Integration of Expert’s Preferences in Pareto Optimization by Desirability Function Techniques

J. Mehnen\textsuperscript{1}, H. Trautmann\textsuperscript{2}

\textsuperscript{1}Department of Machining Technology, University of Dortmund, Germany
\textsuperscript{2}Department of Statistics, University of Dortmund, Germany

Abstract
Many real-world problems have a multiobjective character. A-posteriori techniques such as multiobjective evolutionary algorithms (MOEA) generate best compromise solution sets, i.e. Pareto-fronts and Pareto-sets. Classic MOEA are able to find quite efficiently good approximations of the complete Pareto-fronts also for very complex problems. In real-world applications only small sections of the complete front are of practical interest. Desirability functions are a flexible, intuitive and mathematically sound technique to focus on Pareto-front segments without losing the power of the a-posteriori population search methods. A systematic analysis of typical multiobjective evolutionary algorithms applied to a set of test functions and a real-world problem, the mold temperature control design problem, using the desirability function technique is presented.

Keywords:
Desirability Functions, Mold Temperature Control, Multiobjective Evolutionary Algorithms, Multiobjective Particle swarm Optimization

1 INTRODUCTION
Many technical optimization problems have to cope with several and often conflicting objectives. The aim of multiobjective optimization is to find parameter settings that correspond to "optimal" solutions. Such solutions are not necessarily unique and especially in conflicting situations only a set of compromise solutions is available if no a-priori knowledge about the user’s preferences is given. In general multiobjective optimization can be distinguished into three classes regarding at which stage preference of the decision maker is integrated. A-priori techniques utilize expert’s knowledge before the optimization process for guiding the solutions in a-priori known directions between the conflicting objectives. Methods such as desirability indices [1, 2], utility functions [3], aggregation schemes [4], or lexicographic ordering [4] generally yield a single compromise solution. Here, the user must have exact knowledge about the weighting of the objectives or the optimal relation between the different quality properties. In this case, the optimization problem – although often still difficult – can be solved by single-objective algorithms. Problems with many objectives or with unknown structure cannot be solved this way. A-posteriori approaches create sets of solutions from which a decision maker can choose. Pareto sets form a spectrum of best possible compromises, i.e. one objective cannot be improved without deteriorating at least one of the other objectives. Modern population based search algorithms that follow the a-posteriori approach try to approximate the complete Pareto front in a single run. The approximation also follows two objectives: the distribution of the solutions on the Pareto front should be broad and homogenous while the allower distance of the approximation to the real Pareto front should be as small as possible.

Experts have good experience and a lot of general knowledge about real-world problems. On the one hand problems can be so difficult that the best or just a good solutions cannot be given just by good intuition. Here efficient algorithms are needed to calculate the complete Pareto set in order to get an overview of the problem structure. On the other hand general solution properties can be given a-priori. For example specific areas of meaningful solutions such as costs or technical limitations can be characterized beforehand. A typical solution to find feasible solutions is to use penalty functions. These functions actually reduce the number of legal solutions and therefore decrease the search ‘power’ of a population based search algorithm. A focusing on desired areas would be more helpful.

A mixture of a-priori and a-posteriori approaches is realized by desirability functions (DF). Expert’s knowledge can be utilized to focus the search direction of a population based search algorithm on specific areas in the search space. Intuitive parameter settings and continuously adjustable desirability functions help the engineer characterizing the desired solution properties. Desirability functions do not just filter out solutions using strict restrictions. Their continuous character concentrate the search process and allow to use all individuals during a complete run. Desirability functions are transformations of the original objective space and, therefore, can be applied without change of the original functions.

Few concepts for integrating preferences into a-posteriori optimization exist. They focus mainly on a weighting of objectives rather than on individual objective preference regions, i.e. the objectives are seen to be not equally important [5].

