1.45e-03	1.57e-03	1.67e-03	9.69e-04	0.6465
GDE3-SMS-D	**6.17e-03**	6.38e-03	7.79e-04	9.10e-05	1.20e-04	6.38e-05	**1.37e-04**	1.44e-04	2.63e-05	**0.0035**
SMS-EMOA(DE)	9.56e-03	9.61e-03	4.41e-04	1.09e-04	1.16e-04	3.34e-05	2.90e-04	2.91e-04	4.62e-06	0.0601
SMS-EMOA	6.21e-03	6.77e-03	1.40e-03	**7.63e-05**	9.31e-05	4.35e-05	2.93e-04	3.48e-04	1.54e-04	0.0176
SMS-EMOA*	1.67e-02	1.66e-02	5.25e-03	4.17e-04	4.33e-04	2.03e-04	1.32e-03	1.38e-03	5.99e-04	0.3231
NSGA-II	1.96e-02	1.94e-02	2.42e-03	3.53e-04	3.74e-04	7.78e-05	1.04e-03	1.06e-03	2.43e-04	0.3062
NSGA-II*	3.82e-02	3.74e-02	6.22e-03	1.28e-03	1.24e-03	3.69e-04	3.15e-03	3.04e-03	6.91e-04	0.9572

Table 3.9: Performance with optimized parameter set on M_S_DTLZ2(30D)

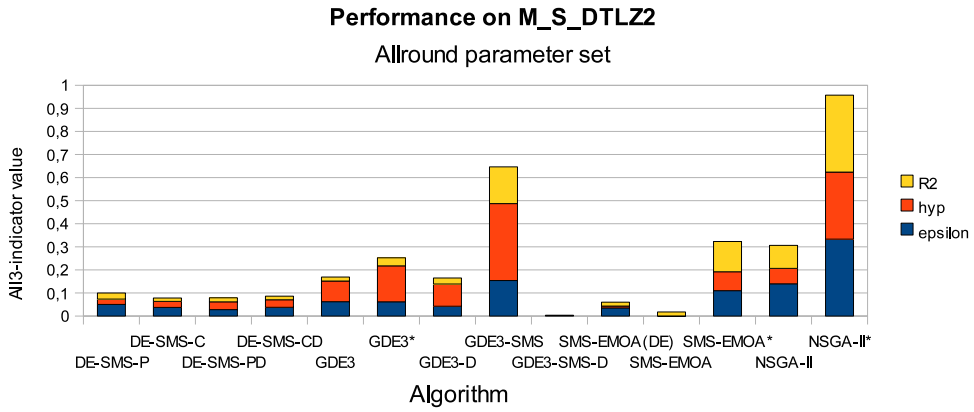


Figure 3.12: All3 indicator on M_S_DTLZ2

	ε-indicator			hyp-indicator			R2-indicator			All3
Algorithm	median	mean	std	median	mean	std	med	mean	std	**indicator**
DE-SMS-P	1.24e-02	1.23e-02	3.33e-03	8.38e-03	9.55e-03	3.58e-03	3.13e-03	3.56e-03	1.37e-03	0.0291
DE-SMS-C	**5.31e-03**	5.45e-03	1.19e-03	**1.78e-03**	2.11e-03	8.35e-04	**5.83e-04**	6.63e-04	2.82e-04	**0.0000**
DE-SMS-PD	8.51e-03	8.69e-03	1.17e-03	5.16e-03	5.16e-03	5.57e-04	1.95e-03	1.94e-03	1.67e-04	0.0144
DE-SMS-CD	5.90e-03	5.83e-03	1.20e-03	2.13e-03	2.42e-03	8.62e-04	6.96e-04	6.92e-04	1.52e-04	0.0018
GDE3	1.38e-02	1.42e-02	5.53e-03	1.15e-02	1.21e-02	5.52e-03	4.50e-03	4.73e-03	2.18e-03	0.0404
GDE3*	5.05e-02	4.98e-02	1.19e-02	4.56e-02	4.63e-02	1.15e-02	1.75e-02	1.73e-02	4.38e-03	0.1903
GDE3-D	4.91e-02	5.81e-02	3.23e-02	4.59e-02	5.41e-02	3.06e-02	1.77e-02	2.04e-02	1.14e-02	0.1891
GDE3-SMS	7.48e-02	8.54e-02	4.50e-02	9.17e-02	1.08e-01	5.54e-02	2.62e-02	3.30e-02	1.86e-02	0.3215
GDE3-SMS-D	2.62e-02	2.85e-02	7.35e-03	2.24e-02	2.48e-02	7.58e-03	8.82e-03	9.78e-03	2.82e-03	0.0899
SMS-EMOA(DE)	5.69e-03	5.90e-03	9.46e-04	2.67e-03	2.79e-03	4.19e-04	1.04e-03	1.04e-03	1.47e-04	0.0034
SMS-EMOA	1.39e-01	1.37e-01	1.18e-02	1.55e-01	1.53e-01	1.44e-02	5.98e-02	5.86e-02	5.53e-03	0.6277
SMS-EMOA*	2.14e-01	2.16e-01	1.92e-02	2.49e-01	2.51e-01	2.41e-02	9.59e-02	9.66e-02	9.43e-03	1.0000
NSGA-II	1.35e-01	1.37e-01	1.65e-02	1.52e-01	1.53e-01	1.97e-02	5.85e-02	5.87e-02	7.56e-03	0.6126
NSGA-II*	2.05e-01	2.07e-01	1.65e-02	2.37e-01	2.39e-01	2.07e-02	9.09e-02	9.18e-02	8.05e-03	0.9515

Table 3.10: Performance with optimized parameter set on M_S_ZDT1(30D)

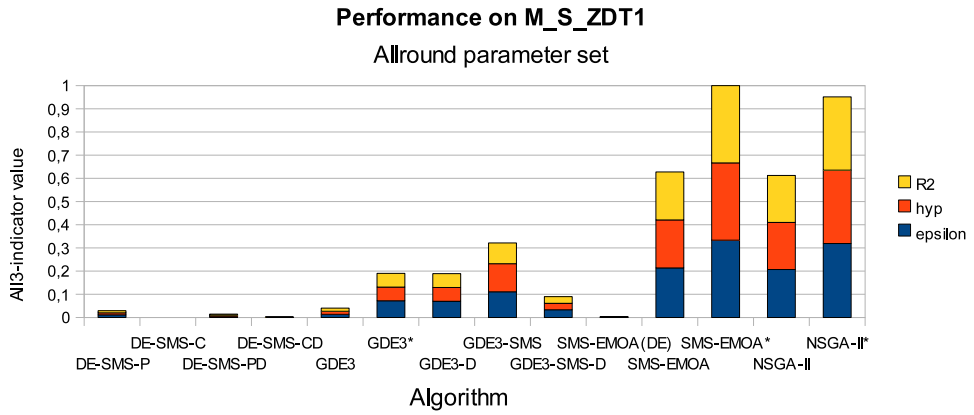


Figure 3.13: All3 indicator on M_S_ZDT1

	ε-indicator			hyp-indicator			R2-indicator			All3
Algorithm	median	mean	std	median	mean	std	med	mean	std	**indicator**
DE-SMS-P	2.00e-02	2.18e-02	8.82e-03	1.69e-02	1.81e-02	7.73e-03	4.74e-03	5.04e-03	2.44e-03	0.0397
DE-SMS-C	6.57e-03	1.81e-02	2.68e-02	1.44e-02	1.77e-02	1.77e-02	4.47e-03	6.49e-03	7.41e-03	0.0255
DE-SMS-PD	1.13e-02	1.14e-02	1.41e-03	7.91e-03	7.84e-03	8.02e-04	1.99e-03	2.00e-03	1.99e-04	0.0185
DE-SMS-CD	7.23e-03	1.01e-02	1.81e-02	4.20e-03	8.33e-03	1.33e-02	1.04e-03	2.66e-03	5.53e-03	0.0097
GDE3	1.61e-02	1.93e-02	2.06e-02	1.18e-02	1.48e-02	1.55e-02	3.30e-03	4.45e-03	4.94e-03	0.0288
GDE3*	8.39e-04	3.49e-02	6.49e-02	1.84e-03	3.13e-02	5.49e-02	8.47e-04	1.15e-02	1.98e-02	0.0018
GDE3-D	4.54e-02	4.71e-02	4.35e-02	3.49e-02	3.96e-02	4.04e-02	1.09e-02	1.31e-02	1.47e-02	0.0903
GDE3-SMS	**4.81e-04**	5.67e-02	9.42e-02	**1.06e-03**	7.31e-02	1.19e-01	**4.85e-04**	2.39e-02	3.96e-02	**0.0000**
GDE3-SMS-D	4.08e-02	3.74e-02	3.46e-02	3.10e-02	2.83e-02	2.42e-02	9.51e-03	8.93e-03	7.64e-03	0.0801
SMS-EMOA(DE)	7.09e-04	6.18e-03	1.53e-02	1.56e-03	1.33e-02	3.26e-02	7.16e-04	6.24e-03	1.54e-02	0.0011
SMS-EMOA	2.58e-01	2.66e-01	3.38e-02	2.36e-01	2.43e-01	3.40e-02	9.52e-02	9.84e-02	1.54e-02	0.6289
SMS-EMOA*	3.92e-01	3.87e-01	3.11e-02	3.76e-01	3.71e-01	3.28e-02	1.58e-01	1.56e-01	1.51e-02	1.0000
NSGA-II	2.68e-01	2.67e-01	1.87e-02	2.46e-01	2.44e-01	1.73e-02	9.86e-02	9.88e-02	7.93e-03	0.6529
NSGA-II*	3.78e-01	3.74e-01	2.74e-02	3.59e-01	3.56e-01	2.88e-02	1.51e-01	1.50e-01	1.33e-02	0.9590

Table 3.11: Performance with optimized parameter set on M_S_ZDT2(30D)

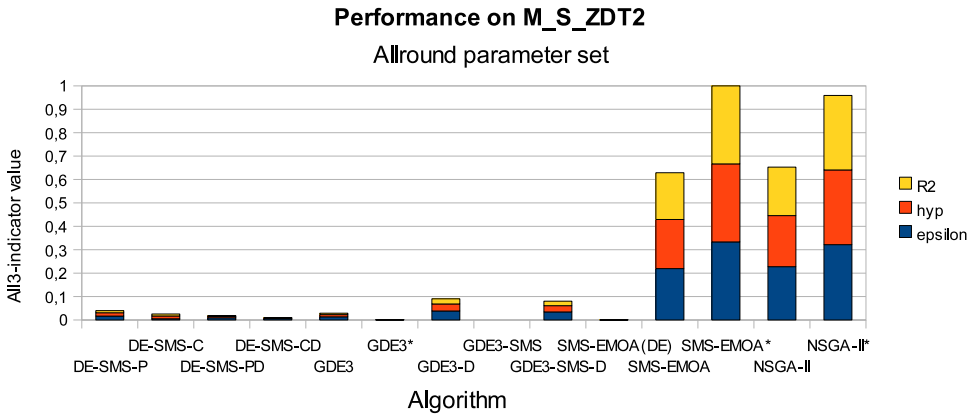


Figure 3.14: All3 indicator on M_S_ZDT2

	Covered sets indicator		
Algorithm	median	mean	std
DE-SMS-P	1	1.12	0.33
DE-SMS-C	1	1.56	0.96
DE-SMS-PD	**6**	**6.16**	0.55
DE-SMS-CD	2	2.48	1.05
GDE3	1	1.00	0.00
GDE3*	1	1.00	0.00
GDE3-D	1	1.00	0.00
GDE3-SMS	1	1.00	0.00
GDE3-SMS-D	1	1.00	0.00
SMS-EMOA(DE)	1	1.00	0.00
SMS-EMOA	1	1.00	0.00
SMS-EMOA*	1	0.96	0.20
NSGA-II	1	1.04	0.20
NSGA-II*	1	1.16	0.37

Table 3.12: Diversity with optimized parameter set on SYM-PART (covered sets indicator)

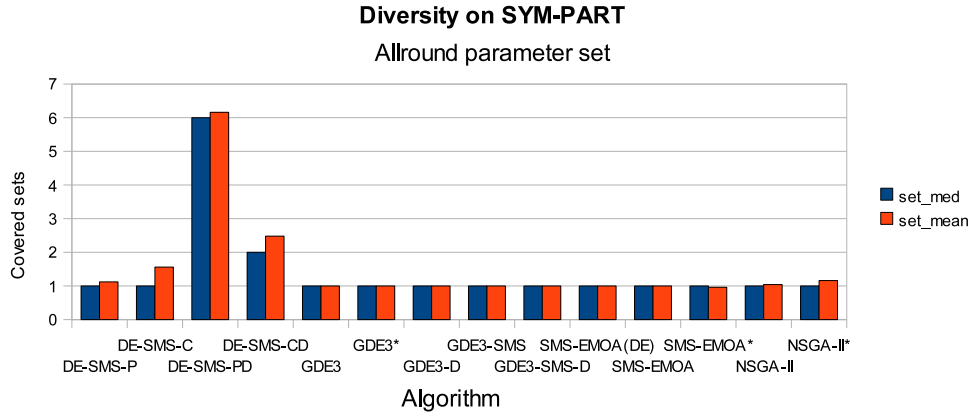


Figure 3.15: Covered sets on SYM-PART

	HVset-indicator								
	M_S_DTLZ2			M_S_ZDT1			M_S_ZDT2		
Algorithm	median	mean	std	median	mean	std	med	mean	std
DE-SMS-P	0.2108	0.2118	0.0437	0.1626	0.1658	0.0512	0.0505	0.0632	0.0307
DE-SMS-C	0.2823	0.2752	0.0437	0.2856	0.2697	0.0400	0.0054	0.0161	0.0259
DE-SMS-PD	0.3977	0.3952	0.0199	**0.6845**	**0.6702**	0.0482	**0.2940**	**0.2926**	0.0237
DE-SMS-CD	0.2584	0.2672	0.0485	0.4497	0.4058	0.1314	0.0609	0.0770	0.0799
GDE3	0.3964	0.3860	0.0903	0.1781	0.1867	0.0677	0.0398	0.0372	0.0326
GDE3*	0.4962	0.4866	0.0299	0.1053	0.1184	0.0572	0.0000	0.0019	0.0096
GDE3-D	0.5444	0.5483	0.0161	0.1051	0.0945	0.0678	0.0187	0.0271	0.0295
GDE3-SMS	0.1129	0.1101	0.0181	0.0523	0.0519	0.0172	0.0000	0.0007	0.0021
GDE3-SMS-D	**0.5795**	**0.5801**	0.0207	0.1456	0.1573	0.0667	0.0194	0.0216	0.0224
SMS-EMOA(DE)	0.2601	0.2418	0.0466	0.2908	0.2808	0.0267	0.0000	0.0000	0.0000
SMS-EMOA	0.1665	0.1730	0.0400	0.0000	0.0009	0.0014	0.0000	0.0000	0.0000
SMS-EMOA*	0.0952	0.0955	0.0286	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
NSGA-II	0.1321	0.1343	0.0331	0.0000	0.0015	0.0027	0.0000	0.0000	0.0000
NSGA-II*	0.0228	0.0334	0.0220	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

Table 3.13: Diversity with optimized parameter set on M_S_DTLZ2, M_S_ZDT1 and M_S_ZDT2 (HVset indicator)

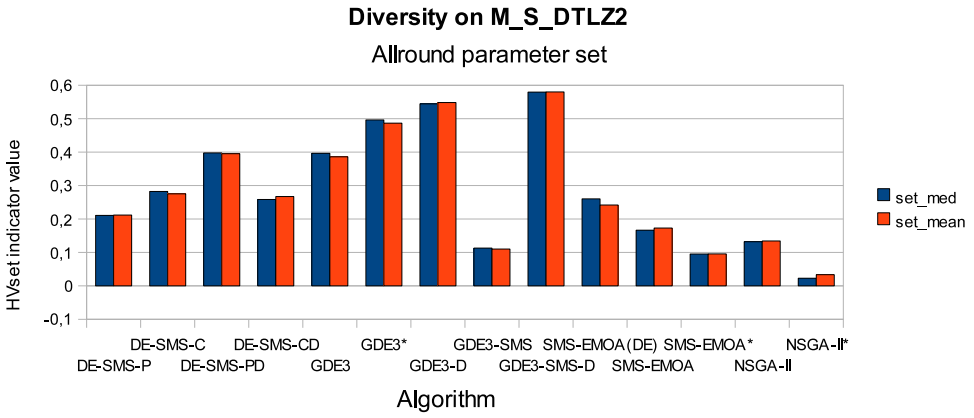


Figure 3.16: HVset indicator on M_S_DTLZ2

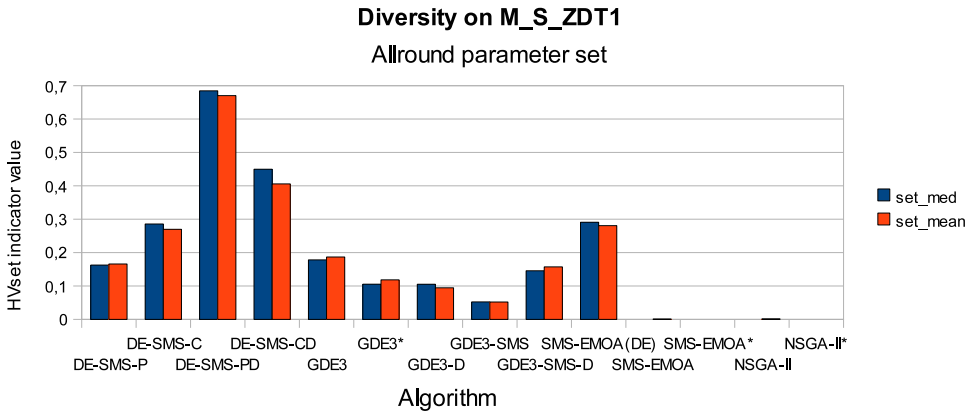


Figure 3.17: HVset indicator on M_S_ZDT1

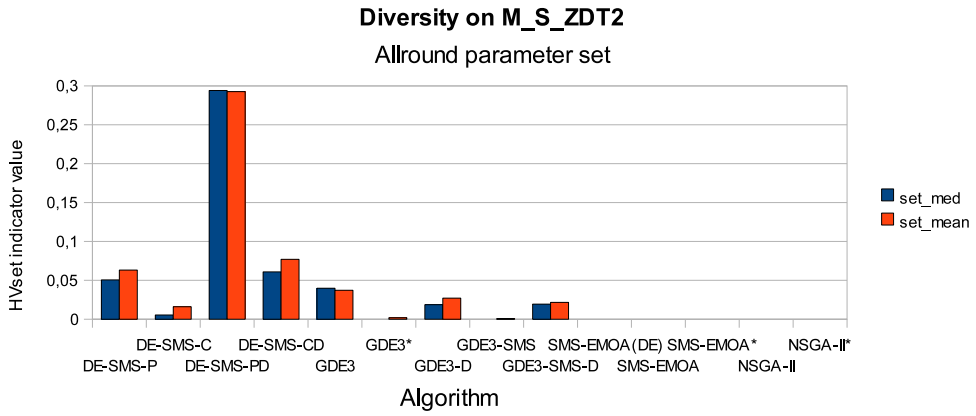


Figure 3.18: HVset indicator on M_S_ZDT2

Due to the vast amount of data generated, some tables and diagrams not displayed in this chapter have been moved to appendix A for further reference. However, since inclusion of all tables e.g. for all parameter combinations, algorithms and test problems is not feasible, the data missing in the appendix is available from the author upon request in an electronic form.

3.3.5 Observations

As far as performance is concerned, there is no algorithm, which, according to the metrics, outperforms all other algorithms on all test problems considered. Some modifications enhance performance, while some do not:

- Cluster-based choice of the parents (local and global search) is only better than random choice if the selection is also modified to enhance diversity. The metric values for DE-SMS-P are worse than those of DE-SMS-PD. GDE3-D and GDE3-SMS-D only significantly improve their counterparts on OKA2 (GDE3-SMS-D also is performing better on M_S_DTLZ2 and M_S_ZDT1). For DE-SMS-C and DE-SMS-CD the difference is not noticeable, though on several test functions DE-SMS-C computes slightly better results, which supports this hypothesis.

- The best algorithms according to the square sum of the All3 indicators are DE-SMS-C and DE-SMS-CD, so the most important improvement seems to be the cluster-based selection (together with the hypervolume contribution).

- Comparing GDE3 and GDE3-SMS is difficult because the parameter combinations dramatically differ, but from the comparison between GDE3-D and GDE3-SMS-D, hypervolume contribution also seems to improve GDE3.

