# Sequential Multi-Objective Target-Value Optimization

**Dissertation**

in fulfillment of the requirements for the degree of
“Doktor der Naturwissenschaften”

submitted to the Department of Statistics,
Technische Universität Dortmund

by

**Simone Wenzel**

Dortmund, November 2011

1. Supervisor:

Prof. Dr. J. Kunert, Technische Universität Dortmund

2. Supervisor:

Prof. Dr. C. Weihs, Technische Universität Dortmund

Date of oral examination:

23. September 2011

# Acknowledgements

Though only my name appears on the cover of this dissertation, this work would not have been possible without the help, support and guidance of a lot of people.

I am heartily thankful to my supervisor, Joachim Kunert, who convinced me to start this thesis some years ago. I have been fortunate to have an advisor who gave me the freedom to explore on my own and having the perfect timing to show me the light at the end of the tunnel whenever I was giving up. I am also very grateful to my co-supervisor, Claus Weihs, for mediating humorously between the theoretical world of Joachim Kunert and my applied research.

I would like to thank Paul Schmelzer, Lukas Kwiatkowski and Oliver Melsheimer for their help and assistance with the case studies. My thanks also go to Silke Straatmann and Robin Nunkesser for their cooperation concerning the development and implementation of the double description method. Further, it is a great pleasure to thank the whole faculty of the Department of Statistics for the friendly atmosphere and for the probably most family-friendly working environment in the world. I will never forget all the humorous lunch and cake breaks.

A special thank you goes to my family and all my friends that have helped me to stay sane through this stressful time. At the same time they helped me to stay focused on my studies, and not to forget the other important things in life. I deeply appreciate their belief in me. Most importantly, I wish to express my deepest gratitude to my beloved Sven. Just let me say “immer zweimal mehr wie du”, you are my perfect match. I also would like to say sorry to my wonderful children Sarah and Lars for often having been so busy when you would rather have liked to play or cuddle. I love you.

Finally, I appreciate the financial support from the “Graduate School of Production Engineering and Statistics” and the “Deutsche Forschungsgesellschaft” (project DFG-SFB 475 and DFG-SFB 823).

# Contents

# Glossary

## Abbreviations

| | |
|---|---|
| $A_{Mm}$ | Maximin design |
| $A_{mM}$ | Minimax design |
| $A_{unif}$ | Uniform coverage design |
| AIC | Akaike criterion evaluating the goodness of fit of a model |
| BIC | Schwarz criterion evaluating the goodness of fit of a model |
| BLUP | Best linear unbiased predictor |
| CNC | Computer numerical control |
| DI | Desirability index |
| DD pair | Double description pair |
| EA | Evolutionary algorithm |
| EGO | Efficient global optimization algorithm by Jones et al. (1998) |
| EI | Expected improvement |
| LHD | Latin-Hypercube design |
| ML | Maximum Likelihood |
| MSCVE | Mean squared cross-validation error |
| MSPE | Mean squared prediction error |
| mtEGO | Multi-objective target-value efficient global optimization algorithm |
| mtEGOimp | Improved version of mtEGO |
| PCC | Polyhedral convex cone |
| REML | Restricted maximum likelihood |
| SCVR | Standardized cross-validated residual |
| SDO | Sequential design optimization by Cox and John (1997) |