This paper focuses on the application of desirability functions in combination with three population based metaheuristics. They are applied to two benchmark functions and one real-world problem. The mold temperature control (MTC) optimization problem is a highly difficult multiobjective real-world problem where a mixture of expert’s experience meets the demand for a flexible best compromise solution selection. The influence of DF on the convergence behavior of SPEA 2 and NSGA-II, the two mostly applied multiobjective evolutionary algorithms (MOEA) and MOPSO, a multiobjective particle swarm optimizer, is analyzed. A detailed description of the desirability functions is given. A short description of the benchmark functions and the real-world problem is presented. The experimental setup introduces the concrete parameter settings. The section ‘results’ gives an empirical overview over the characteristics of the functions and metaheuristics in combination with different desirability functions. The conclusions summarize the main topics of the article.
2 DESIRABILITY FUNCTIONS

The general multiobjective optimization problem with \( m \) objectives and \( n \) dimensions of the decision space is defined as follows [6]:

\[
D \subset \mathbb{R}^m, f_i : D \to \mathbb{R}, i = 1, \ldots, m
\]

\[
\min_{x \in D} (f_1(x), \ldots, f_m(x))^T
\]

In the following the transformation of \( x \) under \( f \) in the objective space is defined as \( Y := (f_1(x), \ldots, f_m(x))^T \). Without loss of generality minimization can be assumed because any component of the problem can be transformed from maximization to minimization by multiplication with \(-1\). In the following, minimization in a multiobjective domain is defined via Pareto optimality [4], i.e. \( x^* \) is Pareto optimal iff

\[
\exists x \in D : f_i(x) \leq f_i(x^*), i = 1, \ldots, m \quad \text{and} \quad f_j(x) < f_j(x^*) \quad \text{for some} \quad j \in \{1, \ldots, m\}.
\]

The desirability function (DF) is a nonlinear transformation of the objective space. This is realized by the desirability function \( d_i : Y \to [0, 1], i = 1, \ldots, m \) and the original function \( f_i, i = 1, \ldots, m \). Following Trautmann [7] and definition (1) the multiobjective desirability function aims at a minimization problem (in a Pareto-dominance sense) as follows:

\[
D \subset \mathbb{R}^m, E \subset \mathbb{R}^m, f : D \to Y, d_i : Y \to [0, 1], i = 1, \ldots, m
\]

\[
\min_{x \in E} (−d_1(f_1(x)), \ldots, −d_m(f_m(x)))^T
\]

A DF most generally is defined as a function \( d_i : Y \to [0, 1] \), specifying preferences in the objective space. Harrington [1] introduces two types of DF. The one-sided specification of a DF aims at a minimization problem while the two-sided specification of a DF reflects a target value problem. The one-sided DF uses a special form of the Gompertz curve, where the kurtosis of the function is determined by the solutions \( b_i^1 \) and \( b_i^0 \) of a system of two linear equations that require parameters \( y_i^{(1)}, y_i^{(2)} \in Y_i \) and \( d_i^{(1)}, d_i^{(2)} \in [0, 1] \). The DF maps the original fitness value of objective \( y_i = f_i(x) \in \mathbb{R} \) to \( d(y_i) \in [0, 1], i = 1, \ldots, m \).

The definition of the one-sided desirability function is:

\[
\begin{align*}
    b_i^1 & = -\frac{\log(-\log(d_i^{(2)})) + \log(-\log(d_i^{(1)}))}{y_i^{(2)} - y_i^{(1)}} \\
    b_i^0 & = -\log(-\log(d_i^{(1)})) - b_i y_i^{(1)} \\
    y_i' & = b_i^0 + b_i^1 y_i \\
    d(y_i) & = e^{-e^{-y_i'}}
\end{align*}
\]

The parameters \( d_i^{(1)}, d_i^{(2)} \) and \( y_i^{(1)}, y_i^{(2)} \) control the mapping of an objective \( Y_i \) to its respective desirability. A high desirability value \( d_i \) is near \( 1.0 \) and unwanted objective values are mapped to values near \( 0.0 \). The Gompertz curve interpolates the control parameters. Figure 1 shows an example of an one-sided DF for \( (d_i^{(1)}, y_i^{(1)}) = (0.7, 2) \) and \( (d_i^{(2)}, y_i^{(2)}) = (0.2, 7) \), respectively. In the example small objective values are more desirable than high values.

The two-sided desirability function focuses the Pareto front to a target region. Three parameters \( (LSL_i, USL_i, n_i) \), \( i = 1, \ldots, m \) per objective function are needed to describe this DF. The \( LSL_i \in \mathbb{R} \) (lower specification level) and \( USL_i \in \mathbb{R} \) (upper specification level) characterize a symmetric desirability around a target value in the middle between both parameters. The third parameter \( n_i \in \mathbb{R}_+ \setminus \{0\} \) defines the shape of the two-sided desirability function.