Several interesting patterns can be observed regarding which parameter combinations turns out to be good:

- Algorithms with modified selection to increase diversity all need a small population size (μ approx. 30-45) and a similar step size $F \in [0.4, 0.5]$. The number of clusters C for either clustering during breeding or for modified selection is always the same ($C = 7$) for these algorithms.
- About 60% local search and 40% global search seem to be promising.
- All algorithms directly derived from GDE3 on the other hand perform well with a bigger population size ($\mu \in [100, 160]$) and a smaller parameter $F \in [0.1, 0.2]$.
- GDE3-D and GDE3-SMS-D have the same optimized parameter set.
- Even though for GDE3-SMS a different parameter set than for GDE3 is calculated, it should be noted that the same parameter combination recommended for GDE3 also works good on GDE3-SMS and the other way around.
- With the exception of SMS-EMOA(DE), the crossover probability is always small ($CR \in [0.1, 0.2]$). For DE-SMS-EMOA a high value seems to be necessary ($CR \in [0.5, 0.6]$).
- The optimized parameters for NSGA-II and SMS-EMOA are both the same, in contrast to the recommendations from the authors of those algorithms, the population size is slightly higher and both the crossover probability and the mutation probability are much lower. Also the η_c value is much smaller indicating a larger step size, but with a decreased probability.

As far as the diversity in decision space is concerned, the diversity preserving algorithms seem to work just fine, especially if the enhancement is applied to both breeding and selection:

- DE-SMS-PD is the superior algorithm on SYM-PART, M_S_ZDT1 and M_S_ZDT2.
- DE-SMS-CD performs much better than DE-SMS-C and DE-SMS-PD is also superior to DE-SMS, which indicates that as far as diversity is concerned the combination of both diversity enhancement measures is crucial.
- On M_S_DTLZ2 all algorithms with a population size > 100 attain a very good diversity.

For both additional diversity in decision space and performance in objective function space DE-SMS-CD is a good compromise, because its performance is the second best algorithm, very close to the best according to the metrics used for evaluation. Calculated solutions also significantly improve the amount of diversity in the population.

3.3.6 Discussion

Qualitative evaluation

In this section the reasons for good or bad performance of the modifications are discussed separately from the algorithms that incorporate them on a qualitative level. Just from the metric values no detailed analysis can be conducted, thus a closer look at the behavior of the algorithms involved is necessary. Because of the clear patterns described above, the nature of the enhancements can be immediately transfered to the algorithms that make use of those enhancements.

Breeding based on clustering

Surprisingly the separation into local and global search alone does neither enhance performance nor diversity on R_ZDT4, S_DTLZ3, SYM-PART, M_S_DTLZ2 and M_S_ZDT1. During the approx. 60% local search, the distance of a child from its predecessor is smaller on average than with random choice of the parents. The idea is, that with shorter distance the probability of the child being good enough to be accepted for the next generation increases significantly, so less function evaluations are wasted. In fact this behavior can be observed on all the functions, where the improvement does not work. Two factors are responsible for the increased probability:

1. In all fitness landscapes of the test problems good solutions are not located in a single peak in the landscape but rather surrounded by other good (and possibly better) solutions, otherwise, due to their nature, evolutionary algorithms would be reduced to randomly probing the decision space and could not be used on the problem successfully. If the children are bred in close proximity to their predecessor in decision space, their fitness is also expected to be similar because the transition from good to bad fitness is smooth on most test problems.

2. Differential evolution uses the difference between individuals to find a way to a better optimum than currently reached. The difference can therefore be seen as an estimation where to travel next based upon the previous travel path. If a child is created and its fitness is better, the difference between this child and other individuals can be used to go further into the same direction to hopefully find even better fitness. But the greater the distance, the less probable this concept works. Figuratively speaking, it is like a blind sailor trying to navigate a boat on a river. He knows what direction he comes from and can try to estimate where to go next along the river based upon that information. The information where the search was oriented previously is encoded in the difference between the individuals. The prediction which orientation the river is floating towards is less accurate the further away from his local horizon the navigator tries to estimate the direction he has to steer. Also he might pick

the correct direction, but would run ashore by trying to leave the river in order to continue - leaving the river is analog to bad fitness.

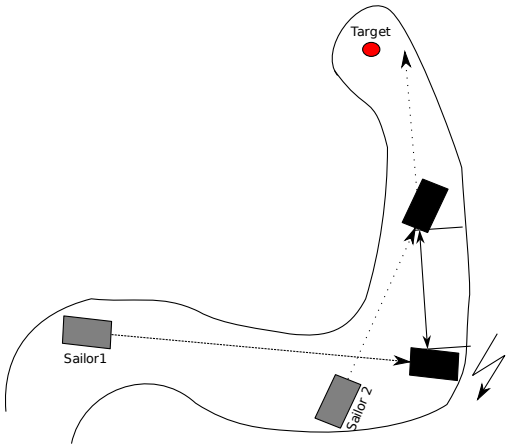


Figure 3.19: Two blind sailors navigating a river

There are also two general major downsides of this measure:

1. Global search is rarely successful, because it probes new parts of the decision space, often areas with bad fitness.

2. Though local search is more often successful, the steps taken are smaller.

Exemplary table 3.14 shows the probability of a child being transfered into the next generation between GDE3 and GDE3-D. For OKA2, GDE3-D performs better than GDE3. On

Problem	**GDE3**	**GDE3-D**	**Improvement in %**
OKA2	4347	6433	47.99
R_ZDT4	4402	8126	84.6
S_DTLZ3	2953	3639	23.23
SYM-PART	6309	7833	24.16
M_S_DTLZ2	8257	8066	-2.31
M_S_ZDT1	10264	10035	-2.23
M_S_ZDT2	10397	10605	2

Table 3.14: Success probability of a child bred in GDE3 and GDE3-D

this test problem most of the population of GDE3 gets caught in the local optima, located on the left and right border of decision space, while only few solutions attain the global optimum. With local search GDE3-D can find other points on the Pareto front (the spiral curve) much easier. On the other hand, despite an 84.6% increased success probability of

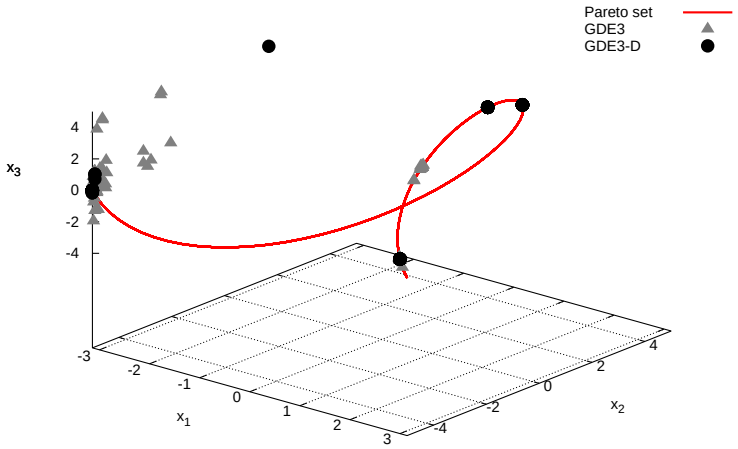


Figure 3.20: GDE3 vs. GDE3-D on OKA2 (decision space)

any child to survive the initial generation it is created in, GDE3-D cannot perform better on R_ZDT4, because R_ZDT4 has so many local optima. While GDE3 can jump from minimum to minimum with larger steps, GDE3-D makes several improvements in each minimum first (by finding better solutions inside the minimum) before actually leaving it. The overall result is, that the increased probability cannot compensate for the downsides,

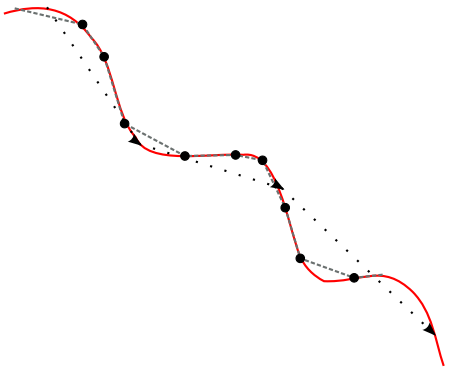


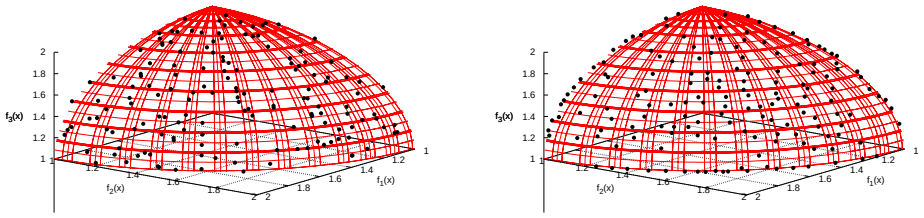
Figure 3.21: Small vs. big steps

the algorithm without clustering can progress faster. Additional runs with higher local search probability indicate better results, so more research regarding this parameter is necessary.

Secondary selection criterion

The difference between the selection criteria crowding distance and hypervolume contribution can be analyzed by comparing GDE3-SMS-D and GDE3-D. Again, GDE3 and GDE3-

SMS are difficult to compare, because the population size differs by approximately factor 6. On OKA2, R_ZDT4, S_DTLZ3 and SYM-PART only a small improvement in the absolute indicator values can be observed while the difference is much bigger on M_S_ZDT1 and M_S_DTLZ2. The reason is simple: for the later group of test problems at least one global optimum is detected, the modified algorithm can fully profit from the change while on the other test functions, the search is stopped in a local optimum. A stronger impact can be seen on the M_S_DTLZ2, due to its three objectives, the Pareto front is a surface which can only be allocated very sparsely by the solutions of both algorithms, the position of each individual becomes more important than in two dimension objective function space, where the Pareto front is a curve for which allocation is possible more densely, given the same amount of individuals. From the graphical point of view, there are bigger holes in the population covering the Pareto front when using the crowding-distance.



(a) GDE3-D (b) GDE3-SMS-D

Figure 3.22: Objective space attained on M_S_DTLZ2

Diversity through selection

Two different types of selection were examined:

1. a child only has to compete against its predecessor (DE-SMS-P and DE-SMS-PD), and

2. a child has to prove itself against either predecessor, or if that fails, the individuals of the cluster it is located in (DE-SMS-C and DE-SMS-CD).

If a child replaces its predecessor, it has to either weakly dominate the predecessor or at least have a higher hypervolume contribution otherwise it is deleted even if some or even all other individuals in the population are worse according to the same criteria. As a result, the probability of a child being accepted is much lower than without this modification, thus decreasing the speed of convergence of the entire algorithm. No positive impact on

diversity can be observed via the covered set or HVset indicators. The quality of the solutions computed by DE-SMS-P in most regions is too low to be accepted by the HVset and covered sets indicators, only one set is counted covered and the HVset value on the other diversity test problems is worse than the one computed for GDE3.

As far as the quality of solutions is concerned, the population size μ is only set to 28, per individual in the population more children are bred and more improvements per individual are possible. On OKA2 and the multimodal function R_ZDT4 and S_DTLZ3, the success probability of a child in DE-SMS-P is less than a third of the corresponding value from GDE3, therefore many potentially better children are deleted and the performance decreases, e.g. on OKA2 and R_ZDT4 less than 5% of the function evaluations DE-SMS-P are actually used to improve the population, more than 95% are wasted because the child is immediately dropped. In many cases child and predecessor are not in close proximity to each other - if the child replaces the predecessor, it does no longer represent the area its predecessor did and thus diversity can be lost if several of those areas are merged. As we will see later, this behavior turns out to be a desirable feature of DE-SMS-C(D).

We have established above no significant performance improvement and no increased diversity could be achieved by the modified breeding phase alone. With an additionally modified selection phase however this no longer holds. Compared to DE-SMS-P, the

Problem	**DE-SMS-P**	**DE-SMS-PD**	**Improvement in %**
OKA2	791	1667	110.75
R_ZDT4	867	1409	62.51
S_DTLZ3	1286	1602	24.57
SYM-PART	1830	3778	106.45
M_S_DTLZ2	4323	4380	1.32
M_S_ZDT1	4495	4461	-0.76
M_S_ZDT2	4936	5344	8.27

Table 3.15: Success probability of a child bred in DE-SMS-P and DE-SMS-PD

counterpart with modified breeding phase DE-SMS-PD is much better regarding both performance and diversity. There are two reasons as to both improvements work well together:

1. the success probability of a child improves significantly on the test problems where DE-SMS-P performs badly and

2. the created children are in closer proximity to their predecessor, which helps preserving diversity.

The first feature increases the quality of the solutions, while the second improves diversity in decision space, both together result in diverse solutions of good quality, counteracting

the two downsides of additional diversity in selection phase. DE-SMS-PD is the algorithm that by far outperforms all other tested algorithms on SYM-PART(30D) with 6 covered sets. Similar behavior can be observed on M_S_ZDT1 and M_S_ZDT2 regarding the HVset value. On M_S_ZDT1 the HVset indicator value is almost 3 times (284%) as big for DE-SMS-PD as for GDE3 and more then 3 times (321%) the value of DE-SMS-P. On M_S_ZDT2 the difference is even bigger. An exception is M_S_DTLZ2, where all other algorithms reach a much higher HVset indicator value. This test function has a design flaw: almost the entire objective function space of a box is part of a global minimum, hence the minima are almost directly attached to each other like an egg carton. On such a function the most important factor to a good HVset value is a big population size because the more individuals there are, the more space can be covered.

An interesting fact is also that, according to the All3 indicator, DE-SMS-PD outperforms GDE3 regarding quality of the solutions on all test problems but S_DTLZ3, even though it significantly offers a more diverse set of solutions, the quality of these solutions is better. At first sight, this seems contradicting, because preserving diversity was only expected to be available at the cost of additional function evaluations. A more detailed look at the properties of DE-SMS-PD discloses the reason. To properly work, any successful evolutionary algorithm needs a to some degree diverse population, because otherwise the variation operators cannot exploit structural properties of the fitness landscape by combining individuals. DE-SMS-PD keeps the diversity up with additional measures, the number of individuals required in the population to maintain this minimum amount of diversity is significantly lower, thus the algorithm can further optimize each individual in the population more often. In this case the population of DE-SMS-PD is only 45, statistically each individual is enhanced more than twice as much as in GDE3, where μ is ≈ 100. To compensate and progress faster in the beginning with a bigger population, many MOEA (i.e. NSGA-II) use a binary tournament selection [57] to neglect parents with bad fitness in the breeding process, however GDE3 does not make use of such a feature.

Parameter	**GDE3**	**DE-SMS-PD**
μ	101	45
F	0.18/0.5	0.49
CR	0.18/0.1	0.14
L	-	0.61

Table 3.16: Comparison of parameter values: GDE3 vs DE-SMS-PD

Another important point is, that DE-SMS-PD makes smaller steps in a local area around the predecessor, but, due to the 1.5 times bigger F value, each step is weighted more. The effect resulting from this behavior is, that search is still local, which is good to

preserve diversity, but with bigger steps than GDE3-D and GDE3-SMS-D employ. This behavior can be compared to a music company contracting a lot of musicians. If from many of those contracts no or only few money can be harnessed because customers do not buy many of their CDs, those musicians that actually have success need to bring the company a lot of money, in order to keep the companies balance sheet up.

DE-SMS-CD and DE-SMS-C show a similar behavior like DE-SMS-PD (because the of the common concept). There are however some differences to be discussed. One big disadvantage of DE-SMS-PD is still, that too many children are dropped prematurely. To reduce this effect towards better performance, while still preserving diversity, those children are given a second chance to prove their fitness in a contest against all individuals located in the same cluster. The amount of diversity in the entire population is reduced by this measure, while performance regarding the quality of the solutions increases. The

Problem	**DE-SMS-PD**	**DE-SMS-CD**	**Improvement in %**
OKA2	1667	11362	581.58
R_ZDT4	1409	5853	315.4
S_DTLZ3	1602	4201	162.23
SYM-PART	3778	5031	33.17
M_S_DTLZ2	4380	6474	47.81
M_S_ZDT1	4461	6838	53.28
M_S_ZDT2	5344	11638	117.78

Table 3.17: Success probability of a child bred in DE-SMS-P and DE-SMS-PD

reasons for improvement over DE-SMS-PD are obvious: the success probability of a child is increased. The steady-state approach also introduced with this algorithm further improves the performance (see section 3.1.4). In DE-SMS-P(D) the predecessor represents a region in decision space to be searched. Of course these regions are not all entirely separated but some regions can merge into a bigger region. This region is now extended to the entire cluster, the amount of regions available for search is therefore decreased to the number of clusters, the number of individuals in that region however increases to the number of individuals in the cluster. The positive effect is that now via parameter C the amount of regions can be configured towards the users needs.

Typically DE-SMS-C(D) forms few big clusters in decision space with many individuals in it, in these clusters most of the performance improvements occur. The individuals in smaller clusters, which often contain dominated solutions, are kept to enhance diversity and explore other areas in decision space. While those dominated solutions do not improve the fitness of the set of solutions, they can serve as parents for global search if one of the bigger clusters gets stuck in a local optimum. If the cluster contains at least two individuals, even

local improvements via local search is possible. During the optimization process individuals in a small cluster might not improve further for a long time. Small clusters that contain only one individual can be easily dissolved or migrated to another area in decision space with better fitness if the individual is chosen as a predecessor and the fitness of the child has a better fitness. If the child is located somewhere entirely different, the cluster is either moved or can even be merged with another cluster in close proximity. This way clusters containing individuals with bad fitness are slowly, but automatically either migrated and the individuals in it improved or the cluster is decreased by one individual at a time as the optimization progresses and areas with better fitness are explored. This feature is very important, because it prevents the algorithm from keeping lots of individuals with bad fitness and thereby decreasing the flexibility of the algorithm.

The main difference in methodology between DE-SMS-CD and DE-SMS-C is, that without local search, the algorithm tends to form clusters closer to one another, hence the lower div+ indicator value. The overall area searched is decreased, so is diversity of the population, but on the other hand, faster progress is possible because of the bigger steps. Results are therefore slightly better. This behavior can be understood by looking at the div+ indicator curves (see figure 3.24). For all test problems, the div+ indicator value of DE-SMS-C is smaller than the value of DE-SMS-CD.

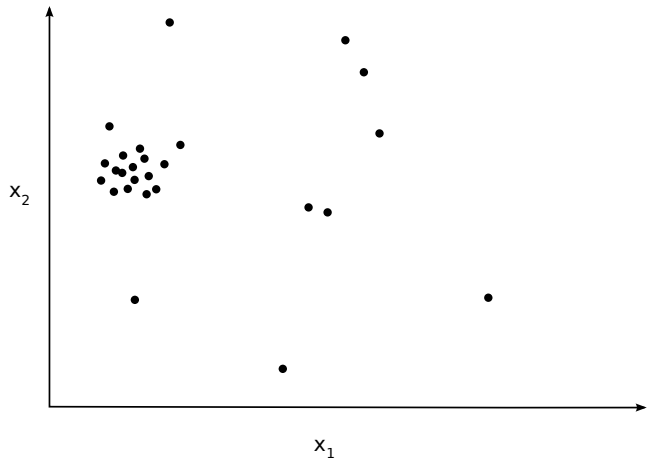
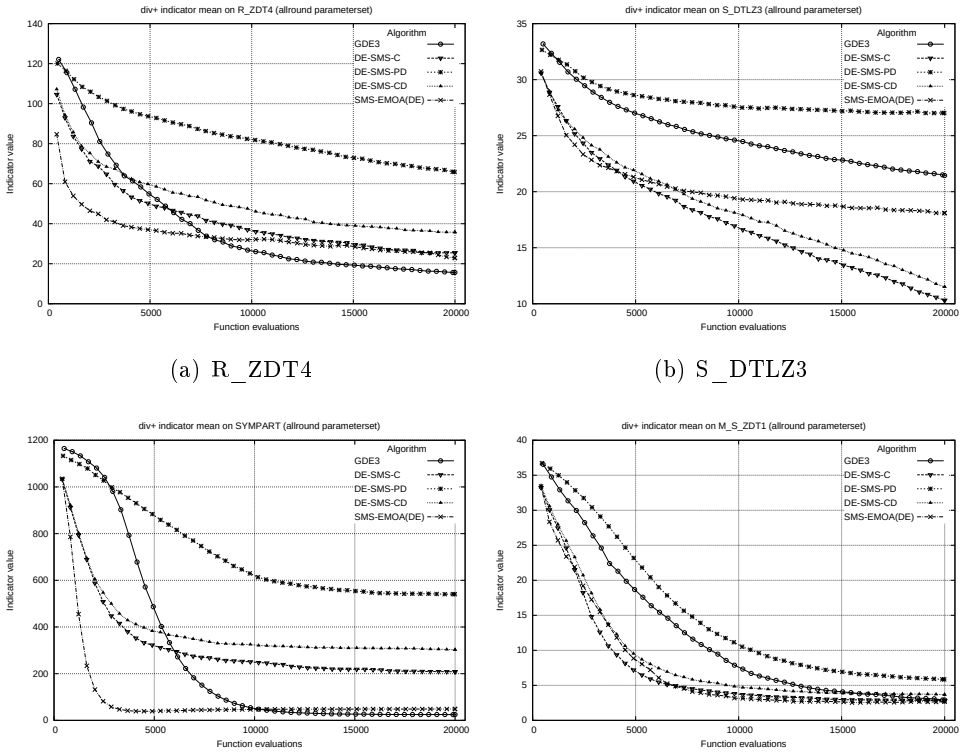


Figure 3.23: Search pattern of DE-SMS-C(D) in decision space

Development of diversity

The development of the div+ indicator values over the number of function evaluations is only useful under some conditions. Among the algorithms tested DE-SMS-PD achieves the greatest diversity in decision space from the beginning, the gradient of the curve is the lowest. The div+ indicator seems to indicate, GDE3 is able to preserve the diversity longer than DE-SMS-C(D) on S_DTLZ3 and M_S_ZDT1. This is due to the significantly

better performance of DE-SMS-C(D) on these test problems. DE-SMS-CD is figuratively speaking multiple steps ahead as far as the quality of the solutions is concerned. When GDE3 reaches the same quality the div+ indicator value is actually lower. The same argument holds for SMS-EMOA(DE) on S_DTLZ3.