## Constants, variables, functions

| | |
|---|---|
| $a$ | Number of $\alpha$ levels used in mtEGO |
| $A = x_1, .., x_n$ | Design with $n$ points |
| $b$ | Number of objectives in the optimization problem |
| $b_0, b_1$ | Specification parameters of the one-sided Harrington desirability function |
| $C$ | Set of suggested candidates in mtEGO |
| $C^*$ | Set of suggested candidates reduced by $x^*$ |
| $d(y)$ | Desirability function of objective $y$ |
| $d_1(.), .., d_b(.)$ | Desirability functions belonging to the $b$ objectives $y_1, .., y_b$ |
| $d_1, d_{11+}, d_{12-}, ...$ | Desirabilities of the virtual observations |
| $DI(Y) = DI(y_1, ..., y_b)$ | Desirability index of objectives $y_1, ..., y_b$ |
| $DI_{max}$ | Maximum desirability index within the current observations |
| $D(G_a, G_b)$ | Distance between two clusters $G_a, G_b$ |
| $EI = E[I(x)\|y^{(n)}]$ | Expected improvement |
| $E[I^g(x)\|y^{(n)}]$ | Generalized expected improvement |
| $E[I^g(x)\|y_1^{(n)}, ..., y_b^{(n)}]$ | Multivariate expected improvement |
| $f(d)$ | Distribution of a desirability function |
| $F(x)$ | Distribution function in point $x$ |
| $F_n(x)$ | Empirical distribution function in point $x$ |
| $f_1(.), ..., f_p(.)$ | Known regression functions in the Kriging model |
| $f_{Y(x)\|y^{(n)}}$ | Distribution of the uncertainty of the model for objective $Y(x)$ |
| $f_{DI(x)\|y_1^{(n)},...,y_b^{(n)}}$ | Conditional density of the desirability index $DI(x)$ given the observations $y_1^{(n)}, ..., y_b^{(n)}$ |
| $g$ | Factor in the generalized EI to weigh between local and global search |
| $G = \{x_1, ..., x_q\}$ | Group / cluster of $q$ points from $C^*$ in mtEGO |
| $h = x_i - x_j$ | Distance between two points $x_i$ and $x_j$ |
| $I(x)$ | Improvement function |
| $I_M^g(x)$ | Multivariate improvement function |

| | |
|---|---|
| $l$ | Specification parameter of the Derringer-Suich desirability function |
| $LSL$ | Lower specification limit in the two-sided desirability functions |
| $MSPE(\hat{y}(x_0))$ | MSPE for the BLUP of an unknown point $x_0$ in the Kriging model |
| $n$ | Number of design points and number of observations respectively |
| $p$ | Number of influencing parameters / dimension of parameter space |
| $q$ | Number of grid points representing $\chi$ |
| $r$ | Specification parameter of the Derringer-Suich desirability function |
| $R = Corr[Z(x_i), Z(x_j)]$ | Correlation matrix of the stochastic component $Z(x)$ in the Kriging model |
| $R[h;\theta]$ | Power exponential correlation function |
| $s$ | Parameter affecting the smoothness of $R[h;\theta]$ |
| $s^2_{\hat{y}}(x_0)$ | Short notation for $MSPE(\hat{y}(x_0))$ |
| $t_{1-\alpha/2,df}$ | $(1-\alpha/2)$-quantile of the t-distribution with $df$ degrees of freedom |
| $T$ | Target value of an objective |
| $[T_{-0.25}, T_{+0.25}]$ | Decision interval around the target $T$ in mtEGOeff |
| $USL$ | Upper specification limit in the two-sided desirability functions |
| $x_i$ | $i$-th sampled design point from $\chi$ |
| $\mathbf{X} = (x_1, ..., x_l)$ | Even set of $l$ grid points representing $\chi$ |
| $x^*$ | Point $x \in \chi$ with best predictions in the currently fitted model |
| $x_{cent}$ | Center point of a cluster / group that is used as updating point in mtEGO |
| $x_{curopt}$ | Design point $x \in A$ with currently observed optimum |
| $x_{NA}$ | Design point $x$ that leads to a failure point and hence yields missing response value |
| $x_{opt}$ | Parameter setting from $\chi$ with optimum DI |