The definition of the two-sided desirability function is:

\[
\begin{align*}
    y_i' & = \frac{2y_i - (USL_i + LSL_i)}{USL_i - LSL_i} \\
    d(y_i) & = e^{-|y_i'|^{n_i}}
\end{align*}
\]

Figure 2 shows three examples of two-sided desirability functions for \( (LSL_i, USL_i) = (3.0, 7.0) \) and \( n = 1, 2.5, 0.5 \), respectively. It holds that \( d(LSL_i) = d(USL_i) = 1/e \). The target value \( Y_i = (USL_i + LSL_i)/2 \) has the desirability 1.0. A disadvantage of the two-sided DF is that the function cannot be inverted because it is not monotonic. This implies that the optimization algorithms have to store both \( d(f_i) \) and \( f_i \). The algorithms approximate the Pareto front using \( d(f_i) \). The respective \( f_i \) of the transformed Pareto front can be plotted and analyzed in the original decision space. An advantage is that the symmetric peak structure allows – especially for small \( n \) – a very precise modeling of small regions with high continuous desirability. The intuitive setting of the limit values USL and LSL simplifies the decision maker’s choice of the target value and the region of interest.
### 3 Optimization Problems

There exist several classes of multiobjective problems [6]. Looking only at the shape of the Pareto fronts, one can distinguish convex and concave types. The concave type is difficult for algorithms that use simple a-priori aggregation schemes that just calculate a weighted sum of the fitness function values. Current a-posteriori population based algorithms are able to cover the complete Pareto front independently of its shape. In order to analyze the effect of desirability functions, one convex and one concave benchmark function are used. The mold temperature control problem is an example of a very difficult real-world problem. In the following a very short description of all three problems will be given.

The definition of the convex **Binh (1)** [8] problem is:

\[
f_1(x_1, x_2) = x_1^2 + x_2^2 \\
f_2(x_1, x_2) = (x_1 - 5)^2 + (x_2 - 5)^2 \\
\text{with} \\
-5 \leq x_1, x_2 \leq 10
\]

(6)

The definition of the concave **Fonseca (2)** [9] problem is shown below. In the tests \( n = 2 \) was used to simplify the comparison with the results with Binh (1).

\[
f_1(x) = 1 - \exp \left( -\sum_{i=1}^n (x_i - \frac{1}{\sqrt{n}})^2 \right) \\
f_2(x) = 1 - \exp \left( -\sum_{i=1}^n (x_i - \frac{1}{\sqrt{n}})^2 \right) \\
\text{with} \\
-4 \leq x_i \leq 4
\]

(7)

In order to simplify readability Binh (1) and Fonseca (2) are abbreviated to Binh and Fonseca, respectively. The Pareto fronts and the corresponding Pareto sets of the Binh and the Fonseca problems are shown in Figure 3. The Pareto optimal points in the decision space are indicated with small crosses.

The **mold temperature control problem (MTC)** is a high dimensional and highly restricted real-world problem. The general aim of an MTC is to find a bore structure that optimizes all three objectives a) an intensive local cooling of the die surface, b) a good global temperature distribution in the tool, and c) minimal manufacturing costs. A bore structure is modeled via interconnected lines in three dimensions. The bores must be within the mold and must not intersect the die surface nor any other bore. A very fast analyzer [10] was applied that estimates the cooling effects, checks for intersections, and calculates the manufacturing costs of a complete bore structure. The exact formulas are described in [6]. Figure 4 shows an example of a bore structure drawn from the corresponding Pareto front.

Figure 4: A mold temperature control design (left). A Pareto front (right) of the MTC problem generated with MOPSO after 150 generations.

### 4 Experimental Setup

The parameter of the desirability functions in the experiments with the Binh and the Fonseca problems have been set to focus on values that correspond to experiments with the MTC problem. The numbers given in the Binh experiments follow [11]. All MOEA experiments have been performed at least five times. The figures show the best typical solutions found.