(a) R_ZDT4

(b) S_DTLZ3

(c) SYM-PART

(d) M_S_ZDT1

Figure 3.24: Development of the div+ indicator value over function evaluations

SMS-EMOA(DE)

Using differential evolution from GDE3 on SMS-EMOA also reveals some interesting results.

- SBX and PM seem to create only very small improvements with each generation.
- Because the DE variation operator is invariant against the rotation of the search coordinate system, it also performs much better on rotated functions like SYM-PART and R_ZDT4.
- Parameter $CR = 0.56$ is in contrast to all other algorithms 2-3 times as big. Such a big value is necessary because given the small population size, the diversity of the population is very low and the algorithm therefore has to create more diverse individuals at any time to prevent it from getting stuck in a local optimum.

In order to determine a ranking among the algorithms the sum of squares of the six All3 indicator values can be used the same way as to find a good allround parameter set for each algorithm. As can be seen from the sum of squares of the All3 Indicator, there are

Algorithm	Sum_{All3}	Sum^2_{All3}	CS_{mean}	Sum_{HVset}	Sum^2_{HVset}	Rank
DE-SMS-C	0.1390	0.0074	1.56	0.5733	0.1613	1
DE-SMS-CD	0.1910	0.0125	2.48	0.7690	0.2727	2
DE-SMS-D	1.0688	0.3871	6.16	1.3763	0.7132	3
GDE3-SMS-D	1.2183	0.4200	1	0.7445	0.3574	4
GDE3-D	1.6256	0.5599	1	0.6682	0.3078	5
SMSEMOA(DE)	1.3196	0.7827	1	0.5510	0.1523	6
GDE3	1.5186	0.8047	1	0.6144	0.1905	7
DE-SMS-P	1.7439	0.9378	1.12	0.4239	0.0734	8
GDE3-SMS	2.0145	1.0635	1	0.1652	0.0155	9
GDE3*	1.9985	1.1632	1	0.6015	0.2573	10
NSGA-II	3.8131	2.5551	1.04	0.1321	0.0174	11
SMS-EMOA	3.8558	2.6188	1	0.1665	0.0277	12
SMS-EMOA*	5.8273	5.3177	0.96	0.0952	0.0091	13
NSGA-II*	6.3407	5.8231	1.16	0.0228	0.0005	14

Table 3.18: Ranking among the algorithms based upon the sum of squares of the All3 indicator values

some contradictions to statements made above. The most notable is, that GDE3-D has a better rank than GDE3. This is only because for GDE3 the indicator value on OKA2 is much worse than for GDE3-D. The sum of squares penalizes bigger values, hence GDE3-D seems better according to the sum of squares. The diagram above also shows which improvements are most important. Regarding the performance, algorithms can be roughly separated into three groups:

1. DE-SMS-C and DE-SMS-CD
2. All other algorithms using differential evolution
3. NSGA-II and SMS-EMOA

In each group, the performance is similar, while the difference in performance among the groups is significant. From the difference between the algorithms in the groups, the major improvements to the algorithm can immediately be identified.

1. Differential evolution is an improvement over SBX and PM.
2. Additionally selection based on clustering significantly improves overall performance.

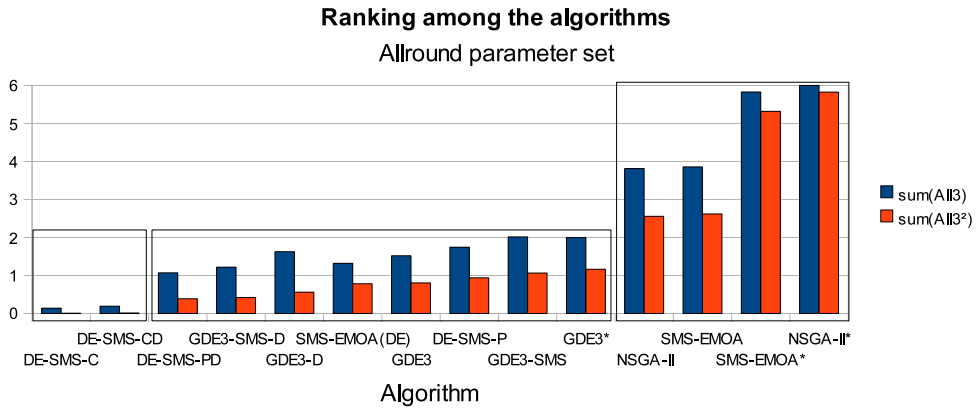


Figure 3.25: Ranking among the algorithms

As far as diversity is concerned, the rank can also be divided into two groups, one group which members significantly preserve diversity (DE-SMS-C, DE-SMS-CD, DE-SMS-PD) on the one side and all remaining algorithms on the other. A good compromise between performance and diversity should therefore be chosen from the first group. The best compromise between performance and diversity is in the authors opinion DE-SMS-CD, because compared to GDE3 and SMS-EMOA it offers significantly more diversity in decision space and better performance according to the metrics. If more diversity is required, DE-SMS-PD is the alternative to be considered, if the focus lies on performance alone, DE-SMS-C is a good option.

Chapter 4

Application

In this chapter some of the algorithms evaluated in the previous chapter are applied to surface reconstruction to see which of the algorithms performs best regarding both diversity and quality of the solutions.

4.1 Pre-experimental planning

Due to time constraints, only some algorithms can be investigated. Therefore, the most promising algorithms are chosen to measure quality and diversity of the solutions and some algorithms are examined for reference and comparison to previous results. DE-SMS-CD is used because regarding performance it is almost as good as the best DE-SMS-C in the previous experiments, but significantly performs better as far as diversity in decision space is concerned. The algorithm performing best concerning diversity, DE-SMS-PD, is also included into the test set. GDE3 served as the basic template for all algorithms, NSGA-II is a commonly established and well-known algorithm often used for comparisons in the area of multi-objective optimization and SMS-EMOA was previously used for surface reconstruction, hence these three algorithms are investigated as well to see if any improvement can be achieved from the application of DE-SMS-CD and DE-SMS-PD.

4.1.1 Problem-specific optimizations

Even though standard genetic algorithms are black-box optimizers that do not make use of any special knowledge on the task they perform, their performance can be improved by incorporating knowledge on the structure of the problem into the variation operators. Also the process of generating the initial population is crucial, since the path towards a good solution can be much shorter if the algorithm can already start half-way towards the optimum, saving many function evaluations for the second half (assuming the algorithm can finish the second half equally fast). Clearly, diversity is also an issue. Entirely random distribution in decision space will demand more function evaluations, with too few diversity

the algorithm is likely going to get stuck in a local optimum prematurely and the solutions might not meet the user's quality constraints. If possible, both a good variation operator and a good set of initial individuals are desirable if a genetic algorithm is to be employed on a practical application.

Variation operator

As already stated above, the design of a suitable variation operator depends on knowledge about the structure of the problem. Typically, genetic algorithms are applied to problems that are not fully understood (yet), but even in those cases there is information available, e.g., on how the problem description is encoded into the individual's allele or what data structure is used to represent the solution. For surface reconstruction an individual represents a surface, which is described by the net of control points of a NURBS surface [1] (see chapter 2.5). While little is known on the process of surface reconstruction, enough information is available on NURBS and their properties. The most prominent reason NURBS are often chosen as representation of the surface consists in local support. Local support means that the modification of one control point only affects a limited area near the control point, while the rest of the NURBS is entirely unaffected by the change. A genetic algorithm can thus vary parts of the surface separately and an improvement in the affected area will not result in a deterioration somewhere else on the surface.

Standard variation operators

Both the differential evolution used in GDE3 and simulated binary crossover are unsuitable for surface reconstruction. Differential evolution, which only picks entries in the allele at random, creates many, small and locally different changes. This is not a good idea, because an improvement in one area might be counteracted by a worsening somewhere else, leading to rejection of the child. The algorithm does not make use of the previously discussed local support feature of the surface representation. Since the variation is comprised of calculating the difference between two corresponding control points of two different surfaces, the result is not as bad as it may seem, but there is obviously room for improvement. Polynomial mutation (PM) introduces a major problem into the process of surface reconstruction, because it causes too large, punctual and completely random changes. Mutated surfaces may indeed have a good fitness value with the objectives used, but also show other properties which disqualify them for practical use. A typical shape disorder are "nibbled edges" resulting from a control point at the border of the control net being moved outside the original object. Such a surface looks as if someone had nibbled off parts of its border.

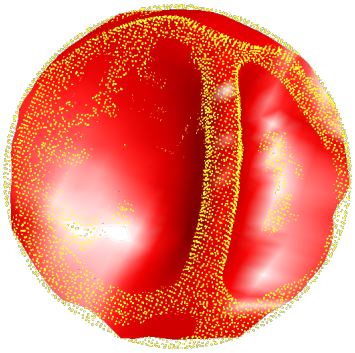


Figure 4.1: Nibbled edges

This can happen with differential evolution as well, but the effect is much less dramatic there, as the distance a border control point is moved is a lot smaller and not entirely random, because it happens to be moved only for the difference between the position of the corresponding control point of two (other) surfaces. The opposite can of course also happen, the resulting edges can appear like a saw. Even with small mutation probabilities p_m, as recommended for the NSGA-II and SMS-EMOA, the calculated surfaces often contain at least either disorder. So if polynomial mutation is to be used at all, border control points should be excluded from mutation. As an alternative an additional objective could be employed to penalize moving border control points too far away from the scan points.

As value for the modified dimensions of the children, simulated binary crossover (SBX [58]) basically uses the weighted mean of the corresponding values from the parents. In decision space, the children are always located somewhere in between the parents, depending on the distribution of the initial population, using the SBX operator alone can therefore limit the search space (see below). In NSGA-II and SMS-EMOA polynomial mutation is introduced to counteract this problem, with the described side-effects for surface reconstruction. Polynomial mutation constantly increases the diversity of the population if the mutated children are selected. Simulated binary crossover cannot counteract the side-effects from polynomial mutation, because the parameter combinations indicate, PM affects a few ($p_m \leq \frac{1}{D}$), random dimensions in a dramatic way, while SBX replaces values in a lot of dimensions at once ($p_c \approx 1$). The standard variation operators are only employed to show the improvement of differential evolution, thus both mutation and binary crossover are left unchanged; none of the ideas proposed for their enhancement or different parameter combinations are tested.

Differential evolution for NURBS surfaces

A common approach to repairing your broken computer at home is to find out what is wrong and then fix it. Fixing hardware is often difficult, especially if you do not know anything about computers, but identifying which hardware component is damaged is simple, by

trial and error - replace all components, one at a time and see if it works again. A very naive approach to fix two broken computers without special knowledge is to randomly switch two corresponding hardware components between the two PCs and check if one of them works again. Given that each essential component works in at least one of the computers, eventually one of them will be functioning. Delegated to surface reconstruction: the basic idea for a better variation operator for NURBS surfaces is to choose one local area at random for each act of recombination and apply the operator only there. From now on this area will be called recombination area. This area needs to fulfill the following requirements:

- variable size, to allow for recombining small and big areas,
- border of the area is defined by control points, since the only thing we modify are control points and
- simple shape.

A square shape fulfills all the requirements and is chosen because it is easy to understand and implement. The recombination area is determined as a sequence of control points, which are covered by that area:

1. Choose one control point at random. This control point will be designated the center of the recombination area and the point of origin of the sequence.
2. Go from control point to control point in a spiral around the center, at each position draw a random number $w_{rand} \in [0, 1]$, stop iff
   (a) $w_{rand} < CR$ or
   (b) all control points from the surface are in the sequence.

   If the sequence reaches the border of the control net and would normally leave the control net, it simply continues on virtual helper control points until it reaches the next control point that really exists and only then decides whether to continue or not by drawing another w_{rand}.

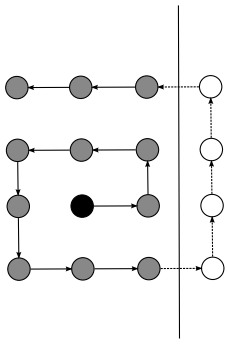


Figure 4.2: Sequence of control points

Once the recombination area is determined, differential evolution can be employed on all the control points in that area. This variation operator is called differential evolution for NURBS surfaces, short: DE_NURBS.

Please note, that while it is possible to do a binary crossover only in the recombination area, this is not pursued any further as this work is focused on differential evolution and such a variation operator would still suffer from the issues appointed. Among the features of DE_NURBS are:

- local support,
- great variety of recombination areas and
- less dimensions in decision space.

DE_NURBS can also be seen as classic differential evolution [13], but instead of going from index to index in a linear manner, it simply uses a spiral walk among the control points.

Initial Population

Because the number of function evaluations is limited to 20000, function evaluations are a precious resource. Thus, an effective way to create an initial population, that already offers acceptable fitness and is as diverse as necessary to allow the evolutionary algorithm further optimization, is needed.

Interpolation

Weinert et al. [59] propose a linear interpolation between a pre-optimized surface $surf_{pre}$ with good fitness regarding proximity to the control points and a flat surface $surf_{flat}$ to create intermediate solutions $surf_i$ to fill the gaps in the Pareto front. Interpolation is a good means to get intermediate solutions as this way a lot of surfaces can be created very fast and yet recombination between two surfaces can result in better children (e.g. because

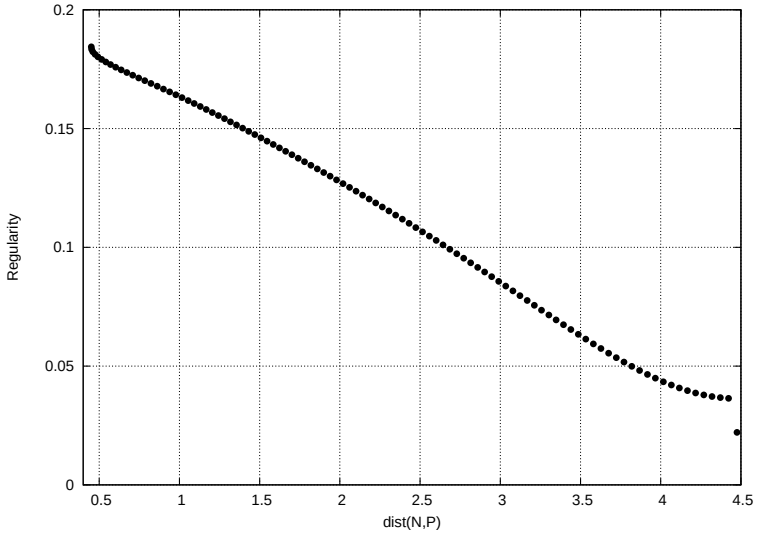


Figure 4.3: Initial fitness values through interpolation [59]

Algorithm 12 Interpolation between two surfaces

1: $P_0[0] \leftarrow surf_{pre}$ // first individual
2: $P_0[\mu - 1] \leftarrow surf_{flat}$ // last individual
3: **for** $i = 1$ to $\mu - 2$ **do** // all interpolated individuals
4: $n = \frac{i}{\mu - 2}$ // weight factor
5: $surf_i \leftarrow surf_{pre} + n \cdot (surf_{flat} - surf_{pre})$ // component-wise interpolation
6: $P_0[i] \leftarrow surf_i$
7: **end for**

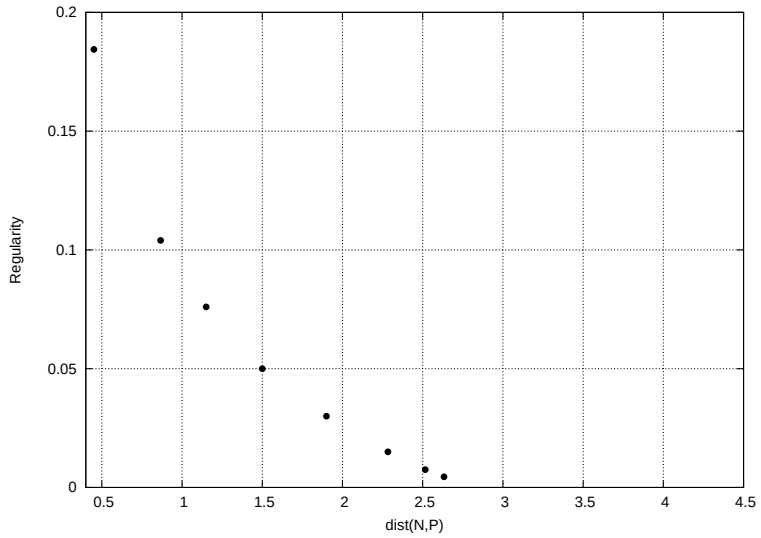
one parents contribution smooths parts of the surface that are not flat enough to get a good fitness value). While the approach offers a great deal of diversity to begin with, the fitness of the initial population could still be improved and the design of another surface is required. Even with the same approach, it is possible to get slightly better initial results by minimizing the average distance of the points to $surf_{flat}$, thus finding an optimal initial position for the second surface for interpolation. Interpolation is a simple means to get solutions covering the entire spectrum from proximity to the scan points to smoothness of the surface, however in most cases only parts of the Pareto front generated this way contain usable surfaces, but the basic idea to interpolate between two surfaces seems a good means to generate additional initial individuals because no additional function evaluations are required during the generation.

Filter

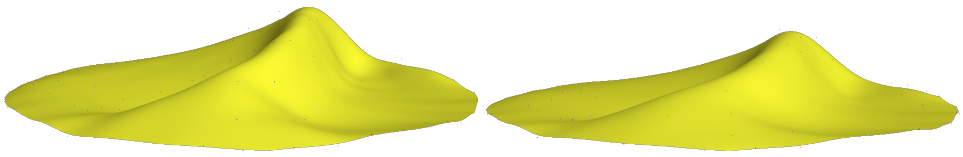
$surf_{pre}$ is expected to have a good fitness value regarding the proximity to the scan points, what it lacks is smoothness. Instead of using an additional surface for interpolation, one

can therefore smooth the pre-optimized surface with a simple spatial filter e.g. the well known mean filter from digital image processing [60]. The mean filter uses an operator window that is moved through an image from the beginning to the end pixel by pixel, calculating the mean value of all the pixels in the operator window and assigning this mean value to the pixel it is calculated for. The filter reduces the difference in intensity between pixels and their neighbors in the image and makes the picture look smoother in the process.

Instead of reducing the difference in intensity, the same filter will smoothen the control net, that is reducing the distance variation of the control points and their neighbors in each direction. The result is a smoother surface than the pre-optimized one. If necessary this filter can be applied multiple times in a row in order to get an even smoother second surface. Various operator windows are common, but as the whole point of using NURBS

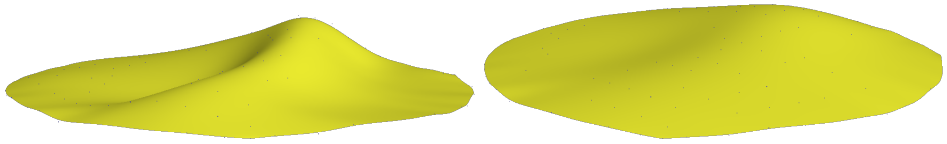


(a) Fitness values for the iterations 0,1,2,4,8,16,32 and 64



(b) Surface after iteration 1

(c) Surface after iteration 2



(d) Surface after iteration 4

(e) Surface after iterations 16

Figure 4.4: Smoothening iterations

is to have only a small number of (control) points to save, e.g. a 16x16 net, as used in the experiments later on, a small operator window is preferable. A bigger operator window will in most cases smooth the net too fast and that is often not desirable for reasons discussed below. The smallest useful operator window is composed of the control point and its left, right, upper and lower neighbor (if available).