| | |
|---|---|
| $y, Y(x)$ | Single-objective, quality characteristic |
| $y(x_i)$ | Univariate response for point $x_i$ |
| $\mathbf{Y}(x) = (y_1(x), ..., y_b(x))'$ | b-dimensional objective |
| $\mathbf{Y}(x_0) = (y_1(x_0), ..., y_b(x_0))'$ | Multivariate response for point $x_0$ |
| $Y^{(n)} = (Y(x_1), .., Y(x_n))'$ | $n$-dimensional normal distributed random vector |
| $y^{(n)} = (y(x_1), .., y(x_n))'$ | Vector of $n$ univariate observations, realization of $Y^{(n)}$ |
| $\hat{y}(x_0)$ | BLUP for an unknown point $x_0$ given the observations $y^{(n)}$ in the Kriging model |
| $\hat{y}, s^2$ | Short notation of the vector of predictions and prediction errors for the whole parameter space |
| $y_{min}, y_{max}$ | Currently observed minimum / maximum value of objective $y$ |
| $y_{curopt} = y(x^*)$ | Currently optimum value of the objective |
| $y_{imputed}$ | Imputation value that is subset for the missing response $y(x_{NA})$ |
| $y_{+dec\alpha}, y_{-dec\alpha}$ | Reference confidence boundaries in mtEGOeff |
| $\tilde{y}_1, \tilde{y}_{11+}, \tilde{y}_{12-}, ....$ | Virtual observations |
| $Z(x_i)$ | Stochastic component in the Kriging model in point $x_i$ |
| $\alpha = (\alpha_1, ..., \alpha_n)$ | Vector of confidence levels used for mtEGO |
| $\beta = (\beta_1, ..., \beta_p)$ | Unknown regression coefficients in the Kriging model |
| $\hat{\beta}$ | ML estimators of $\beta$ |
| $\chi$ | $p$-dimensional parameter space $\subset \mathbb{R}^p$ |
| $\mu(x_i)$ | Deterministic trend component in the Kriging model in point $x_i$ |
| $\phi, \Phi$ | Density and distribution function of the standard normal distribution |
| $\rho_p(x_i, x_j)$ | $p^{th}$ order distance (Minkowsky metric) between two points $x_i$ and $x_j$ |
| $\rho_p(x, A)$ | $p^{th}$ order distance (Minkowsky metric) between a point $x$ and a set $A$ |
| $\rho(x_i, x_j)$ | Mahalanobis distance between two points $x_i$ and $x_j$ |
| $\sigma_Z^2$ | Variance of the stochastic component $Z(x)$ in the Kriging model |
| $\hat{\sigma}_Z^2$ | ML estimator of $\sigma_Z^2$ |

| | |
|---|---|
| $\tilde{\sigma}_Z^2$ | REML estimator of $\sigma_Z^2$ |
| $\theta = (\theta_1, ..., \theta_p)'$ | Unknown parameters of $R[h; \theta]$ determining how fast the correlation between parameters decreases |
| $\nu, l, r$ | Specification parameters for the curtosis of the two-sided desirability functions |

## Parameters in the case studies

| | |
|---|---|
| $smin$ | Wall thickness |
| $nt$ | Cup depth |
| $dmax$ | Radius of the cup |
| $k_{hin}, k_{ruck}$ | Curvatures of the passes of the roller tool in the spinning process |
| $rdw$ | Working radius of roller tool in the spinning process |
| $eaqu$ | End point of the last pass of the roller tool on the equidistant in the spinning process |
| $vrbk$ | Distribution of the passes on the formed rim in the spinning process |
| $f_s$ | Ratio of feed rate f and spindel speed s |
| $w_o$ | Outer width of the machined region in the necking-in process |
| $w_i$ | Inner width of the region with reduced diameter in the necking-in process |
| $\alpha_1, \alpha_2$ | Taper angles of the machined region in the necking-in process |
| $\beta$ | Necking-in ratio |
| $w_{iobs}$ | Observed inner width of the region with reduced diameter in the necking-in process |
| $\alpha_{1obs}, \alpha_{2obs}$ | Observed taper angles of the machined region in the necking-in process |
| $\beta_{obs}$ | Observed necking-in ratio |

# 1 Introduction

The optimization of a mechanical engineering process usually is a complex problem. On the one hand, large numbers of influencing parameters and several quality characteristics are to be optimized simultaneously. On the other hand, due to time and cost constraints there is a need to reduce the number of experimental runs. A well-known tool for such situations is sequential optimization.