Experiment Binh 1a (\( \epsilon = 10^{-2} \)), Binh 1a’ (\( \epsilon = 10^{-4} \)):

\[
y_1^{(1)} = 0.0, d_1^{(1)} = 1.0 - \epsilon, y_2^{(1)} = 50.0, d_2^{(1)} = \epsilon \\
y_1^{(2)} = 0.0, d_1^{(2)} = 1.0 - \epsilon, y_2^{(2)} = 5.0, d_2^{(2)} = \epsilon
\]

Experiment Binh 2c (\( \epsilon = 10^{-2} \))

\[
y_1^{(1)} = 20.0, d_1^{(1)} = 0.9, y_2^{(1)} = 50.0, d_2^{(1)} = \epsilon \\
y_1^{(2)} = 0.0, d_1^{(2)} = 1 - \epsilon, y_2^{(2)} = 10.0, d_2^{(2)} = \epsilon
\]

The dimension of the Fonseca objective space lies in \([0,1] \times [0,1] \). The corresponding one-sided desirability function values are:

\[
y_1^{(1)} = 0.4, d_1^{(1)} = 0.9, y_2^{(1)} = 1.0, d_2^{(1)} = 0.01 \\
y_1^{(2)} = 0.0, d_1^{(2)} = 0.99, y_2^{(2)} = 0.2, d_2^{(2)} = 0.01
\]

The parameters used for a mixed combination of one-sided and two-sided desirability functions applied to Fonseca’s problem are:

\[
y_1^{(1)} = 0.0, d_1^{(1)} = 0.99, y_2^{(1)} = 1.0, d_2^{(1)} = 0.01 \\
LSL_1 = 0.3, USL_2 = 0.7, n = 2.0
\]

MTC problem (focus on intensive cooling):

\[
y_1^{(1)} = 0.0, d_1^{(1)} = 0.99, y_2^{(1)} = 20, d_2^{(1)} = 0.01 \\
y_1^{(2)} = 0.0, d_1^{(2)} = 0.99, y_2^{(2)} = 50, d_2^{(2)} = 0.01
\]

The parameters of the algorithms NSGA-II, SPEA 2 and MOPSO are shown in table 1:

<table>
<thead>
<tr>
<th>NSGA-II</th>
<th>SPEA 2</th>
<th>MOPSO</th>
</tr>
</thead>
<tbody>
<tr>
<td>pop. size</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>generation</td>
<td>150</td>
<td>150</td>
</tr>
<tr>
<td>p_c</td>
<td>0.8</td>
<td>0.8</td>
</tr>
<tr>
<td>p_m</td>
<td>0.1</td>
<td>0.2</td>
</tr>
<tr>
<td>dist. crossover</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>dist. crossover</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>front ratio</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>selection type</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>crossover type</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>C1</td>
<td>2.0</td>
<td>2.0</td>
</tr>
<tr>
<td>C2</td>
<td>2.0</td>
<td>2.0</td>
</tr>
<tr>
<td>inertia</td>
<td>0.6</td>
<td>0.6</td>
</tr>
</tbody>
</table>

Table 1: Optimization algorithm parameters
5 MULTIOBJECTIVE EVOLUTIONARY OPTIMIZATION

The multiobjective optimization algorithms NSGA-II [12] and SPEA 2 [13] have been tested using the KEA toolbox [14]. MOPSO was implemented separately following [15]. The algorithms were programmed according to the literature.

6 RESULTS

6.1 Experiments with the Binh 1a/1a’ problems

The effect of the one-sided desirability functions of type 1a and 1a’ is to set an upper limit to the $f_2$ values of the Pareto front. This can be done by varying the parameter $y_2^{(1)}$. The Binh problems with parameter setting 1a or 1a’ mainly differ in the exactness of the parameter settings.

The limitation abilities of the desirability functions strongly depend on an exact parameter setting. This is due to the continuous and smooth character of the continuous DF. In Binh 1a’ very small $\epsilon = 10^{-6}$ in $d_2^{(2)} = 1.0 - \epsilon$ and $d_2^{(2)} = \epsilon$, respectively, cause a quite sharp limitation to values under $y_2^{(1)} = 5$. In Binh 1a using a comparatively large $\epsilon$ some values $5.0 < f_2 < 10.0$ are mapped by the desirability function only near to 0.0, and, therefore, are ‘accepted’ to belong to the desired Pareto front. The resulting Pareto front has several $f_i$-values that are larger than 5.0 (see Figure 5, left). Using very small $\epsilon$ yields desirability functions with sharp edges and more exact results (Figure 5, right). The decision to apply smooth or more “crisp” characteristics of the DF depends on the application.