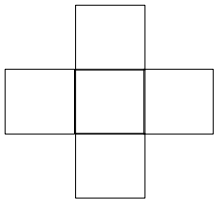


Figure 4.5: Operator window for the mean filter

Complications

When applying the filter some things need to be taken care of, in order to not alienate the surface too far. Too much difference will load a lot of additional work on the evolutionary algorithm, and since that work is expensive, the less left to do the better. The mean filter must not be applied to the border points of the control net. The most important reason is, that these points lack one neighbor, corner points in the control net even lack two neighbors. This is a problem, because a border point will move either into or out of the current pre-optimized surface when the mean value is calculated, shrinking or extending the surface in the process. This effect is cumulative, the more often the filter is applied, the smaller (bigger) the surface gets. Because of the reason above, the border points will stay the same for the pre-optimized surface, the second smoothed surface and all surfaces interpolated in between. Differential evolution is then not be able to change any of the border points, for if the values for all parents are the same, the difference between any of the parents will always be zero. To avoid this the border points need to be modified another way after the smoothing process is complete.

In order to add diversity and still preserve the basic shape of the surfaces one can move each border point of one surface a portion of the distance into the direction it would have been moved during the last smoothing iteration and move the border point of the other surface the same distance into the opposite direction. Interpolation between the surface will then result in values in between. It is obvious that this will most likely decrease the fitness of the pre-optimized surface, so the portion needs to be small while still adding enough diversity to the border points. The more border points there are, the smaller the distance can be since the created surfaces differ at more points. Initial experiments show that 1/16 seems to be a decent value for a 16x16 control net, smaller control nets will probably need a bigger value. Please note, that there is no need to actually decrease the

performance of the pre-optimized surface and loose fitness in the process, but rather add a copy of the surface and interpolate between that copy and the smoothed surface.

Advantages

The comparison between the two types of interpolation shows that the use of smoothing results in a far superior initial population than interpolation alone does achieve on the considered surface. Also the number of iterations chosen has an impact on the quality of the initial population. The more iterations, the worse the initial population is, but on the other hand the more iterations employed, the smoother the second surface becomes and the more of the Pareto front will be covered. Even with smoothing and interpolation a better Pareto front can be constructed by interpolating between surfaces $surf_i$ and $surf_{i+1}$. This can be done without a single function evaluation, as long as none of those surfaces are evaluated. Since the mean filter only calculates the value for each of the D control points once based on a constant operator window, a smoothing iteration can be calculated in $\mathcal{O}(D)$.

4.1.2 Recommended approach

The creation of the initial population as described in the section above can be tuned to suit the users needs. Therefore, the following systematic approach is suggested:

1. Run deterministic algorithm to optimize the proximity of the NURBS surface to the control points.

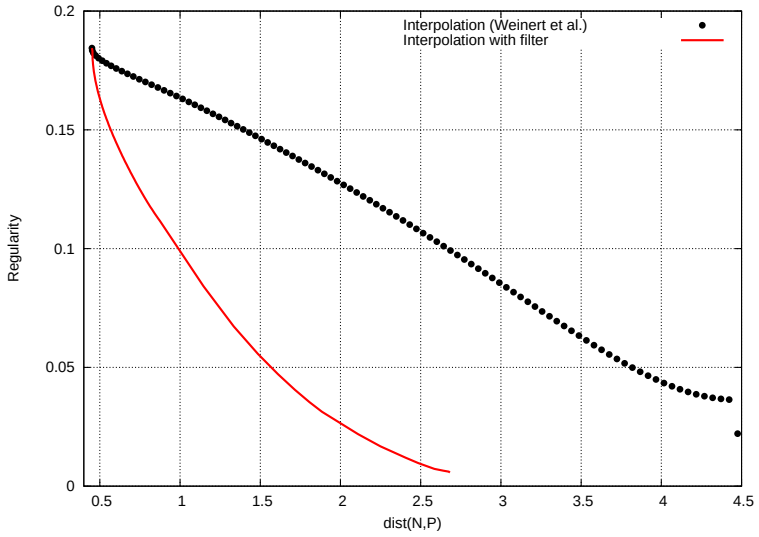


Figure 4.6: Fitness values attainable by the filter approach

2. Run smoothing for n iterations with $n \in \{1, 2, 3, ...\}$ without evaluating these surfaces. Usually no more than 16 iterations should be necessary. This requires no function evaluations.

3. Look at the surfaces created by interpolation. Find the region you wish to investigate further by choosing two surfaces the following way:

   (a) Choose as first surface the one that seems good enough with respect to the first objective and still looks good enough to be used as a final solution. Usually this will be the pre-optimized surface.

   (b) Choose the second surface by looking at the smoothness of the candidates. Take the one that in your opinion is smooth enough but still offers a good fitness value for the second objective and would be a candidate if it were a little improved in some areas. In most cases this will be the one that was smoothed once. If you do not know which surface to choose, use the surface created by the first iteration.

   The space between the two surfaces chosen is the users region of interest and the task of the evolutionary algorithm will be to search for good solutions in that area. If this region is significantly limited by the user, the entire function evaluations can be concentrated to improve the solutions in that area, so no function evaluations will be wasted by calculating solutions in a region that will be entirely dismissed anyway and more useful results will be offered when the algorithm has finished the optimization.

4. Do linear interpolation between the two surfaces to fill the population. This step is available at no additional cost since a population needs to be generated anyway. If one of the surfaces already is good enough for the user, there is no need to continue.

5. Run evolutionary algorithm for 20000 function evaluations.

6. One can also try to interpolate solutions between the generated solutions to get a wider variety of solutions (optional). This however will only work, if the surfaces chosen for interpolation reside in the same local optimum.

7. Choose the best solution from the Pareto front.

4.2 Experiments

4.2.1 Task

The task of the following experiments is to find out which algorithm is best suited for surface reconstruction and to check whether the rank established from the experiments on the test problems is preserved. If the test problems were representative for surface reconstruction, then the rank among the algorithms should be the same. As secondary objective is to determine, whether the new variation operator DE_NURBS outperforms the differential evolution variant from GDE3 and to validate.

4.2.2 Setup

	Parameters				
Algorithm	μ	F	CR	L	C
DE-SMS-PD	45	0.49	0.14 (0.9)	0.61	7
DE-SMS-CD	45	0.49	0.14 (0.9)	0.61	7
GDE3	101	0.18	0.18 (0.9)	-	-
	μ	p_m	p_c	η_m	η_c
SMS-EMOA	157	-1.33	0.7	21	5
SMS-EMOA*	100	-1	1	20	15
NSGA-II*	100	-1	1	20	15

Table 4.1: Parameter combinations for surface reconstruction

Common setup

Each algorithm is run for 20000 function evaluations 25 times with the same parameter combination. Like in chapter 3.3.3 for every run of an algorithm the ε-, hypervolume-R2-indicator are calculated. From the median of these three values the All3 indicator is computed for easier comparison. These three indicators require a reference set for computing the results. Additionally the diversity is measured via the div+ indicator. As the div+ indicator measures diversity aside any quality concerns, the diversity is determined from the entire population, even if some individuals are not located in the region solutions are accepted from. To see whether the results are statistically relevant, a Kruskal-Wallis test is performed.

Initial population

For all algorithms, the initial population is interpolated between the pre-optimized surface $surf_{pre}$ and the first filtering iteration $surf_{f1}$, because on the one hand, the quality of $surf_{f1}$ is not very good regarding the proximity to the scan points and every later iteration would be even worse. On the other hand, compared to the other methods, this initialization already creates better results according to metrics without a single function evaluation. To analyze the improvement, 25 runs of SMS-EMOA with the default parameter combination are run with a randomly initialized population and additional 25 runs with the interpolation alone as proposed by Weinert et al. [59].

Variation operator

DE-SMS-PD, DE-SMS-CD and GDE3 are run with both the standard variation operator from GDE3 and DE_NURBS separately to examine the impact of the new variation operator. The parameter CR requires a different value for DE_NURBS because the dimensions are picked in an entirely different way than in the ordinary GDE3. Since $CR = 0.9$ is often recommended for differential evolution [13] and DE_NURBS and differential evolution proposed by Storn and Price create the sequence of dimensions to mutate in a similar way (the stopping criterion is the same), a similar value can also be expected to work for DE_NURBS, especially because the number of control points is high. As 25 runs of each algorithm take approximately 24 hours, no optimization of this parameter is feasible.

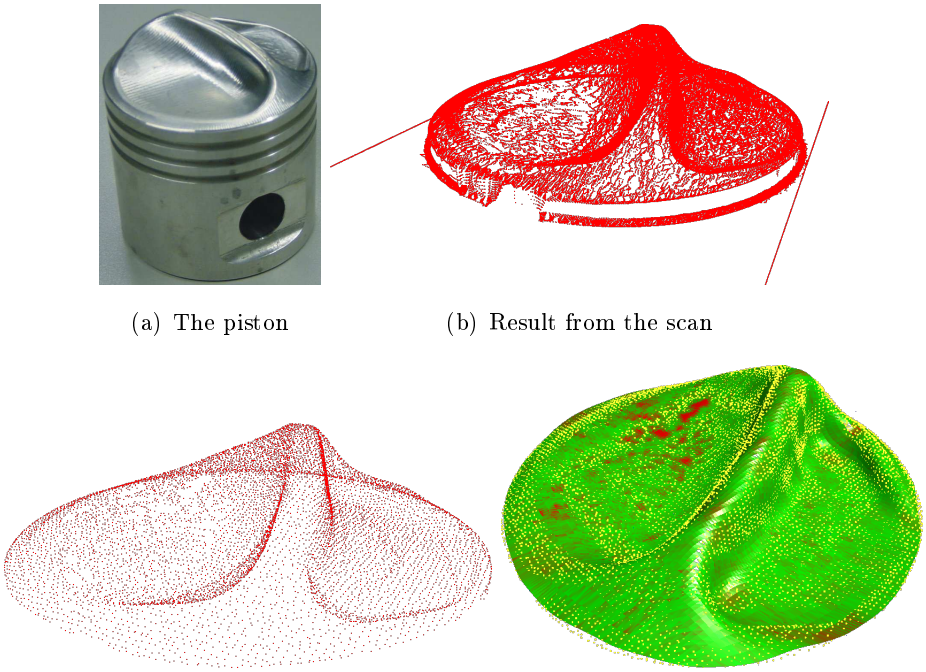
The surface

The scan of the top of a piston is reconstructed in the experiments. This surface features several different properties the algorithms have to deal with, like a circular shaped foundation, flat areas and a steep peak and is therefore considered difficult to reconstruct. The control net is setup with 256 control points in a 16x16 grid, $surf_{pre}$ is obtained via the SVD approach (see chapter 2.5.2). The position of the control points is optimized for:

1. average distance over all minimal distances between the NURBS surface and the control points.

2. regularity of the NURBS surface.

The first objective is chosen because it represents the proximity to the surface adequately and is yet relatively fast to evaluate. The second objective is employed because it is (partially) not correlative to the first objective and prefers smoother surfaces. Many real objects to be reconstructed feature smooth areas, hence the second objective punishes control nets where the corresponding NURBS surface is irregular in those areas.



(a) The piston (b) Result from the scan

(c) Filtered scan points (d) Pre-optimized surface

Figure 4.7: Steps leading to the pre-optimized surface in the hybrid approach. Pictures: © 2009 ISF

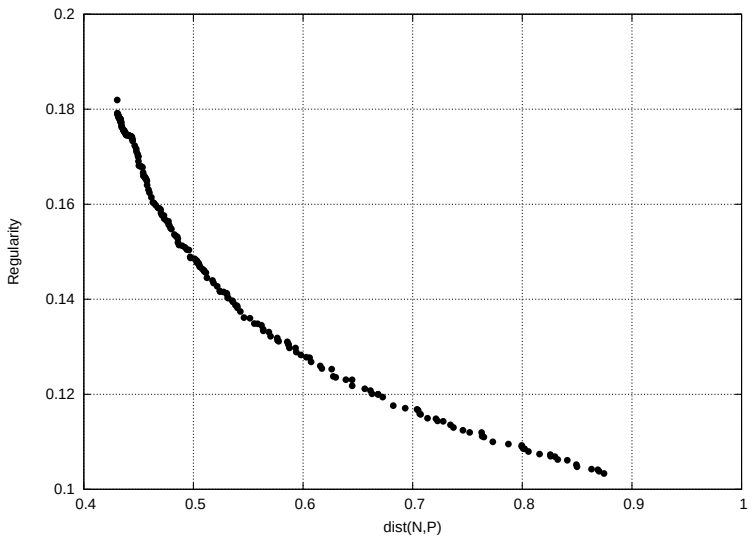


Figure 4.8: Approximation set for the surface

As no suitable reference set is available for the surface, the non-dominated set of all computed solutions is used instead. Before the ε-, hypervolume- and R2 indicators are calculated, solutions with a worse fitness value than $surf_{f1}$ regarding the first objective and solutions with a worse fitness value than $surf_{pre}$ regarding the second objective are removed. It may not seem fair to compare interpolation alone and interpolation with additional filtering in the (limited) area between $surf_{pre}$ and $surf_{f1}$, as the strength of the approach by Weinert at al. is to cover the entire bandwidth between regularity and proximity to the scan points. However, since the large areas of the Pareto set are not useful for practical purposes on the given surface, taking solutions into consideration which do not meet quality constraints of the user anyway, is not feasible.

4.2.3 Results

In the experiments, some algorithms have been used which only differ in the variation operator. Algorithms employing DE_NURBS are denoted with the index "$_N$" in the tables below. SMS-EMOA is analyzed in different types of initialization and with two different parameter combinations. To tell these apart an index is appended as well. Index "$_B$" denotes the best overall parameter combination determined in chapter 3, "$_D$" marks the recommended (default) parameter combination, "$_R$" labels the initial population was initialized randomly and "$_W$" means the interpolation proposed by Weinert et al. [59] was used instead.

	ε-indicator			**hyp-indicator**			**R2-indicator**			**All3**
Algorithm	median	mean	std	median	mean	std	med	mean	std	**indicator**
DE-SMS-CD$_N$	**0.0390**	0.0459	0.0202	**0.0384**	0.0383	0.0077	**0.0103**	0.0105	0.0027	**0.0000**
DE-SMS-PD$_N$	0.0425	0.0443	0.0064	0.0449	0.0458	0.0063	0.0118	0.0121	0.0018	0.0071
DE-SMS-CD	0.0532	0.0687	0.0488	0.0489	0.0534	0.0170	0.0128	0.0154	0.0085	0.0151
GDE3	0.0459	0.0473	0.0039	0.0571	0.0582	0.0037	0.0153	0.0155	0.0010	0.0199
DE-SMS-PD	0.0511	0.0556	0.0110	0.0697	0.0694	0.0082	0.0179	0.0184	0.0026	0.0324
GDE3$_N$	0.0691	0.0677	0.0054	0.0828	0.0828	0.0035	0.0224	0.0222	0.0011	0.0530
SMS-EMOA$_D$	0.0871	0.0866	0.0030	0.1068	0.1064	0.0028	0.0277	0.0276	0.0013	0.0810
NSGA-II	0.1070	0.1078	0.0043	0.1345	0.1344	0.0048	0.0333	0.0335	0.0017	0.1123
SMS-EMOA$_B$	0.1130	0.1132	0.0014	0.1555	0.1557	0.0014	0.0412	0.0409	0.0010	0.1364
Initial[1]	0.1159			0.1649			0.0436			0.1459
SMS-EMOA$_W$	0.6665	0.6565	0.0274	0.5199	0.5204	0.0092	0.2397	0.2360	0.0096	0.8144
SMS-EMOA$_R$	0.8263	0.8264	0.0125	0.5987	0.5988	0.0126	0.3018	0.3019	0.0063	1.0000

Table 4.2: Performance on surface reconstruction

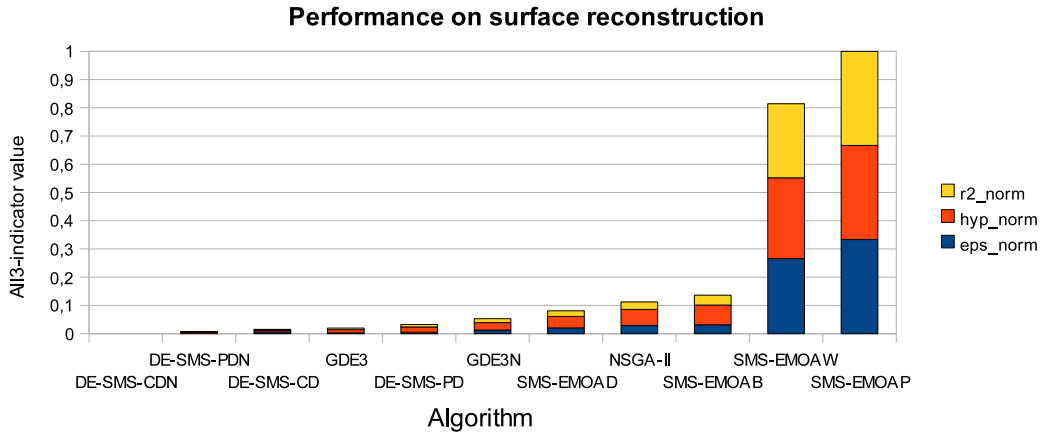


Figure 4.9: All3 indicator on surface reconstruction

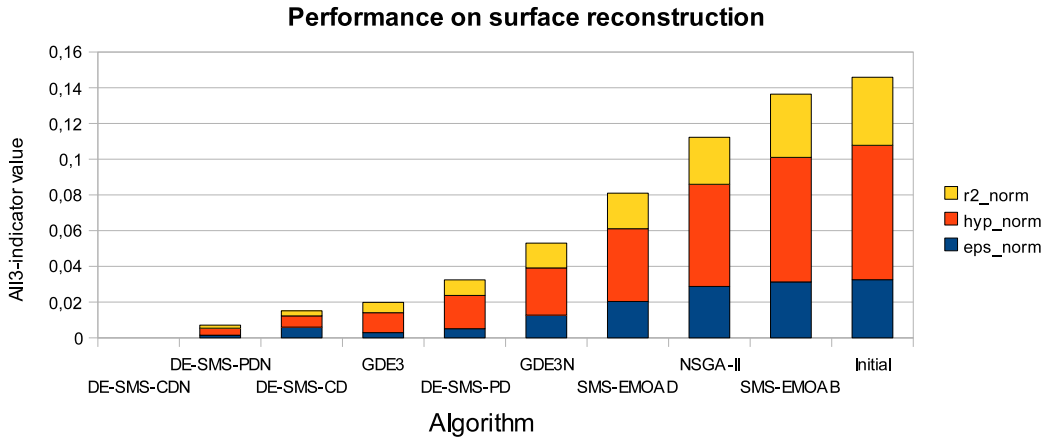


Figure 4.10: All3 indicator on surface reconstruction

	div+ indicator		
Algorithm	median	mean	std
DE-SMS-CD$_N$	850.70	853.93	42.19
DE-SMS-PD$_N$	900.73	903.19	25.59
DE-SMS-CD	995.58	977.90	70.38
GDE3	805.05	806.43	15.56
DE-SMS-PD	975.59	976.28	25.82
GDE3$_N$	729.78	730.30	13.72
SMS-EMOA$_D$	1958.18	1952.36	57.06
NSGA-II	1518.80	1517.68	46.11
SMS-EMOA$_B$	1127.59	1129.96	27.15
Initial[1]	648.94		
SMS-EMOA$_W$	**4225.55**	4204.49	81.93
SMS-EMOA$_R$	491.30	480.69	86.45

Table 4.3: Diversity on surface reconstruction (div+ indicator)

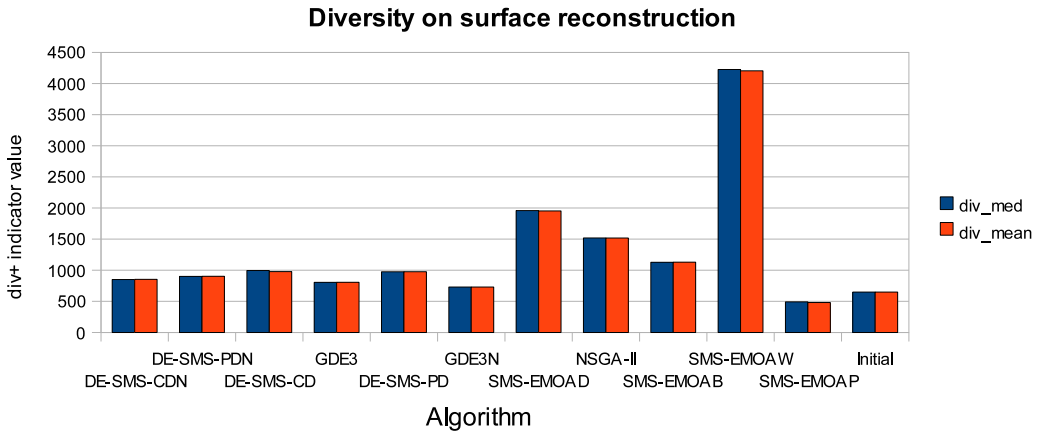


Figure 4.11: div+ indicator on surface reconstruction

[1] from SMS-EMOA$_B$

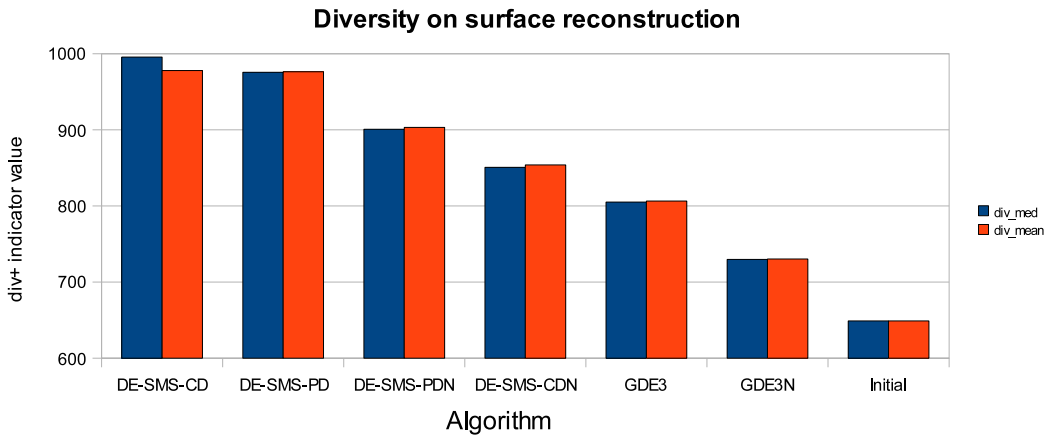


Figure 4.12: div+ indicator on surface reconstruction

4.2.4 Observations

From the results several expected patterns can be noticed:

- The biggest improvement of all is achieved by the smart creation of the initial population.
- The new variation operator DE_NURBS is better on DE-SMS-CD and DE-SMS-PD than the standard differential evolution from GDE3. In GDE3 itself the situation is reverse.
- The differential evolution from GDE3 is a better variation operator for surface reconstruction then SBX and PM, hence DE-SMS-CD, DE-SMS-PD and GDE3 are preferable to NSGA-II and SMS-EMOA.