Based on a small initial design a surrogate model is fitted. This model is refined by sequentially adding updating points until the global optimum is found. An example is the Efficient Global Optimization algorithm (EGO) by Jones et al. (1998) that uses the expected improvement criterion to generate updating points. EGO pioneered the idea of incorporating the model uncertainty to get a truly global search for an optimum. It was designed only for single-objective unconstrained problems. Various similar algorithms or direct enhancements of EGO for multivariate problems or constraint problems appeared. A common practice for multi-objective problems is to transform the multi-objective problem into a single-objective problem using scalaring weighting functions or the concept of desirabilities. These multivariate expected improvements are always determined numerically, assuming the distribution of the uncertainty of the surrogate model to be normal or at least to be known. Due to difficulties with the distribution of the uncertainty in a target-value problem all multivariate algorithms are restricted to minimization problems and maximization problems respectively. However, during the optimization of real-world processes a minimization or maximization may achieve over-optimized results.

## 1.1 Motivating example

We optimize a simple pot produced by a sheet metal spinning process. This is an incremental forming process that produces complex rotationally symmetric workpieces in low and medium volume of production. In Figure 1.1 the principle of the process is

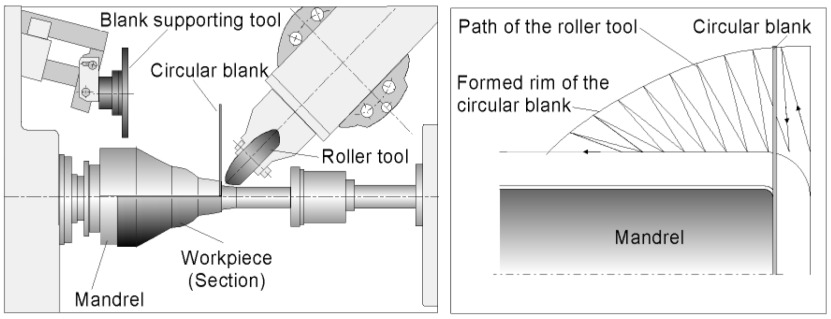


Figure 1.1: Producing a pot with the sheet metal spinning process

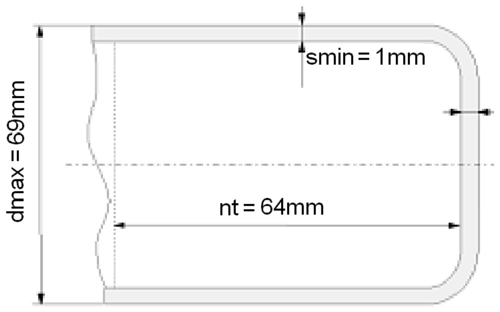


Figure 1.2: The optimization targets

illustrated. The pot is formed incrementally from a blank to the die. The forming tool (a roller tool) follows a path like it is shown on the right hand side of the figure.

The ideal pot is illustrated in Figure 1.2. The optimum shape is reached if the blank encloses the die as close as possible after the forming process. According to the theory of the sheet metal spinning process, the blank is only bent to the die and not buckled. The thickness of the blank, hence, stays constant. Hence, the die gives the targets of the optimization process. The diameter of the pot is given by the diameter of the die plus two times the thickness of the blank. The pot depth results from the radius of the blank minus the radius of the die. However, due to physical effects such as rubbing the sheet usually thins and thickening is assumed not to happen. In order to minimize thinning, we therefore aim to maximize the resulting sheet thickness. If the sheet thins out the pot depth also enlarges undesirably, i.e. the depth is to be minimized. The diameter of the pot is optimal if the die and sheet conform, but often not reached due to springback. Hence, the diameter is optimal if it is minimum. We use a sequential multivariate optimization technique to simultaneously minimize diameter and depth and maximize thickness. We find the 'optimum' pot that has a sheet thickness of

1.02mm and a depth of 59mm. It is too thick and too small, and hence a rather bad result instead of being the optimum. Against the assumption that the sheet continually thins, it thickened unexpectedly.

Obviously, for this particular application an optimization towards defined target values is needed. The simplest way would be to transform the quality characteristics to characteristics that contain the absolute deviations from the target values and minimize those objectives. However, it would be more preferable to consider that it is worse if the thickness is too thick than being too thin. Such an asymmetric penalization of the direction of deviation from the target can be reached using two-sided desirability functions.