Figure 5: Pareto front for Binh 1a (left) and 1a’ (right).

The evolutionary algorithms generally find more solutions in the desired region. All algorithms show good results even when applied to Binh 1a (see Figure 6).

Figure 6: Application of MOPSO (left) and SPEA 2 (right) to the Binh 1a problem.

6.2 Experiments with the Binh 2c problem

Experiment Binh 2c illustrates the differences of the MOEA when applied to a single problem. The mapping of a regular grid shows the desirability function effect (Figure 7, left). The parameter adjustment of the DF defines the shape of the Pareto front in the $d(f_i)$-space. The evolutionary algorithms approximated the Pareto fronts in the $d(f_i)$-space. A special property of one-sided desirability functions guarantees that the corresponding $f_i$-values also from a Pareto front [11]. This Pareto front can also be calculated via the inverse function of the DF or, better, by saving the corresponding $f_i$ values in the evolutionary algorithm. The application of the inverse should only be used, if the optimization algorithm cannot be changed. The inverse behaves not always numerically stable because undesired as well as strongly desired values are mapped to ‘numerically’ the same 0.0 or 1.0 values, respectively, implicating that these values cannot be inverted. A third alternative is to save the Pareto set solutions and to re-calculate the $f_i$. This approach can only be suggested for fast evaluations of the fitness functions. In real-world applications this assumption is not always realistic.

Figure 7: Mapping of a grid in the objective space into the $d(f_i)$-space (dots) and corresponding Pareto front (crosses) for Binh (Experiment 2c) (left). Corresponding focused Pareto front in the decision space (right).

Figures 8 to 10 show typical examples of the results of the different stochastic optimization algorithms. The approximation quality of the NSGA-II Pareto fronts is very good. Figure 8 (left) shows the good approximation to the $d(f_i)$-Pareto front of the last population of the NSGA-II after 150 generations. Because the NSGA-II algorithm was not changed, the inverse of the desirability functions are used. In Figure 8 (right) a ‘shadow effect’ can be seen in the upper part of the graph because not all individuals of the population approximate the $d(f_i)$-Pareto front perfectly.

Figure 8: Application of NSGA-II to Binh, Exp. 2c, with Pareto front in the $d(f_i)$-space (left) and corresponding Pareto front in the $f_i$-space (right).

Due to the fact that the maximum number of generations per run was limited to 150, the relatively slow SPEA 2 was not able to find sufficiently good solutions. Having more time, SPEA 2 often outperforms NSGA-II or MOPSO. SPEA 2 is well known for its good distribution properties that can be seen clearly in Figure 9 although here only few Pareto-optimal solutions have been found.

Figure 9: Application of SPEA 2 to Binh, Exp. 2c, with Pareto front in the $d(f_i)$-space (left) and corresponding Pareto front in the $f_i$-space (right).
The particle swarm optimizer shows very good approximation results after 150 generations. A problem of MOPSO – especially in its older version – is that it does not always have a stable behavior. The solutions tend to focus to a single point on the Pareto front. The MOPSO [15] used here shows a very robust behavior under the application of desirability functions. This is also true for the application of two-sided DF.

Figure 10: Application of MOPSO to Binh, Exp. 2c, with Pareto front in the \( d(f_i) \)-space (left) and corresponding Pareto front in the \( f_i \)-space (right).

6.3 Experiments with Fonseca’s problem

Fonseca’s problem is a concave test problem with two objectives. In the first experiment using this function two one-sided DF have been applied with parameter setting similar that of the Binh experiments. Figure 11 shows that the desired region of the Pareto front can be calculated independently from the shape of the front. This corresponds to the theorem in [11]. The solution is also numerically stable.

Figure 11: Pareto front of Fonseca’s problem in the \( d(f_i) \)-space using two one-sided desirability functions (left) and corresponding Pareto front (right).

Figure 12 shows the results of an experiment with a mixed strategy of an one-sided and a two-sided DF applied to Fonseca’s problem. The limit of the desired region of 0.5 < \( f_0 < 1.0 \) can be adjusted very precisely with the two-sided DF, which was applied to the second criterion. Although the two-sided desirability functions do not have the property that any Pareto front in the \( d(f_i) \)-space maps to a corresponding Pareto front in the \( f_i \)-space, the two-sided DF technique is advantageous in practice because of its very intuitive parameter adjustment and exact response behavior.