Surprisingly also some unexpected results can be observed from the data:

- More diversity in decision space is maintained through both DE-SMS-CD and DE-SMS-PD compared to GDE3 according to the div+ indicator, though the difference in div+ indicator values between the algorithms is smaller than expected from the experiments in chapter 3.
- Algorithms using the SBX and PM variation operators reach a higher diversity than algorithms using differential evolution regarding the div+ indicator.
- Without the new variation operator, GDE3 performs better than DE-SMS-PD according to the All3 indicator.
- SMS-EMOA$_B$ doesn't seem to achieve a significant improvement.

No.	Expected rank	Rank (new variation)	Rank
1	DE-SMS-CD	DE-SMS-CD	DE-SMS-CD
2	DE-SMS-PD	DE-SMS-PD	GDE3
3	GDE3	GDE3	DE-SMS-PD
4	SMS-EMOA	SMS-EMOA$_D$	
5	SMS-EMOA$_D$	NSGA-II	
6	NSGA-II	SMS-EMOA	

Table 4.4: Expected ranking and ranking determined from the experiments

- In the area examined, interpolation alone improves performance only slightly compared to interpolation and filtering.

As far as the rank is concerned the situation is inconsistent (table 4.4). If only the algorithms using DE_NURBS are considered, the ranking is maintained as determined in the previous chapter with the exception that the best parameter combination for SMS-EMOA does not work very well at all and therefore only reaches the last rank. If the algorithms using the standard variation operator from GDE3 are taken into account, also GDE3 and DE-SMS-PD switch positions.

4.2.5 Discussion

In this section a more detailed analysis of the observations is given, with emphasis on the practical use of the algorithm which turned out to be the best. The behavior of all algorithms is analyzed and the implications for surface reconstruction, derived from the behavior, are explained.

The initial population is the key

With the right initialization (mean filter and interpolation), every algorithm investigated outperforms SMS-EMOA$_I$ and SMS-EMOA$_R$ according to the metrics used in the experiments. Since the initial population already offers better indicator values than SMS-EMOA$_I$ and SMS-EMOA$_R$ can calculate with 20000 function evaluations, this is no surprise. But the difference in performance shows how important a good initial population for surface reconstruction is. Problem-specific knowledge can additionally be encoded into the way the initial population is created. Future research could investigate other real-world problems for potential improvement regarding the use of problem-specific knowledge to create the initial population. Most objects which can be adequately described by a NURBS surface contain smooth parts. Approximation to the scan points via the SVD approach does not care about regularity, which is why the additional objective is introduced into

the evolutionary algorithm. The variation in the surface does not adequately describe the surface, but nevertheless the surface happens to show a good proximity to the scan points. The linear interpolation between $surf_{pre}$ and $surf_{flat}$ creates surfaces lying fairly distributed between the interpolated surfaces in objective function space. With each step from the $surf_{pre}$ to $surf_{flat}$ all areas of the interpolated surface almost equally decrease performance regarding the first objective and increase performance regarding the second objective. By using the mean filter, areas of the surface with bigger variation are smoothed with one or few filtering iterations without decreasing the proximity to the points as far as interpolation alone does. The mean filter literally "irons" the surface. The obtained surface $surf_{f1}$ can then be used for interpolation as usual, the only difference is, that its fitness is much better than $surf_{flat}$ and so are the interpolated solutions.

Best algorithm for surface reconstruction

According to the the All3 indicator the best algorithm is DE-SMS-CD using the DE_NURBS variation operator (DE-SMS-CD$_N$ in the tables). The Pareto fronts of DE-SMS-CD and DE-SMS-CD$_N$ show the standard variation operator is unable to find better solutions regarding the first objective than $surf_{pre}$ already offers, improvements are only obtained between the two interpolation points. The reason is simple. Most areas of the surface $surf_{pre}$ already feature a good approximation of the desired surface regarding the proximity between surface and scan points, only some small areas need to be adjusted. Uniformly distributed choice of many dimensions in decision space selects less likely control points from a region that can be improved, than it selects control points from a region, where an improvement can be expected. On average more harm is done than improvement achieved as far as the first objective is concerned in the upper left part of the Pareto front. Due to the locally restricted choice of control points DE_NURBS is able to improve a region that needs improvement without downsides elsewhere on the surface.

By applying the mean filter to $surf_{pre}$ and interpolating additional solutions between $surf_{pre}$ and $surf_{f1}$ solutions are generated with a worse fitness regarding the first objective. The closer to $surf_{f1}$ the solution is, the worse its fitness as far as proximity to the scan points is concerned. Almost in all areas of the surface, the proximity to the scan points can be improved, thus the standard variation operator can improve multiple parts of the surface at the same time without too many deteriorations elsewhere on the surface.

Regarding the second objective the standard variation operator slightly outperforms DE_NURBS in the right part of the Pareto front. As the crossover parameter CR is set to 0.14, on average approximately every 7th of the 768 dimensions (≈ 110) in decision space is recombined during every function evaluation. The probability DE_NURBS modifies exactly n control points is $CR^n(0.9^n)$, therefore the probability $\lfloor \frac{110}{3} \rfloor$ control points are

modified equals $CR^{\lfloor \frac{110}{3} \rfloor} = 0.9^{36} = 0.0225$. Thus, on average a much smaller amount of control points is modified by DE_NURBS (see below).

Nevertheless children bred by the DE_NURBS variation operator have a much greater chance of success than those created via differential evolution from GDE3. For DE-SMS-CD almost three times as many children survive, for DE-SMS-PD the increase is even higher (approx. 3.5 times). For GDE3 almost twice as many children survive when DE_NURBS is used. GDE3 cannot profit from the better variation operator. The parents are chosen uniformly distributed from the entire population, hence the distance between the parents is often too great. As a result the area is modified too far into the right direction and the child therefore features worse fitness regarding both objectives. GDE3-D_N and GDE3-SMS-D_N breed the same way as DE-SMS-CD does and should show a similar behavior as DE-SMS-CD. Those two algorithms can therefore be investigated in the future regarding their performance in surface reconstruction. Due to the smaller population size, per

Algorithm	**#Children**
DE-SMS-CD	2209
DE-SMS-CD$_N$	5994
DE-SMS-PD	1316
DE-SMS-PD$_N$	4782
GDE3	1549
GDE3$_N$	3013
NSGA-II	1869
SMS-EMOA$_D$	6519
SMS-EMOA$_B$	8342
SMS-EMOA$_W$	7735
SMS-EMOA$_I$	11680

Table 4.5: Number of children surviving the generation they were bred in

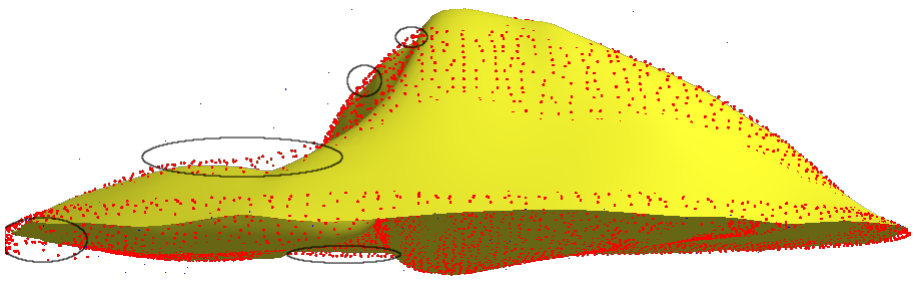
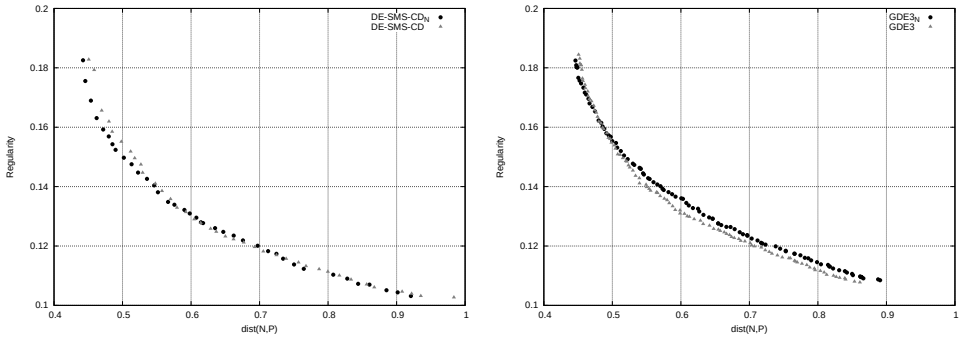


Figure 4.13: Areas, where the distance between surface and scan points needs improvement

individual in the population, more optimizations than in GDE3 are possible. Usually one

CP	Modified dimensions	Probability
1	3	0.1
2	6	0.19
3	9	0.27
4	12	0.34
5	15	0.41
7	21	> 0.5
14	42	> 0.75
22	66	> 0.90
29	87	> 0.96
44	132	> 0.99

Table 4.6: Probability less than CP control points are modified by DE_NURBS (with $CR = 0.9$)



(a) DE-SMS-CD (b) GDE3

Figure 4.14: Comparison: standard vs new variation operator

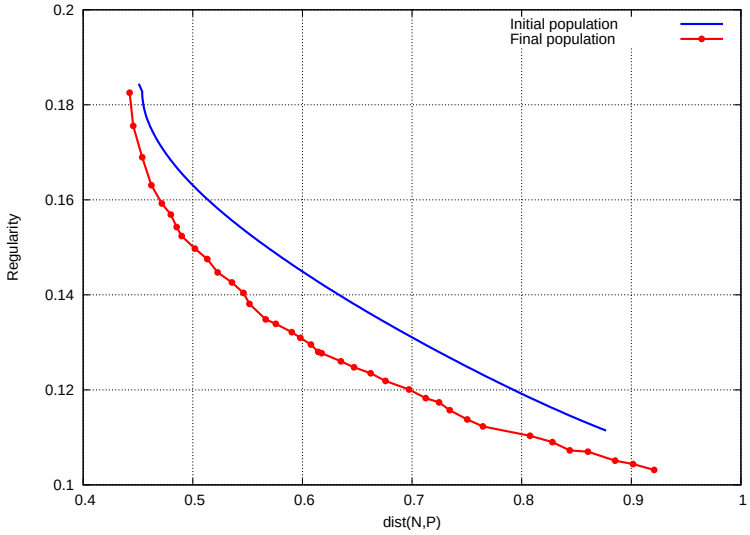
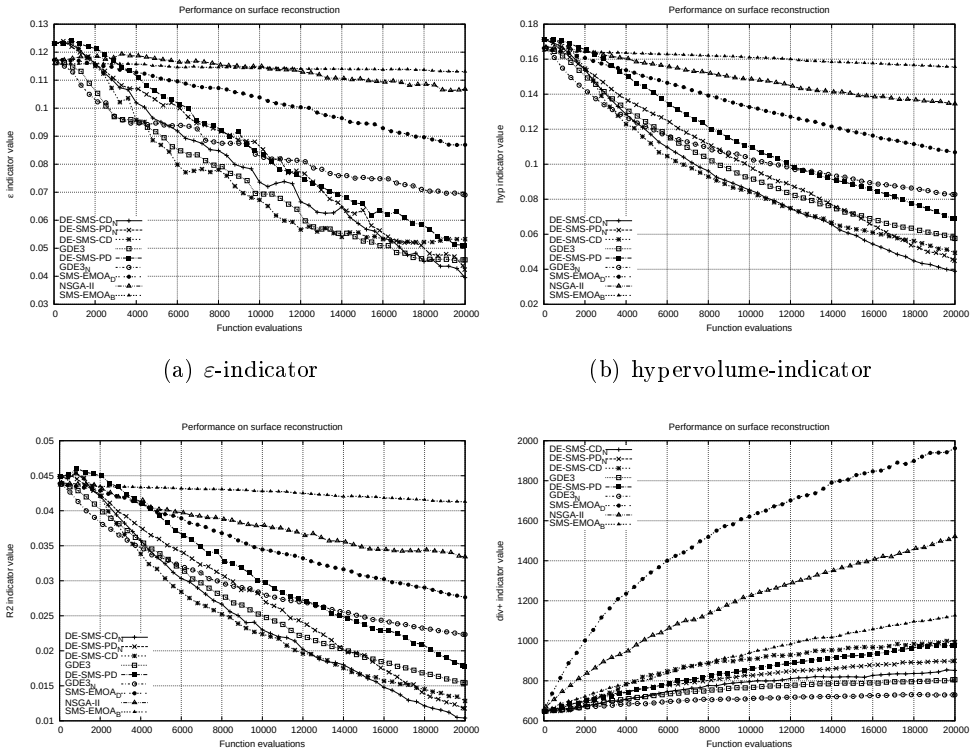


Figure 4.15: Interpolation of new solutions from the final population of DE-SMS-CD$_N$

of the downsides of a smaller population size is the smaller number of solutions to choose from when the optimization process is complete. For surface reconstruction this does not hold since the same way the initial population was generated, additional solutions can be interpolated, where desired, between neighbors in objective function space. If these solutions are not evaluated, no additional function evaluation is required. DE-SMS-PD behaves analog to DE-SMS-CD. Since DE-SMS-PD is more restrictive concerning the selection, the performance is slightly worse and the div+ indicator value is slightly better.

All algorithms using simulated binary crossover and polynomial mutation can be considered inferior to the algorithms using differential evolution. For example, after less than 6500 function evaluations DE-SMS-CD$_N$ attains the same hypervolume indicator value as SMS-EMOA$_D$. The reasons have been described in the motivation for the development of DE_NURBS, basically simulated binary crossover only creates marginally improved children, while mutation is the dominant variation operator. Unfortunately polynomial mutation also favors the generation of malformed surfaces (see section 4.1.1). The parameter set of SMS-EMOA$_B$ practically disables polynomial mutation ($p_m = 0.000145$) in surface reconstruction. Approximately only one child in 10 is mutated in one dimension. As we established above, mutation is causing most of the improvements in SMS-EMOA on surface reconstruction, almost disabling mutation it is therefore not feasible. The previously determined best overall parameter set is therefore not useful for surface reconstruction. Though both algorithms use the same parameter combination, SMS-EMOA$_D$ outperforms NSGA-II mostly due to the steady-state approach. More interesting than their performance is the diversity attained by those algorithms. For SMS-EMOA$_D$ the div+ indicator value is more than twice as high as for the algorithms using differential evolution. At



(a) ε-indicator

(b) hypervolume-indicator

(c) R2-indicator

(d) div+ indicator

Figure 4.16: Development of the indicator values

first sight this seems a desirable feature despite the performance issues SMS-EMOA suffers from, especially as preserving additional diversity is the major goal of this work. A more detailed analysis of how this diversity is attained shows, the diversity is indeed higher, but in the case of surface reconstruction not desirable. Again polynomial mutation is the problem: large, punctual changes in the surfaces cause the difference between the individuals to grow, thus the diversity increases and the div+ indicator detects this correctly. However many of the mutations cause an unwanted shape-disorder, like "nibbled edges" or "blister". Figuratively speaking the additional diversity is bought at the cost of mostly unusable surfaces and thus neither desirable nor a feature.

To recapitulate the results, the combination of the established hybrid approach (see chapter 2.5.2) with DE-SMS-CD$_N$ is a huge improvement for surface reconstruction. The most important factors for the improvement are

1. the enhanced initial population,

2. the improved variation operator for NURBS surfaces in combination with the cluster analysis during breeding.

Chapter 5

Summary and future work

5.1 Summary

The first two chapters introduced the basic algorithms and methods used to create an improved algorithm for surface reconstruction and described state-of-the-art approaches for the reconstruction of a surface from a huge number of scan points. A hybrid approach between numerically solving a set of linear equations to determine a surface with good fitness values regarding proximity to the scan points and a multi-objective evolutionary algorithm to find solutions additionally satisfying regularity of the surface, computes acceptable results in a reasonable amount of time.

In chapter 3 several enhancements to the algorithm GDE3 [5] for performance diversity were proposed. The performance related improvements included known tweaks for evolutionary algorithms, like S-metric selection and the steady-state selection scheme used in SMS-EMOA [3]. Significant changes to the selection of parents were investigated. Based on a cluster analysis the population can be partitioned into groups of parents with similar properties. By choosing similar or totally different parents, the search can be divided into local and global search. This improvement alone turned out not to be successful. To attain more diversity the selection was modified as well. The main idea behind the modifications was to systematically limit the group of individuals a child can replace. If a child only can replace similar individuals, the population remains more diverse than if always the worst individual is deleted. Combined with the cluster-based choice of parents this approach showed good performance on test problems and was able to search the decision space more widely.

Some of the algorithms were applied to surface reconstruction to see if any improvement regarding diversity and performance can be achieved during the multi-objective optimization part. To increase performance problem-specific knowledge was employed to create the initial population. With a pre-optimized surface already available, several other also very good solutions can be obtained by first smoothing the control net from the surface

already available with a mean filter and then linearly interpolate between the original and the smoothened surface. Using this measure alone dramatically improved the results.

Additionally the variation operator DE_NURBS (differential evolution for NURBS surfaces), which exploits the local support of NURBS surfaces, was designed to improve shortcomings of the variation operators simulated binary crossover and polynomial mutation. If used together with cluster-based breeding, DE_NURBS is able to also significantly improve the performance of a multi-objective evolutionary algorithm for surface reconstruction. Both the general enhancements from chapter 3 and the special optimizations to surface reconstruction turned out to work well together. Even though the improvements to surface reconstruction are explicit, it remains hard to put reliable numbers to the overall performance gain.

5.2 Future work

During this thesis a lot of new ideas came up which cannot be covered by this work alone. The algorithms developed offer a great deal of opportunities for further enhancement, both for multi-objective evolutionary algorithms in general and optimizers for surface reconstruction. The most important new ideas are briefly discussed below.

General

As only the behavior of the algorithms for 20000 function evaluations was investigated, parameter combinations for more or less function evaluations cannot be recommended. A full parameter optimization [52] may prove useful before applying the algorithm to other real-world problems. But even for 20000 function evaluations the parameter L can be improved. Some tests on the function evaluations indicate $L = 0.8$ is a better choice, but a more profound analysis is required to support this claim. Porting back some of the improvements to single-objective differential evolution could be explored to further enhance single-objective algorithms.

Clustering

One type of cluster analysis (HACM with number criterion) with one distance measure (euclidean distance) was implemented and tested. However clustering can be done in various different ways, future research should hence compare suitable clustering algorithms for use in evolutionary algorithms. The C clustering library [49] already offers implementations of various algorithms and distance measures.

Variation operator

Even though cluster-based breeding alone did not outperform uniformly distributed choice of the parents in the experiments on the test functions, the concept of dividing into local and global search offers opportunities for further upgrades. For instance, a dynamic step-size can be integrated into the parameter F. For each cluster the value of F can be stored and updated dynamically (e.g. implementing the 1/5th rule by Rechenberg [61]) according to common approaches. If local search is performed, the F value is determined from the cluster the parents are chosen from, if all parents are chosen from different clusters, the mean value of the F values from the clusters containing $\vec{x}_1$ and $\vec{x}_2$ is computed. Such an approach could improve convergence while global search prevents the algorithm from getting stuck in a local optimum prematurely.

Other problems

There are also lots of other test and real-world problems, evolutionary algorithms are applied to, where e.g. DE-SMS-CD can promise improvements because of its major features. One step to further acceptance of this algorithm could be to attend MOEA contests (assuming it is able to attain decent performance on the test problems involved).

Surface reconstruction

As far as surface reconstruction is concerned, the performance of DE-SMS-CD$_N$ on other surfaces needs to be evaluated. Additional or other objectives can be used and also more than two objectives should be tested. Interpolation with three or more objectives also needs modifications. A parameter optimization for DE-SMS-CD on surface reconstruction can also lead to additional performance improvements. Since the variation operator DE_NURBS was proven successful and exploits the local support feature of NURBS surfaces, determining areas of a surface, where the most improvement can be achieved, seems the next logical step towards better performance of the evolutionary algorithm within the hybrid approach. Once these areas are determined, the focus of the variation operator can be limited or at least more often applied there.

Appendix A

Data from experiments

A.1 Latin Hypercube Designs

GDE3(-SMS), DE-SMS-P			Algorithms with clustering					NSGA-II, SMS-EMOA				
μ	CR	F	μ	CR	F	L	C/μ	μ	$t(p_m = D^t)$	p_c	η_m	η_c
22	0.65	0.41	22	0.62	0.67	0.73	0.12	22	-1.54	0.9	20	25
28	0.38	0.22	28	0.4	0.3	0.49	0.07	28	-1.24	0.87	20	6
34	0.43	0.56	34	0.76	0.27	0.53	0.18	34	-1.21	0.61	15	28
39	0.77	0.65	39	0.46	0.41	0.09	0.17	39	-0.64	0.78	25	23
45	0.28	0.8	45	0.14	0.49	0.61	0.15	45	-1.9	0.67	18	15
50	0.2	0.38	50	0.36	0.78	0.19	0.1	50	-0.76	0.75	6	15
56	0.57	0.82	56	0.28	0.86	0.57	0.2	56	-1.06	0.55	17	9
62	0.64	0.17	62	0.43	0.48	0.85	0.24	62	-1.63	0.92	8	13
67	0.86	0.25	67	0.65	0.64	0.29	0.25	67	-1.36	0.74	30	12
73	0.17	0.61	73	0.41	0.19	0.93	0.14	73	-1.57	0.56	6	18
79	0.48	0.31	79	0.72	0.4	0.07	0.08	79	-0.7	0.99	16	16
84	0.83	0.83	84	0.86	0.7	0.23	0.15	84	-0.88	0.89	12	29
90	0.61	0.52	90	0.27	0.14	0.43	0.21	90	-1.66	0.68	29	26
95	0.41	0.7	95	0.83	0.32	0.69	0.07	95	-1.15	0.76	17	17
101	0.18	0.18	101	0.25	0.73	0.77	0.06	101	-0.58	0.73	23	8
107	0.88	0.54	107	0.2	0.35	0.05	0.09	107	-1.81	0.81	8	27
112	0.35	0.48	112	0.78	0.43	0.97	0.16	112	-1.87	0.94	24	14
118	0.73	0.33	118	0.73	0.84	0.75	0.21	118	-0.94	0.54	28	21
123	0.46	0.1	123	0.19	0.56	0.17	0.23	123	-0.52	0.59	15	21
129	0.15	0.78	129	0.38	0.72	0.99	0.16	129	-1.12	0.93	29	24
135	0.67	0.73	135	0.7	0.17	0.11	0.22	135	-1.78	0.66	9	7
140	0.44	0.88	140	0.57	0.11	0.41	0.12	140	-0.91	0.88	9	5
146	0.12	0.43	146	0.49	0.52	0.45	0.1	146	-1.69	0.51	27	12
152	0.78	0.14	152	0.48	0.88	0.13	0.19	152	-1.93	0.57	18	27
157	0.52	0.36	157	0.75	0.8	0.37	0.05	157	-1.33	0.7	21	5
163	0.33	0.27	163	0.12	0.25	0.65	0.09	163	-0.97	0.64	5	28
168	0.49	0.62	168	0.52	0.51	0.51	0.19	168	-1.45	1	13	22
174	0.27	0.67	174	0.6	0.12	0.79	0.22	174	-1	0.69	22	30
180	0.7	0.51	180	0.11	0.76	0.35	0.14	180	-1.03	0.98	26	9
185	0.9	0.35	185	0.16	0.36	0.89	0.2	185	-1.27	0.52	16	19
191	0.85	0.72	191	0.84	0.75	0.87	0.11	191	-1.84	0.86	23	29
197	0.69	0.9	197	0.88	0.46	0.59	0.23	197	-1.39	0.8	5	16
202	0.22	0.12	202	0.51	0.44	0.95	0.08	202	-0.55	0.84	10	20
208	0.72	0.28	208	0.89	0.38	0.21	0.12	208	-0.85	0.6	7	10
213	0.31	0.44	213	0.3	0.81	0.67	0.24	213	-1.75	0.91	14	6
219	0.23	0.86	219	0.56	0.65	0.01	0.1	219	-0.61	0.77	28	20
225	0.56	0.15	225	0.44	0.89	0.63	0.11	225	-1.96	0.72	25	13
230	0.62	0.69	230	0.64	0.57	0.03	0.22	230	-1.3	0.79	19	18
236	0.1	0.59	236	0.24	0.28	0.15	0.16	236	-0.73	0.53	22	11
241	0.39	0.64	241	0.54	0.22	0.25	0.06	241	-1.99	0.85	12	19
247	0.14	0.3	247	0.81	0.2	0.81	0.13	247	-1.18	0.71	30	10
253	0.54	0.39	253	0.22	0.6	0.33	0.06	253	-0.82	0.96	21	26
258	0.82	0.49	258	0.8	0.83	0.39	0.18	258	-1.72	0.58	7	22
264	0.51	0.85	264	0.32	0.33	0.47	0.24	264	-1.42	0.83	11	30
270	0.8	0.77	270	0.17	0.62	0.83	0.13	270	-0.67	0.82	19	7
275	0.36	0.2	275	0.59	0.54	0.91	0.2	275	-1.48	0.62	27	25
281	0.75	0.23	281	0.35	0.16	0.71	0.14	281	-1.6	0.95	26	17
286	0.25	0.75	286	0.67	0.24	0.31	0.18	286	-1.09	0.97	10	14
292	0.3	0.46	292	0.68	0.59	0.55	0.08	292	-1.51	0.63	14	8
298	0.59	0.57	298	0.33	0.68	0.27	0.17	298	-0.79	0.65	13	23

Table A.1: LHS design file for parameter optimization

A.2 Kruskal-Wallis test results

The parameter α was set to 0.01 for all problems. The tables contain the p-values for each pair of algorithms A_{row}, A_{column} for the alternative hypothesis (= the indicator values of A_{column} are significantly better than the ones of A_{row}).

	DE-SMS-C	DE-SMS-CD	DE-SMS-PD	GDE3-SMS-D	GDE3-D	SMS-EMOA(DE)	GDE3	DE-SMS-P	GDE3-SMS	NSGA-II	SMS-EMOA
DE-SMS-C	-	2.82e-03	1.29e-03	> 0.01	4.61e-03	2.44e-06	1.21e-16	7.81e-09	2.01e-07	1.10e-12	3.12e-15
DE-SMS-CD	> 0.01	-	> 0.01	> 0.01	> 0.01	> 0.01	3.83e-09	1.28e-03	8.37e-03	3.59e-06	4.76e-08
DE-SMS-PD	> 0.01	> 0.01	-	> 0.01	> 0.01	> 0.01	1.46e-08	2.79e-03	> 0.01	1.07e-05	1.67e-07
GDE3-SMS-D	> 0.01	> 0.01	> 0.01	-	> 0.01	> 0.01	2.96e-10	2.66e-04	2.22e-03	4.29e-07	4.24e-09
GDE3-D	> 0.01	> 0.01	> 0.01	> 0.01	-	> 0.01	1.54e-09	7.38e-04	5.27e-03	1.69e-06	2.01e-08
SMS-EMOA(DE)	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	-	2.83e-05	> 0.01	> 0.01	3.65e-03	1.81e-04
GDE3	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	-	> 0.01	> 0.01	> 0.01	> 0.01
DE-SMS-P	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	1.88e-03	-	> 0.01	> 0.01	7.61e-03
GDE3-SMS	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	2.19e-04	> 0.01	-	> 0.01	1.14e-03
NSGA-II	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	-	> 0.01
SMS-EMOA	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	-

Table A.2: ε-indicator on OKA2

	DE-SMS-C	DE-SMS-CD	DE-SMS-PD	GDE3-SMS-D	GDE3-D	SMS-EMOA(DE)	GDE3	DE-SMS-P	GDE3-SMS	NSGA-II	SMS-EMOA
DE-SMS-C	-	> 0.01	3.15e-03	5.15e-03	1.63e-03	1.59e-06	2.48e-15	3.94e-10	2.42e-08	2.39e-18	6.86e-18
DE-SMS-CD	> 0.01	-	> 0.01	> 0.01	> 0.01	6.15e-03	2.17e-09	2.33e-05	4.17e-04	6.47e-12	1.59e-11
DE-SMS-PD	> 0.01	> 0.01	-	> 0.01	> 0.01	> 0.01	3.31e-08	1.72e-04	2.25e-03	1.36e-10	3.19e-10
GDE3-SMS-D	> 0.01	> 0.01	> 0.01	-	> 0.01	> 0.01	1.37e-08	9.10e-05	1.32e-03	5.04e-11	1.20e-10
GDE3-D	> 0.01	> 0.01	> 0.01	> 0.01	-	> 0.01	9.77e-08	3.73e-04	4.24e-03	4.60e-10	1.06e-09
SMS-EMOA(DE)	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	-	2.26e-04	> 0.01	> 0.01	3.91e-06	7.50e-06
GDE3	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	-	> 0.01	> 0.01	> 0.01	> 0.01
DE-SMS-P	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	-	> 0.01	1.78e-03	2.82e-03
GDE3-SMS	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	3.76e-03	> 0.01	-	1.31e-04	2.27e-04
NSGA-II	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	-	> 0.01
SMS-EMOA	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	-

Table A.3: Hypervolume-indicator on OKA2

	DE-SMS-C	DE-SMS-CD	DE-SMS-PD	GDE3-SMS-D	GDE3-D	SMS-EMOA(DE)	GDE3	DE-SMS-P	GDE3-SMS	NSGA-II	SMS-EMOA
DE-SMS-C	-	> 0.01	1.59e-03	2.45e-04	8.04e-05	3.07e-04	2.59e-11	7.97e-10	2.58e-07	1.40e-16	2.63e-15
DE-SMS-CD	> 0.01	-	> 0.01	> 0.01	> 0.01	> 0.01	6.12e-07	8.85e-06	6.20e-04	2.51e-11	3.12e-10
DE-SMS-PD	> 0.01	> 0.01	-	> 0.01	> 0.01	> 0.01	6.80e-05	5.93e-04	> 0.01	1.16e-08	1.07e-07
GDE3-SMS-D	> 0.01	> 0.01	> 0.01	-	> 0.01	> 0.01	5.15e-04	3.43e-03	> 0.01	1.93e-07	1.50e-06
GDE3-D	> 0.01	> 0.01	> 0.01	> 0.01	-	> 0.01	1.38e-03	7.96e-03	> 0.01	8.02e-07	5.67e-06
SMS-EMOA(DE)	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	-	4.14e-04	2.85e-03	> 0.01	1.41e-07	1.12e-06
GDE3	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	-	> 0.01	> 0.01	> 0.01	> 0.01
DE-SMS-P	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	-	> 0.01	6.83e-03	> 0.01
GDE3-SMS	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	-	1.97e-04	8.90e-04
NSGA-II	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	-	> 0.01
SMS-EMOA	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	-

Table A.4: R2-indicator on OKA2

	DE-SMS-C	DE-SMS-CD	DE-SMS-PD	GDE3-SMS-D	GDE3-D	SMS-EMOA(DE)	GDE3	DE-SMS-P	GDE3-SMS	NSGA-II	SMS-EMOA
DE-SMS-C	-	> 0.01	> 0.01	6.12e-03	5.79e-05	> 0.01	9.37e-03	3.05e-05	> 0.01	8.12e-04	5.58e-05
DE-SMS-CD	> 0.01	-	> 0.01	> 0.01	1.37e-03	> 0.01	> 0.01	8.12e-04	> 0.01	> 0.01	1.33e-03
DE-SMS-PD	> 0.01	> 0.01	-	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01
GDE3-SMS-D	> 0.01	> 0.01	> 0.01	-	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01
GDE3-D	> 0.01	> 0.01	> 0.01	> 0.01	-	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01
SMS-EMOA(DE)	> 0.01	> 0.01	> 0.01	6.32e-03	6.06e-05	-	9.66e-03	3.20e-05	> 0.01	8.44e-04	5.84e-05
GDE3	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	-	> 0.01	> 0.01	> 0.01	> 0.01
DE-SMS-P	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	-	> 0.01	> 0.01	> 0.01
GDE3-SMS	> 0.01	> 0.01	> 0.01	3.50e-03	2.65e-05	> 0.01	5.50e-03	1.36e-05	-	4.17e-04	2.55e-05
NSGA-II	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	-	> 0.01
SMS-EMOA	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	-

Table A.5: ε-indicator on R_ZDT4

	DE-SMS-C	DE-SMS-CD	DE-SMS-PD	GDE3-SMS-D	GDE3-D	SMS-EMOA(DE)	GDE3	DE-SMS-P	GDE3-SMS	NSGA-II	SMS-EMOA
DE-SMS-C	-	> 0.01	> 0.01	> 0.01	2.75e-05	> 0.01	> 0.01	8.64e-06	> 0.01	6.68e-07	5.41e-08
DE-SMS-CD	> 0.01	-	> 0.01	> 0.01	3.75e-04	> 0.01	> 0.01	1.37e-04	> 0.01	1.43e-05	1.51e-06
DE-SMS-PD	> 0.01	> 0.01	-	8.65e-03	1.95e-05	> 0.01	> 0.01	6.03e-06	> 0.01	4.49e-07	3.53e-08
GDE3-SMS-D	> 0.01	> 0.01	> 0.01	-	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	4.43e-03	9.07e-04
GDE3-D	> 0.01	> 0.01	> 0.01	> 0.01	-	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01
SMS-EMOA(DE)	> 0.01	> 0.01	> 0.01	> 0.01	1.05e-03	-	> 0.01	4.13e-04	> 0.01	4.96e-05	5.87e-06
GDE3	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	-	> 0.01	> 0.01	3.64e-03	7.24e-04
DE-SMS-P	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	-	> 0.01	> 0.01	> 0.01
GDE3-SMS	> 0.01	> 0.01	> 0.01	> 0.01	4.91e-03	> 0.01	> 0.01	2.16e-03	-	3.28e-04	4.77e-05
NSGA-II	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	-	> 0.01
SMS-EMOA	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	-

Table A.6: Hypervolume-indicator on R_ZDT4

	DE-SMS-C	DE-SMS-CD	DE-SMS-PD	GDE3-SMS-D	GDE3-D	SMS-EMOA(DE)	GDE3	DE-SMS-P	GDE3-SMS	NSGA-II	SMS-EMOA
DE-SMS-C	-	> 0.01	> 0.01	4.67e-03	3.67e-05	> 0.01	6.43e-03	4.48e-06	> 0.01	1.15e-04	7.61e-06
DE-SMS-CD	> 0.01	-	> 0.01	> 0.01	1.16e-03	> 0.01	> 0.01	2.06e-04	> 0.01	2.89e-03	3.19e-04
DE-SMS-PD	> 0.01	> 0.01	-	> 0.01	6.82e-03	> 0.01	> 0.01	1.55e-03	> 0.01	> 0.01	2.27e-03
GDE3-SMS-D	> 0.01	> 0.01	> 0.01	-	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01
GDE3-D	> 0.01	> 0.01	> 0.01	> 0.01	-	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01
SMS-EMOA(DE)	> 0.01	> 0.01	> 0.01	> 0.01	1.71e-04	-	> 0.01	2.44e-05	> 0.01	4.89e-04	4.00e-05
GDE3	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	-	> 0.01	> 0.01	> 0.01	> 0.01
DE-SMS-P	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	-	> 0.01	> 0.01	> 0.01
GDE3-SMS	> 0.01	> 0.01	> 0.01	9.42e-03	9.95e-05	> 0.01	> 0.01	1.34e-05	-	2.94e-04	2.22e-05
NSGA-II	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	-	> 0.01
SMS-EMOA	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	-

Table A.7: R2-indicator on R_ZDT4

	DE-SMS-C	DE-SMS-CD	DE-SMS-PD	GDE3-SMS-D	GDE3-D	SMS-EMOA(DE)	GDE3	DE-SMS-P	GDE3-SMS	NSGA-II	SMS-EMOA
DE-SMS-C	-	> 0.01	8.99e-39	8.52e-39	1.15e-110	1.50e-66	7.96e-27	1.95e-73	> 0.01	4.61e-96	4.46e-87
DE-SMS-CD	> 0.01	-	1.16e-30	1.10e-30	1.06e-104	9.07e-59	2.87e-19	6.77e-66	> 0.01	1.85e-89	4.37e-80
DE-SMS-PD	> 0.01	> 0.01	-	> 0.01	7.49e-66	1.75e-14	> 0.01	6.82e-21	> 0.01	5.70e-47	6.16e-36
GDE3-SMS-D	> 0.01	> 0.01	> 0.01	-	7.87e-66	1.83e-14	> 0.01	7.16e-21	> 0.01	6.02e-47	6.50e-36
GDE3-D	> 0.01	> 0.01	> 0.01	> 0.01	-	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01
SMS-EMOA(DE)	> 0.01	> 0.01	> 0.01	> 0.01	5.00e-38	-	> 0.01	> 0.01	> 0.01	2.16e-19	1.61e-10
GDE3	> 0.01	> 0.01	3.50e-04	3.42e-04	6.01e-77	3.31e-25	-	1.89e-32	> 0.01	6.65e-59	4.63e-48
DE-SMS-P	> 0.01	> 0.01	> 0.01	> 0.01	1.28e-30	> 0.01	> 0.01	-	> 0.01	3.69e-13	7.10e-06
GDE3-SMS	> 0.01	> 0.01	3.83e-31	3.63e-31	4.60e-105	3.06e-59	1.06e-19	2.35e-66	-	7.32e-90	1.64e-80
NSGA-II	> 0.01	> 0.01	> 0.01	> 0.01	4.71e-08	> 0.01	> 0.01	> 0.01	> 0.01	-	> 0.01
SMS-EMOA	> 0.01	> 0.01	> 0.01	> 0.01	3.35e-16	> 0.01	> 0.01	> 0.01	> 0.01	1.00e-03	-