Figure 12: Pareto front in the \( d(f_i) \)-space (left) using a mix of one-sided and two-sided desirability functions and corresponding Pareto front of Fonseca’s problem (right).

6.4 Experiments with the MTC problem

In the following application the optimization of the mold temperature control problem is approached with the MOPSO algorithm. General analyses of the MTC using NSGA-II and evolution strategies with aggregation approaches have been published in e.g. [6]. In these analyses DFs are not used. Here the experiments focus on the application of the new MOPSO algorithm. Figure 13 shows the results of a typical run of MOPSO with and without an application of DF. In the following the analyses focus on the practical application of the one-sided desirability functions.

Figure 13: Pareto front of the complete MTC problem without application of DF (crosses). Pareto front focusing on high quality cooling (blue stars). Small cooling and cost values are generally preferred.

Due to the fact that the optimization of the abstract costs (here mainly the length of the global structure) is an easier problem than the minimization of the aggregated temperature criterion, all evolutionary algorithms tend to minimize the cost problem first. This implies that the complete Pareto front first shows solutions in areas with small costs. SPEA 2 and MOPSO try to spread the approximated Pareto front along the complete Pareto front. After some generations the complete front is covered by solutions. The desirability function can support this process by focusing on regions with good temperature qualities. In Figure 13 also solutions with very good (small) temperature values are found.

The desirability functions focus the complete search process towards the favored part of the Pareto front. No individuals are “wasted” by the use of penalty functions. Therefore, DF have a lot of capacity for increasing the approximation speed of the MOEA.

Figure 14: MTC design with best costs (left), i.e. short bores, and best good cooling properties (right). Both solutions are “best” with respect to extremal solutions taken from the Pareto front. The front is generated using desirability functions.

The desirability function approach is a mixture of an a-priori and an a-posteriori technique. It is useful to run a conventional a-posteriori algorithm first to generate a rough Pareto front approximation in order to scan the dimensions of the complete front. It is recommended to look at the Pareto front as well as at the Pareto sets and the respective phenotypes, i.e. here the bore structures. The expert will then be able to address certain areas of interest.
The general choice of the parameters of the DF is quite intuitive. The adjustment of the specific parameters is sensitive to small changes because of the strong effect of the exponents in the desirability functions. Large values generated by the MTC simulator have been rescaled for the experiments. This was necessary because of numerical problems caused by the exponents in the DFs. Especially problems that use penalty functions may cause problems if the penalty functions map infeasible solutions to high values.

Figure 13 shows that DFs have been used very efficiently in the MTC problem. Extreme solutions from the desired Pareto front (dark line) are chosen. Very small costs result in short bores while a good cooling implies long bores and an uniform approximation of the die surface. The solutions are typical examples.

The general behavior of optimization algorithm MOPSO is not influenced by the desirability functions. Two one-sided DF were used. Therefore, the inverse DF functions have been applied to find the respective desired Pareto front in the objective space. The phenotype descriptions of the solutions shown in Figure 14 were found by taking the Pareto set solutions and rendering the corresponding bore designs.

7 CONCLUSIONS

Desirability functions are a convenient way for integrating expert's knowledge into multiobjective search processes. One-sided as well as two-sided functions allow a focusing to regions of interest on a Pareto front. A-priori expert knowledge is needed to be able to adjust the parameters that guide the optimization search toward the area of interest. Experiments with benchmark functions show that the shape of the Pareto front does not influence the numerical behavior of the DF. An accurate choice of the parameters of the DF and a rescaling of the input values is necessary to get high quality results. The real-world application shows that even for very complex problems the desirability function approach works very well. DF functions can be used to support a more efficient evolutionary search in multiobjective problems with unbalanced complexity of the single objectives.

8 ACKNOWLEDGMENTS

This research is a result of a collaboration between project C12 of the collaborative research centers SFB 531 and the Graduate School of Statistical Modelling and the chair of computer aided statistics (Computergestützte Statistik) at the University of Dortmund, Germany. The SFB 531 and the Graduate School are funded by the German Research Foundation (DFG). Many thanks to Torsten Lickfeld, who contributed the MOPSO optimization of the MTC systems. We also like to thank Thomas Michelitsch who is developing the MTC.

REFERENCES