Table A.8: ε-indicator on S_DTLZ3

	DE-SMS-C	DE-SMS-CD	DE-SMS-PD	GDE3-SMS-D	GDE3-D	SMS-EMOA(DE)	GDE3	DE-SMS-P	GDE3-SMS	NSGA-II	SMS-EMOA
DE-SMS-C	-	> 0.01	4.73e-25	6.35e-18	2.12e-90	2.58e-63	8.12e-08	5.50e-48	1.69e-55	9.16e-70	3.04e-57
DE-SMS-CD	> 0.01	-	2.44e-24	2.84e-17	8.90e-90	1.36e-62	2.33e-07	3.12e-47	9.33e-55	4.59e-69	1.66e-56
DE-SMS-PD	> 0.01	> 0.01	-	> 0.01	7.20e-54	9.84e-24	> 0.01	1.40e-10	1.66e-16	2.47e-30	4.91e-18
GDE3-SMS-D	> 0.01	> 0.01	> 0.01	-	1.75e-61	2.83e-31	> 0.01	1.04e-16	1.72e-23	4.14e-38	3.57e-25
GDE3-D	> 0.01	> 0.01	> 0.01	> 0.01	-	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01
SMS-EMOA(DE)	> 0.01	> 0.01	> 0.01	> 0.01	2.74e-16	-	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01
GDE3	> 0.01	> 0.01	2.92e-09	8.97e-05	5.38e-74	1.14e-44	-	4.70e-29	1.46e-36	1.60e-51	2.38e-38
DE-SMS-P	> 0.01	> 0.01	> 0.01	> 0.01	8.74e-31	6.54e-06	> 0.01	-	> 0.01	3.74e-10	4.18e-03
GDE3-SMS	> 0.01	> 0.01	> 0.01	> 0.01	1.72e-23	> 0.01	> 0.01	> 0.01	-	1.55e-05	> 0.01
NSGA-II	> 0.01	> 0.01	> 0.01	> 0.01	6.54e-11	> 0.01	> 0.01	> 0.01	> 0.01	-	> 0.01
SMS-EMOA	> 0.01	> 0.01	> 0.01	> 0.01	7.73e-22	> 0.01	> 0.01	> 0.01	> 0.01	1.16e-04	-

Table A.9: Hypervolume-indicator on S_DTLZ3

	DE-SMS-C	DE-SMS-CD	DE-SMS-PD	GDE3-SMS-D	GDE3-D	SMS-EMOA(DE)	GDE3	DE-SMS-P	GDE3-SMS	NSGA-II	SMS-EMOA
DE-SMS-C	-	> 0.01	5.96e-52	4.90e-57	2.73e-131	1.20e-92	6.59e-34	2.80e-88	2.14e-11	6.26e-117	1.53e-105
DE-SMS-CD	> 0.01	-	1.25e-44	8.47e-50	1.07e-126	1.41e-86	1.02e-26	4.84e-82	1.32e-06	8.30e-112	5.61e-100
DE-SMS-PD	> 0.01	> 0.01	-	> 0.01	4.84e-82	1.42e-29	> 0.01	1.81e-24	> 0.01	8.00e-62	3.91e-46
GDE3-SMS-D	> 0.01	> 0.01	> 0.01	-	1.67e-77	1.34e-24	> 0.01	1.03e-19	> 0.01	8.08e-57	6.03e-41
GDE3-D	> 0.01	> 0.01	> 0.01	> 0.01	-	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01
SMS-EMOA(DE)	> 0.01	> 0.01	> 0.01	> 0.01	2.67e-39	-	> 0.01	> 0.01	> 0.01	4.50e-18	2.04e-06
GDE3	> 0.01	> 0.01	3.00e-07	1.24e-10	5.83e-97	1.40e-47	-	2.94e-42	> 0.01	1.82e-78	8.17e-64
DE-SMS-P	> 0.01	> 0.01	> 0.01	> 0.01	1.25e-44	> 0.01	> 0.01	-	> 0.01	7.44e-23	1.01e-09
GDE3-SMS	> 0.01	> 0.01	8.65e-28	7.38e-33	1.87e-115	1.11e-71	5.94e-12	9.34e-67	-	3.18e-99	2.69e-86
NSGA-II	> 0.01	> 0.01	> 0.01	> 0.01	7.12e-10	> 0.01	> 0.01	> 0.01	> 0.01	-	> 0.01
SMS-EMOA	> 0.01	> 0.01	> 0.01	> 0.01	4.63e-23	> 0.01	> 0.01	> 0.01	> 0.01	4.57e-06	-

Table A.10: R2-indicator on S_DTLZ3

	DE-SMS-C	DE-SMS-CD	DE-SMS-PD	GDE3-SMS-D	GDE3-D	SMS-EMOA(DE)	GDE3	DE-SMS-P	GDE3-SMS	NSGA-II	SMS-EMOA
DE-SMS-C	-	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	1.13e-09	1.81e-14
DE-SMS-CD	> 0.01	-	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	2.92e-14	1.03e-19
DE-SMS-PD	2.38e-20	4.80e-15	-	3.64e-08	1.28e-08	> 0.01	3.25e-03	2.36e-03	8.10e-22	1.37e-41	5.05e-48
GDE3-SMS-D	7.20e-06	3.99e-03	> 0.01	-	> 0.01	> 0.01	> 0.01	> 0.01	9.27e-07	1.90e-22	1.51e-28
GDE3-D	1.69e-05	7.03e-03	> 0.01	> 0.01	-	> 0.01	> 0.01	> 0.01	2.33e-06	8.55e-22	7.39e-28
SMS-EMOA(DE)	4.63e-36	5.87e-30	2.88e-06	5.13e-21	1.15e-21	-	1.08e-12	5.58e-13	1.12e-37	8.42e-58	5.63e-64
GDE3	2.80e-12	5.19e-08	> 0.01	2.76e-03	1.48e-03	> 0.01	-	> 0.01	1.60e-13	6.30e-32	2.53e-38
DE-SMS-P	5.36e-12	8.85e-08	> 0.01	3.78e-03	2.07e-03	> 0.01	> 0.01	-	3.13e-13	1.48e-31	6.00e-38
GDE3-SMS	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	-	1.33e-08	3.44e-13
NSGA-II	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	-	> 0.01
SMS-EMOA	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	-

Table A.11: ε-indicator on SYM-PART

	DE-SMS-C	DE-SMS-CD	DE-SMS-PD	GDE3-SMS-D	GDE3-D	SMS-EMOA(DE)	GDE3	DE-SMS-P	GDE3-SMS	NSGA-II	SMS-EMOA
DE-SMS-C	-	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	2.33e-09	1.88e-14
DE-SMS-CD	4.31e-03	-	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	5.60e-03	1.73e-16	1.40e-22
DE-SMS-PD	6.85e-26	2.20e-17	-	2.08e-11	2.26e-13	> 0.01	6.57e-05	1.30e-04	1.39e-25	4.21e-47	6.41e-54
GDE3-SMS-D	1.56e-06	> 0.01	> 0.01	-	> 0.01	> 0.01	> 0.01	> 0.01	2.37e-06	3.83e-23	1.03e-29
GDE3-D	3.64e-05	> 0.01	> 0.01	> 0.01	-	> 0.01	> 0.01	> 0.01	5.25e-05	9.14e-21	3.38e-27
SMS-EMOA(DE)	9.92e-40	1.94e-30	4.51e-05	2.87e-23	1.04e-25	-	4.95e-14	1.57e-13	2.10e-39	6.79e-61	2.07e-67
GDE3	8.73e-14	2.89e-07	> 0.01	1.45e-03	1.13e-04	> 0.01	-	> 0.01	1.57e-13	2.15e-33	2.94e-40
DE-SMS-P	2.73e-14	1.20e-07	> 0.01	8.10e-04	5.66e-05	> 0.01	> 0.01	-	4.95e-14	5.04e-34	6.80e-41
GDE3-SMS	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	-	1.41e-09	1.03e-14
NSGA-II	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	-	> 0.01
SMS-EMOA	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	-

Table A.12: Hypervolume-indicator on SYM-PART

	DE-SMS-C	DE-SMS-CD	DE-SMS-PD	GDE3-SMS-D	GDE3-D	SMS-EMOA(DE)	GDE3	DE-SMS-P	GDE3-SMS	NSGA-II	SMS-EMOA
DE-SMS-C	-	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	3.11e-03	2.37e-14	5.65e-20
DE-SMS-CD	5.83e-03	-	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	1.23e-07	4.06e-22	2.26e-28
DE-SMS-PD	2.59e-07	4.83e-03	-	7.15e-06	9.65e-07	> 0.01	> 0.01	> 0.01	3.66e-14	6.56e-31	1.81e-37
GDE3-SMS-D	> 0.01	> 0.01	> 0.01	-	> 0.01	> 0.01	> 0.01	> 0.01	2.90e-04	1.86e-16	2.68e-22
GDE3-D	> 0.01	> 0.01	> 0.01	> 0.01	-	> 0.01	> 0.01	> 0.01	1.32e-03	3.82e-15	7.43e-21
SMS-EMOA(DE)	1.64e-30	5.81e-22	4.99e-14	5.10e-28	1.49e-29	-	8.12e-06	3.65e-11	3.35e-40	1.85e-58	8.67e-65
GDE3	3.33e-16	2.20e-09	3.10e-04	4.12e-14	2.17e-15	> 0.01	-	8.55e-03	6.05e-25	4.15e-43	1.04e-49
DE-SMS-P	1.02e-09	1.46e-04	> 0.01	4.79e-08	4.62e-09	> 0.01	> 0.01	-	2.82e-17	1.25e-34	3.09e-41
GDE3-SMS	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	-	1.87e-07	6.35e-12
NSGA-II	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	-	> 0.01
SMS-EMOA	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	-

Table A.13: R2-indicator on SYM-PART

	DE-SMS-C	DE-SMS-CD	DE-SMS-PD	GDE3-SMS-D	GDE3-D	SMS-EMOA(DE)	GDE3	DE-SMS-P	GDE3-SMS	NSGA-II	SMS-EMOA
DE-SMS-C	-	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	1.04e-11	3.39e-09	> 0.01
DE-SMS-CD	> 0.01	-	> 0.01	> 0.01	> 0.01	> 0.01	7.16e-03	> 0.01	3.25e-14	1.80e-11	> 0.01
DE-SMS-PD	4.27e-09	4.56e-07	-	> 0.01	2.18e-06	> 0.01	5.05e-13	3.85e-10	2.47e-30	7.25e-27	> 0.01
GDE3-SMS-D	1.94e-23	1.85e-20	6.12e-07	-	2.14e-19	8.86e-12	1.22e-28	6.97e-25	7.91e-48	2.95e-44	> 0.01
GDE3-D	> 0.01	> 0.01	> 0.01	> 0.01	-	> 0.01	2.72e-03	> 0.01	3.54e-15	2.33e-12	> 0.01
SMS-EMOA(DE)	6.44e-05	1.65e-03	> 0.01	> 0.01	4.53e-03	-	6.34e-08	1.10e-05	2.36e-23	4.37e-20	> 0.01
GDE3	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	-	> 0.01	5.55e-08	6.44e-06	> 0.01
DE-SMS-P	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	-	1.35e-10	3.37e-08	> 0.01
GDE3-SMS	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	-	> 0.01	> 0.01
NSGA-II	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	-	> 0.01
SMS-EMOA	7.06e-21	5.37e-18	2.00e-05	> 0.01	5.66e-17	8.66e-10	5.85e-26	2.78e-22	4.82e-45	1.84e-41	-

Table A.14: ε-indicator on M_S_DTLZ2

	DE-SMS-C	DE-SMS-CD	DE-SMS-PD	GDE3-SMS-D	GDE3-D	SMS-EMOA(DE)	GDE3	DE-SMS-P	GDE3-SMS	NSGA-II	SMS-EMOA
DE-SMS-C	-	> 0.01	> 0.01	> 0.01	2.29e-05	> 0.01	3.59e-04	> 0.01	3.81e-10	3.01e-03	> 0.01
DE-SMS-CD	> 0.01	-	> 0.01	> 0.01	3.50e-05	> 0.01	5.16e-04	> 0.01	6.86e-10	4.09e-03	> 0.01
DE-SMS-PD	> 0.01	> 0.01	-	> 0.01	2.23e-06	> 0.01	4.74e-05	> 0.01	1.63e-11	5.31e-04	> 0.01
GDE3-SMS-D	2.01e-09	1.13e-09	3.54e-08	-	3.24e-21	> 0.01	6.20e-19	7.78e-04	1.10e-28	6.20e-17	> 0.01
GDE3-D	> 0.01	> 0.01	> 0.01	> 0.01	-	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01
SMS-EMOA(DE)	3.39e-09	1.92e-09	5.77e-08	> 0.01	6.57e-21	-	1.23e-18	1.07e-03	2.34e-28	1.20e-16	> 0.01
GDE3	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	-	> 0.01	1.66e-03	> 0.01	> 0.01
DE-SMS-P	2.09e-03	1.51e-03	9.79e-03	> 0.01	8.57e-12	> 0.01	5.76e-10	-	3.28e-18	1.98e-08	> 0.01
GDE3-SMS	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	-	> 0.01	> 0.01
NSGA-II	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	1.78e-04	-	> 0.01
SMS-EMOA	3.53e-13	1.82e-13	9.98e-12	> 0.01	4.87e-26	> 0.01	1.25e-23	2.50e-06	9.14e-34	1.69e-21	-

Table A.15: Hypervolume-indicator on M_S_DTLZ2

	DE-SMS-C	DE-SMS-CD	DE-SMS-PD	GDE3-SMS-D	GDE3-D	SMS-EMOA(DE)	GDE3	DE-SMS-P	GDE3-SMS	NSGA-II	SMS-EMOA
DE-SMS-C	-	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	1.43e-03	3.16e-15	5.89e-13	> 0.01
DE-SMS-CD	> 0.01	-	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	4.33e-03	3.49e-14	5.62e-12	> 0.01
DE-SMS-PD	> 0.01	> 0.01	-	> 0.01	9.58e-03	> 0.01	> 0.01	8.59e-04	1.10e-15	2.18e-13	> 0.01
GDE3-SMS-D	5.76e-14	5.28e-15	1.59e-13	-	1.39e-20	5.49e-10	2.50e-09	3.31e-23	1.92e-41	1.39e-38	9.35e-11
GDE3-D	> 0.01	> 0.01	> 0.01	> 0.01	-	> 0.01	> 0.01	> 0.01	2.12e-09	1.42e-07	> 0.01
SMS-EMOA(DE)	> 0.01	> 0.01	> 0.01	> 0.01	1.25e-04	-	> 0.01	4.61e-06	8.22e-20	2.57e-17	> 0.01
GDE3	> 0.01	> 0.01	> 0.01	> 0.01	4.34e-05	> 0.01	-	1.35e-06	1.11e-20	3.77e-18	> 0.01
DE-SMS-P	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	-	1.44e-07	6.08e-06	> 0.01
GDE3-SMS	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	-	> 0.01	> 0.01
NSGA-II	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	-	> 0.01
SMS-EMOA	> 0.01	> 0.01	> 0.01	> 0.01	3.85e-04	> 0.01	> 0.01	1.73e-05	7.66e-19	2.18e-16	-

Table A.16: R2-indicator on M_S_DTLZ2

	DE-SMS-C	DE-SMS-CD	DE-SMS-PD	GDE3-SMS-D	GDE3-D	SMS-EMOA(DE)	GDE3	DE-SMS-P	GDE3-SMS	NSGA-II	SMS-EMOA
DE-SMS-C	-	> 0.01	6.85e-21	1.32e-63	1.14e-76	> 0.01	4.82e-39	4.42e-35	3.05e-87	3.77e-104	1.24e-103
DE-SMS-CD	> 0.01	-	1.52e-16	7.59e-59	3.43e-72	> 0.01	4.33e-34	3.54e-30	4.98e-83	2.21e-100	7.50e-100
DE-SMS-PD	> 0.01	> 0.01	-	2.64e-28	2.24e-42	> 0.01	1.20e-07	1.97e-05	1.54e-54	1.15e-74	4.79e-74
GDE3-SMS-D	> 0.01	> 0.01	> 0.01	-	3.90e-05	> 0.01	> 0.01	> 0.01	5.68e-13	2.80e-32	1.34e-31
GDE3-D	> 0.01	> 0.01	> 0.01	> 0.01	-	> 0.01	> 0.01	> 0.01	3.13e-04	6.74e-19	2.70e-18
SMS-EMOA(DE)	> 0.01	> 0.01	5.75e-16	3.53e-58	1.46e-71	-	2.09e-33	1.67e-29	1.95e-82	7.50e-100	2.56e-99
GDE3	> 0.01	> 0.01	> 0.01	7.42e-12	5.91e-24	> 0.01	-	> 0.01	7.32e-36	4.52e-57	2.12e-56
DE-SMS-P	> 0.01	> 0.01	> 0.01	5.98e-15	9.91e-28	> 0.01	> 0.01	-	7.91e-40	6.60e-61	3.03e-60
GDE3-SMS	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	-	2.66e-09	7.58e-09
NSGA-II	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	-	> 0.01
SMS-EMOA	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	-

Table A.17: ε-indicator on M_S_ZDT1

	DE-SMS-C	DE-SMS-CD	DE-SMS-PD	GDE3-SMS-D	GDE3-D	SMS-EMOA(DE)	GDE3	DE-SMS-P	GDE3-SMS	NSGA-II	SMS-EMOA
DE-SMS-C	-	> 0.01	3.14e-27	1.52e-76	9.80e-89	1.96e-06	1.68e-53	8.79e-47	5.94e-105	1.94e-118	5.30e-118
DE-SMS-CD	> 0.01	-	1.50e-21	4.49e-71	1.23e-83	1.46e-03	1.66e-47	9.94e-41	2.25e-100	2.74e-114	7.74e-114
DE-SMS-PD	> 0.01	> 0.01	-	1.84e-35	1.28e-49	> 0.01	5.14e-13	3.06e-08	1.83e-69	2.91e-86	1.02e-85
GDE3-SMS-D	> 0.01	> 0.01	> 0.01	-	3.98e-05	> 0.01	> 0.01	> 0.01	5.68e-20	2.80e-38	1.24e-37
GDE3-D	> 0.01	> 0.01	> 0.01	> 0.01	-	> 0.01	> 0.01	> 0.01	7.92e-09	2.05e-24	8.24e-24
SMS-EMOA(DE)	> 0.01	> 0.01	1.39e-12	4.17e-61	2.65e-74	-	6.98e-37	3.55e-30	5.57e-92	1.10e-106	3.30e-106
GDE3	> 0.01	> 0.01	> 0.01	1.62e-11	1.65e-23	> 0.01	-	> 0.01	9.00e-44	1.92e-62	7.93e-62
DE-SMS-P	> 0.01	> 0.01	> 0.01	6.69e-17	5.42e-30	> 0.01	> 0.01	-	1.58e-50	8.10e-69	3.22e-68
GDE3-SMS	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	-	8.34e-08	2.05e-07
NSGA-II	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	-	> 0.01
SMS-EMOA	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	-

Table A.18: Hypervolume-indicator on M_S_ZDT1

	DE-SMS-C	DE-SMS-CD	DE-SMS-PD	GDE3-SMS-D	GDE3-D	SMS-EMOA(DE)	GDE3	DE-SMS-P	GDE3-SMS	NSGA-II	SMS-EMOA
DE-SMS-C	-	> 0.01	3.74e-34	2.05e-86	8.29e-100	9.24e-13	1.35e-62	1.01e-53	1.59e-111	1.70e-129	2.28e-129
DE-SMS-CD	> 0.01	-	3.03e-29	3.67e-82	6.60e-96	3.49e-09	8.53e-58	8.71e-49	6.21e-108	2.35e-126	3.17e-126
DE-SMS-PD	> 0.01	> 0.01	-	8.89e-40	2.79e-56	> 0.01	7.61e-15	3.40e-08	2.86e-71	1.85e-94	2.70e-94
GDE3-SMS-D	> 0.01	> 0.01	> 0.01	-	2.10e-06	> 0.01	> 0.01	> 0.01	6.74e-18	5.95e-44	9.55e-44
GDE3-D	> 0.01	> 0.01	> 0.01	> 0.01	-	> 0.01	> 0.01	> 0.01	5.68e-06	1.75e-27	2.76e-27
SMS-EMOA(DE)	> 0.01	> 0.01	8.37e-11	6.23e-63	3.44e-78	-	8.76e-37	9.14e-28	1.29e-91	3.77e-112	5.27e-112
GDE3	> 0.01	> 0.01	> 0.01	4.95e-13	9.75e-28	> 0.01	-	> 0.01	2.01e-43	1.19e-69	1.85e-69
DE-SMS-P	> 0.01	> 0.01	> 0.01	9.40e-21	9.37e-37	> 0.01	5.06e-03	-	1.55e-52	6.26e-78	9.51e-78
GDE3-SMS	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	-	1.92e-13	2.78e-13
NSGA-II	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	-	> 0.01
SMS-EMOA	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	-

Table A.19: R2-indicator on M_S_ZDT1

	DE-SMS-C	DE-SMS-CD	DE-SMS-PD	GDE3-SMS-D	GDE3-D	SMS-EMOA(DE)	GDE3	DE-SMS-P	GDE3-SMS	NSGA-II	SMS-EMOA
DE-SMS-C	-	> 0.01	> 0.01	> 0.01	9.19e-04	> 0.01	> 0.01	2.59e-03	> 0.01	1.11e-18	5.22e-18
DE-SMS-CD	> 0.01	-	> 0.01	5.11e-04	1.23e-05	> 0.01	> 0.01	4.69e-05	> 0.01	2.49e-22	1.27e-21
DE-SMS-PD	> 0.01	> 0.01	-	> 0.01	6.17e-03	> 0.01	> 0.01	> 0.01	> 0.01	9.05e-17	4.03e-16
GDE3-SMS-D	> 0.01	> 0.01	> 0.01	-	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	2.23e-12	8.44e-12
GDE3-D	> 0.01	> 0.01	> 0.01	> 0.01	-	> 0.01	> 0.01	> 0.01	> 0.01	6.87e-10	2.30e-09
SMS-EMOA(DE)	5.52e-03	> 0.01	8.05e-04	1.81e-06	1.54e-08	-	1.39e-03	8.29e-08	> 0.01	4.98e-27	2.73e-26
GDE3	> 0.01	> 0.01	> 0.01	> 0.01	3.82e-03	> 0.01	-	9.52e-03	> 0.01	2.83e-17	1.28e-16
DE-SMS-P	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	-	> 0.01	1.06e-10	3.69e-10
GDE3-SMS	> 0.01	> 0.01	> 0.01	6.62e-03	3.06e-04	> 0.01	> 0.01	9.42e-04	-	1.11e-19	5.34e-19
NSGA-II	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	-	> 0.01
SMS-EMOA	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	-

Table A.20: ε-indicator on M_S_ZDT2

	DE-SMS-C	DE-SMS-CD	DE-SMS-PD	GDE3-SMS-D	GDE3-D	SMS-EMOA(DE)	GDE3	DE-SMS-P	GDE3-SMS	NSGA-II	SMS-EMOA
DE-SMS-C	-	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	1.20e-14	3.82e-14
DE-SMS-CD	> 0.01	-	> 0.01	7.91e-04	1.90e-05	> 0.01	> 0.01	2.23e-04	> 0.01	6.15e-21	2.26e-20
DE-SMS-PD	> 0.01	> 0.01	-	> 0.01	9.57e-04	> 0.01	> 0.01	6.52e-03	> 0.01	1.23e-17	4.23e-17
GDE3-SMS-D	> 0.01	> 0.01	> 0.01	-	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	1.40e-11	4.03e-11
GDE3-D	> 0.01	> 0.01	> 0.01	> 0.01	-	> 0.01	> 0.01	> 0.01	> 0.01	4.15e-09	1.07e-08
SMS-EMOA(DE)	5.67e-03	> 0.01	> 0.01	1.45e-04	2.39e-06	-	> 0.01	3.56e-05	> 0.01	1.73e-22	6.52e-22
GDE3	> 0.01	> 0.01	> 0.01	> 0.01	2.80e-03	> 0.01	-	> 0.01	> 0.01	1.31e-16	4.39e-16
DE-SMS-P	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	-	> 0.01	1.19e-10	3.29e-10
GDE3-SMS	> 0.01	> 0.01	> 0.01	> 0.01	4.08e-03	> 0.01	> 0.01	> 0.01	-	3.14e-16	1.04e-15
NSGA-II	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	-	> 0.01
SMS-EMOA	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	-

Table A.21: Hypervolume-indicator on M_S_ZDT2

	DE-SMS-C	DE-SMS-CD	DE-SMS-PD	GDE3-SMS-D	GDE3-D	SMS-EMOA(DE)	GDE3	DE-SMS-P	GDE3-SMS	NSGA-II	SMS-EMOA
DE-SMS-C	-	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	1.98e-14	7.90e-14
DE-SMS-CD	5.87e-03	-	> 0.01	4.68e-04	8.60e-06	> 0.01	> 0.01	3.38e-04	> 0.01	3.36e-22	1.65e-21
DE-SMS-PD	> 0.01	> 0.01	-	7.26e-03	2.81e-04	> 0.01	> 0.01	5.61e-03	> 0.01	2.28e-19	1.05e-18
GDE3-SMS-D	> 0.01	> 0.01	> 0.01	-	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	3.34e-12	1.22e-11
GDE3-D	> 0.01	> 0.01	> 0.01	> 0.01	-	> 0.01	> 0.01	> 0.01	> 0.01	1.37e-09	4.39e-09
SMS-EMOA(DE)	> 0.01	> 0.01	> 0.01	9.46e-04	2.06e-05	-	> 0.01	6.94e-04	> 0.01	1.59e-21	7.72e-21
GDE3	> 0.01	> 0.01	> 0.01	> 0.01	1.95e-03	> 0.01	-	> 0.01	> 0.01	1.40e-17	6.20e-17
DE-SMS-P	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	-	> 0.01	5.90e-12	2.13e-11
GDE3-SMS	> 0.01	> 0.01	> 0.01	> 0.01	3.35e-03	> 0.01	> 0.01	> 0.01	-	4.87e-17	2.12e-16
NSGA-II	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	-	> 0.01
SMS-EMOA	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	-

Table A.22: R2-indicator on M_S_ZDT2

	DE-SMS-CD$_N$	DE-SMS-PD$_N$	DE-SMS-CD	GDE3	DE-SMS-PD	GDE3$_N$	SMS-EMOA$_D$	NSGA-II	SMS-EMOA$_B$
DE-SMS-CD$_N$	-	> 0.01	1.51e-11	> 0.01	2.70e-08	3.68e-20	9.14e-31	2.74e-41	1.22e-48
DE-SMS-PD$_N$	> 0.01	-	1.76e-11	> 0.01	3.07e-08	4.40e-20	1.10e-30	3.31e-41	1.47e-48
DE-SMS-CD	> 0.01	> 0.01	-	> 0.01	> 0.01	1.17e-03	3.16e-10	3.44e-19	2.14e-26
GDE3	> 0.01	> 0.01	1.11e-07	-	4.72e-05	2.42e-15	1.65e-25	5.33e-36	1.85e-43
DE-SMS-PD	> 0.01	> 0.01	> 0.01	> 0.01	-	6.79e-06	9.80e-14	2.32e-23	9.29e-31
GDE3$_N$	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	-	4.11e-04	9.75e-11	4.96e-17
SMS-EMOA$_D$	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	-	5.84e-04	2.73e-08
NSGA-II	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	-	> 0.01
SMS-EMOA$_B$	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	-

Table A.23: ε-indicator on surface reconstruction

	DE-SMS-CD$_N$	DE-SMS-PD$_N$	DE-SMS-CD	GDE3	DE-SMS-PD	GDE3$_N$	SMS-EMOA$_D$	NSGA-II	SMS-EMOA$_B$
DE-SMS-CD$_N$	-	6.96e-05	1.14e-13	1.64e-27	1.01e-41	5.41e-58	2.39e-73	4.04e-86	2.38e-97
DE-SMS-PD$_N$	> 0.01	-	5.45e-05	9.22e-16	2.53e-29	3.30e-46	1.12e-62	1.71e-76	1.22e-88
DE-SMS-CD	> 0.01	> 0.01	-	3.03e-06	3.31e-17	1.12e-33	5.25e-51	6.73e-66	4.61e-79
GDE3	> 0.01	> 0.01	> 0.01	-	6.90e-06	3.72e-19	2.03e-36	2.34e-52	9.28e-67
DE-SMS-PD	> 0.01	> 0.01	> 0.01	> 0.01	-	1.40e-07	2.58e-22	1.91e-38	7.24e-54
GDE3$_N$	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	-	5.50e-08	1.23e-21	2.63e-37
SMS-EMOA$_D$	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	-	4.03e-07	5.54e-20
NSGA-II	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	-	7.54e-07
SMS-EMOA$_B$	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	-

Table A.24: Hypervolume-indicator on surface reconstruction

	DE-SMS-CD$_N$	DE-SMS-PD$_N$	DE-SMS-CD	GDE3	DE-SMS-PD	GDE3$_N$	SMS-EMOA$_D$	NSGA-II	SMS-EMOA$_B$
DE-SMS-CD$_N$	-	6.70e-03	9.96e-10	1.79e-17	6.57e-28	5.43e-41	2.35e-54	1.40e-65	2.96e-76
DE-SMS-PD$_N$	> 0.01	-	1.05e-04	9.60e-11	3.60e-20	5.11e-33	1.02e-46	1.99e-58	1.22e-69
DE-SMS-CD	> 0.01	> 0.01	-	1.96e-03	7.35e-10	3.82e-21	9.51e-35	5.26e-47	5.43e-59
GDE3	> 0.01	> 0.01	> 0.01	-	3.94e-04	8.21e-13	1.84e-25	8.68e-38	3.01e-50
DE-SMS-PD	> 0.01	> 0.01	> 0.01	> 0.01	-	3.00e-05	2.88e-15	6.60e-27	1.43e-39
GDE3$_N$	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	-	1.23e-05	1.24e-14	1.70e-26
SMS-EMOA$_D$	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	-	7.36e-05	1.10e-13
NSGA-II	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	-	5.02e-05
SMS-EMOA$_B$	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	> 0.01	-

Table A.25: R2-indicator on surface reconstruction

List of Figures

List of Algorithms

Bibliography

[1] Les Piegl and Wayne Tiller. *The NURBS Book*. Springer, New York, 1997.

[2] J. Mehnen. *Evolutionäre Flächenrekonstruktion*. PhD thesis, Universität Dortmund, 2000.

[3] Nicola Beume, Boris Naujoks, and Michael Emmerich. SMS-EMOA: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research*, 181(3):1653–1669, September 2007.

[4] Yongxiang Zhao, Shengwu Xiong, and Ning Xu. The geometry optimization of argon atom clusters using differential evolution algorithm. In *ICCS '07: Proceedings of the 7th international conference on Computational Science, Part IV*, pages 1155–1158, Berlin, Heidelberg, 2007. Springer.

[5] S. Kukkonen and J. Lampinen. GDE3: The third evolution step of generalized differential evolution. Technical report, Indian Institute of Technology, Kanpur, India, December 2005.

[6] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6:182–197, 2002.

[7] L. J. Fogel, A. J. Owens, and M. J. Walsh. *Artificial Intelligence through Simulated Evolution*. John Wiley, New York, USA, 1966.

[8] I. Rechenberg. Cybernetic solution path of an experimental problem. Technical report, Royal Air Force Establishment, 1965.

[9] H.-P. Schwefel. Kybernetische Evolution als Strategie der experimentellen Forschung in der Strömungstechnik. Diplomarbeit, Technical University of Berlin, 1965.

[10] John H. Holland. *Adaptation in natural and artificial systems*. MIT Press, Cambridge, MA, USA, 1992.

[11] Kalyanmoy Deb and Hans georg Beyer. Self-adaptive genetic algorithms with simulated binary crossover. *Complex Systems*, 9:431–454, 1995.

[12] Kalyanmoy Deb and Mayank Goyal. A combined genetic adaptive search (geneas) for engineering design. *Computer Science and Informatics*, 26:30–45, 1996.

[13] R. Storn and K. Price. Differential evolution - a simple and efficient adaptive scheme for global optimization over continuous spaces. Technical report, ICSI, 1995.

[14] R. Mendes and A. Mohais. Dynde: a differential evolution for dynamic optimization problems. In *IEEE Congress on Evolutionary Computation*, pages 2808–2815. IEEE, 2005.

[15] Kalyanmoy Deb. *Multi-Objective Optimization Using Evolutionary Algorithms*. Wiley, June 2001.

[16] Y. Liu, C. Zhou, and W.J. Ye. A fast optimization method of using nondominated sorting genetic algorithm (NSGA-II) and 1-nearest neighbor (1NN) classifier for numerical model calibration. In *Granular Computing, 2005 IEEE International Conference on*, volume 2, pages 544–549, July 2005.

[17] Kalyanmoy Deb, Udaya Bhaskara Rao N., and S. Karthik. Dynamic multi-objective optimization and decision-making using modified NSGA-II: A case study on hydrothermal power scheduling bi-objective optimization problems. Technical report, Indian Institute of Technology, Kanpur, India, 2006.

[18] Tarek M. Hamdani, Jin-Myung Won, Adel M. Alimi, and Fakhri Karray. Multi-objective feature selection with NSGA-II. In *ICANNGA '07: Proceedings of the 8th international conference on Adaptive and Natural Computing Algorithms, Part I*, pages 240–247, Berlin, Heidelberg, 2007. Springer.

[19] Mikkel T. Jensen. Reducing the run-time complexity of multiobjective eas: The NSGA-II and other algorithms. *IEEE Trans. Evolutionary Computation*, 7(5):503–515, 2003.

[20] Saku Kukkonen and Kalyanmoy Deb. A fast and effective method for pruning of non-dominated solutions in many-objective problems. In *PPSN*, pages 553–562, 2006.

[21] Eckart Zitzler and Lothar Thiele. Multiobjective optimization using evolutionary algorithms - a comparative case study. In *PPSN V: Proceedings of the 5th International Conference on Parallel Problem Solving from Nature*, pages 292–304, London, UK, 1998. Springer.

[22] Nicola Beume. Hypervolumen-basierte selektion in einem evolutionären algorithmus zur mehrzieloptimierung. Diploma thesis, University of Dortmund, 2006.

[23] J. Knowles. *Local-Search and Hybrid Evolutionary Algorithms for Pareto Optimization*. PhD thesis, Department of Computer Science, University of Reading, UK, 2002.

[24] Nicola Beume and Günter Rudolph. Faster s-metric calculation by considering dominated hypervolume as klee's measure problem. In Boris Kovalerchuk, editor, *Proceedings of the Second IASTED Conference on Computational Intelligence (CI 2006)*, pages 231–236. IASTED/ACTA Press, 2006.

[25] Carlos M. Fonseca, Luís Paquete, and Manuel López-Ibáñez. An improved dimension-sweep algorithm for the hypervolume indicator. In *IEEE Congress on Evolutionary Computation*, pages 1157–1163, Vancouver, Canada, July 2006.

[26] Rainer Storn. On the usage of differential evolution for function optimization. In *in NAFIPS'96*, pages 519–523. IEEE, 1996.

[27] K. Price. Differential evolution vs. the functions of the 2nd iceo. In *Proceedings of 1997 IEEE International Conference on Evolutionary Computation (ICEC '97)*, pages 153–157, April 1997.

[28] Saku Kukkonen and Jouni Lampinen. Performance assessment of generalized differential evolution 3 (GDE3) with a given set of problems. In *IEEE Congress on Evolutionary Computation*, pages 3593–3600, 2007.

[29] Tea Robic and Bogdan Filipic. Demo: Differential evolution for multiobjective optimization. In *EMO*, pages 520–533, 2005.

[30] B.V. Babu and Ashish M. Gujarathi. Elitist-multi-objective differential evolution (e-mode) algorithm for multi-objective optimization. In *Proceedings of 3rd Indian International Conference on Artificial Intelligence (IICAI-2007)*, pages 441–456, December 2007.

[31] Stefan Droste, Thomas Jansen, and Ingo Wegener. A rigorous complexity analysis of the (1+1) evolutionary algorithm for linear functions with boolean inputs. In *Proceedings of the IEEE International Conference on Evolutionary Computation (ICEC '98)*, pages 499–504, Piscataway, NJ, 1998. IEEE Press.

[32] Thomas Jansen and Ingo Wegener. On the analysis of evolutionary algorithms - a proof that crossover really can help. In *Proceedings of the 7th Annual European Symposium on Algorithms (ESA '99)*, pages 184–193, Berlin, Germany, 1999. Springer.

[33] Thomas Jansen and Ingo Wegener. On the choice of the mutation probability for the (1+1) EA. In *Proceedings of the 6th International Conference on Parallel Problem Solving From Nature (PPSN VI)*, pages 89–98, Berlin, Germany, 2000. Springer.

[34] V. L. Huang, A. K. Qin, K. Deb, E. Zitzler, P. N. Suganthan, J. J. Liang, M. Preuss, and S. Huband. Problem definitions for performance assessment on multi-objective

optimization algorithms. Technical report, Nanyang Technological University, January 2007.

[35] Tatsuya Okabe, Yaochu Jin, Markus Olhofer, and Bernhard Sendhoff. On test functions for evolutionary multi-objective optimization. In *Parallel Problem Solving from Nature (PPSN)*, Lecture Notes in Computer Science, pages 792–802. Springer, 2004.

[36] Günter Rudolph, Boris Naujoks, and Mike Preuss. Capabilities of emoa to detect and preserve equivalent pareto subsets. In *Evolutionary Multi-Criterion Optimization, 4th International Conference, EMO 2007, Matsushima, Japan, March 5-8, 2007, Proceedings*, volume 4403 of *Lecture Notes in Computer Science*, pages 36–50. Springer, 2007.

[37] Kalyanmoy Deb, Lothar Thiele, Marco Laumanns, and Eckart Zitzler. Scalable test problems for evolutionary multi-objective optimization. Technical report, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH), 2001.

[38] J. Knowles, L. Thiele, and E. Zitzler. A tutorial on the performance assessment of stochastic multiobjective optimizers. 214, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) Zurich, July 2005.

[39] K. Deb, M. Mohan, and S. Mishra. A Fast Multi-objective Evolutionary Algorithm for Finding Well-Spread Pareto-Optimal Solutions. KanGAL report 2003002, Indian Institute of Technology, Kanpur, India, 2003.

[40] P. Hansen and A. Jaszkiewiecz. Evaluating the quality of approximations to the non-dominated set. Technical report, Institute of mathematical modeling, Technical University of Denmark, Lyngby, 1998.

[41] Tobias Wagner, Thomas Michelitsch, and Alexei Sacharow. On the design of optimisers for surface reconstruction. In *GECCO '07: Proceedings of the 9th annual conference on Genetic and evolutionary computation*, pages 2195–2202, 2007.

[42] Ilijas Selimovic. Improved algorithms for the projection of points on nurbs curves and surfaces. *Comput. Aided Geom. Des.*, 23(5):439–445, 2006.

[43] Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, September 1936.

[44] W.H. Pres, S.A. Teukolsky, W.T. Vettering, and B.P. Flannery. *Numerical recipes in C++*. Cambridge University Press, 2002.

[45] E Achtert. Inkrementelles hierarchisches clustering. Diplomarbeit, Ludwig Maximilians Universität München, 2004.

[46] Robert C. Tryon and Daniel E. Bailey. *Cluster analysis*. McGraw-Hill, New York, 1970.

[47] John A. Hartigan. *Clustering algorithms*. Wiley, New York, 1975.

[48] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM Comput. Surv.*, 31(3):264–323, September 1999.

[49] Michiel de Hoon, Seiya Imoto, and Satoru Miyano. *The C Clustering Library*, August 2005.

[50] R. Sibson. Slink: An optimally efficient algorithm for the single-link cluster method. *Comput. J.*, 16(1):30–34, 1973.

[51] S. Kukkonen and K. Deb. Improved pruning of non-dominated solutions based on crowding distance for bi-objective optimization problems. Technical report, Indian Institute of Technology Kanpur, India, 2006.

[52] Thomas Bartz-Beielstein, Christian Lasarczyk, and Mike Preuß. Sequential parameter optimization. In B. McKay et al., editors, *Proc. 2005 Congress on Evolutionary Computation (CEC'05), Edinburgh, Scotland*, volume 1, pages 773–780, Piscataway NJ, 2005. IEEE Press.

[53] 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 (CEC 2006)*, pages 2002–2009, July 2006.

[54] S. Kukkonen and J. Lampinen. An empirical study of control parameters in generalized differential evolution. Technical report, Indian Institute of Technology, Kanpur, India, December 2005.

[55] Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation*, 8:173–195, 2000.

[56] M. D. McKay, R. J. Beckman, and W. J. Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. In *Technometrics, Vol. 21, No. 2*, pages 239–245, May 1979.

[57] B. T. Zhang and J. J. Kim. Comparison of selection methods for evolutionary optimization. In *Evolutionary Optimization*, pages 55–70, 2000.

[58] Kalyanmoy Deb and Ram B. Agrawal. Simulated binary crossover for continuous search space. Technical report, Departement of Mechanical Engineering, Indian Institute of Technology, Kanpur, India, 1994.

[59] K. Weinert, A. Zabel, P. Kersting, T.Michelitsch, and T. Wagner. On the use of problem-specific candidate generators for the hybrid optimization of multi-objective production-engineering problems. In *Evolutionary Computation*. MIT Press, to appear in 2010.

[60] E. Davies. *Machine Vision: Theory, Algorithms and Practicalities*. Academic Press, 1990.

[61] I. Rechenberg. *Evolutionsstrategie: optimierung technischer systeme nach prinzipien der biologischen evolution*. Frommann-Holzboog, 1973.