## 1.2 Thesis contributions

This thesis introduces a sequential multivariate target-value optimization, called mtEGO, that uses two-sided desirabilities. The main problem that has to be solved is related to the distribution of the uncertainty of the surrogates after the transformation with two-sided desirabilities. This distribution is not known in a closed form and a full Monte Carlo simulation of this distribution is not feasible. It would lead to too many simulations and would be too time consuming. We therefore propose to get only a rough impression of the uncertainty distribution by determining a small number of virtual observations at each design point. The virtual observations are chosen using the predictions and prediction errors of the surrogate model. The virtual observations are transformed to two-sided desirability functions and aggregated to desirability indices. Based on the constructed desirability indices a rough impression of the improvements under the given uncertainty of the surrogate model is determined. We then use the idea from the expected improvement criterion that the parameter setting that maximizes the improvement is a candidate for the global optimum. Instead of determining the expectation of the improvement using the very roughly simulated uncertainty distribution, we generate several updating points simultaneously like other hybrid optimization algorithms do. Usually, many of the suggested updating points are similar, we therefore use cluster analysis to finally choose the essential updating points that find the global optimum efficiently.

The properties of mtEGO are evaluated with the help of simulation studies on various known test problems. As a result of the simulation study a guideline for the appropriate parameterization of the new approach is developed. Furthermore, mtEGO is compared to two brute force methods.

The mtEGO approach is characterized by its flexibility to handle various complex optimization targets. Minimization, maximization and target value problems can be handled as well as mixtures of these problem types, where the possibility to handle asymmetric two-sided target specifications is the most important property. mtEGO also enables to weigh between several objectives. It is not restricted to any dimensionality of the parameter space or number of objectives beside time constraints given by the user.

We further develop an enhancement of mtEGO that accelerates the algorithm for high-dimensional problems. The thesis also deals with the problem of unknown constraints. The handling of missing values during an optimization is discussed and results in another enhancement of mtEGO. Finally, the usability of mtEGO is demonstrated with the help of a target-value optimization of the sheet metal spinning process and the optimization of a necking-in process.

## 1.3 Thesis structure

The remainder of the thesis is organized as follows. Chapter 2 introduces the statistical background of sequential optimization required for the new approach. The chapter comprises sequential optimization in general, space-filling designs, Kriging models, the sequential design optimization algorithm by Cox and John, the popular EGO algorithm, the concept of desirability, a multivariate enhancement of EGO, and cluster analysis. In Chapter 3, the new approach mtEGO is presented. First, we demonstrate the idea of the virtual observations and how they work. Then we give a detailed step-by-step instruction of mtEGO, followed by the introduction of a criterion for the automatic choice of the appropriate number of clusters and the stopping criterion. Further, a step-by-step application example is presented. In Chapter 4 a simulation study evaluating the properties of mtEGO can be found. The algorithm is tested with several test problems for different parameterizations of mtEGO for standard and advanced optimization problems. The results are used to give recommendations for the parameterization of mtEGO and to compare mtEGO with two brute force methods. Limitations of the approach are discussed as well. Chapter 5 suggests two extensions of mtEGO: a more efficient extension for high-dimensional problems and an extension to handle unknown constraints and resulting missing values. Chapter 6, presents the application of mtEGO to the sheet metal spinning process from the motivating example in Section 1.1 and an optimization of a necking-in process. The last chapter contains a discussion on the remaining problems of mtEGO and possible future work.

# 2 Sequential optimization

In this chapter we give the statistical background for the methods presented in the following chapters. First of all, sequential optimization is introduced in general (Section 2.1). It is followed by brief introductions to space-filling designs (Section 2.2) and Kriging models (Section 2.3) which are essential for sequential optimization. We then present an algorithm by Cox and John and the EGO algorithm in Section 2.4 and 2.5. Details on the concept of desirability are given in Section 2.6. It is followed by the presentation of a multivariate extension of the EGO algorithm that uses desirabilities (Section 2.7). Finally, clustering methods are briefly introduced (Section 2.8).

## 2.1 Different approaches to sequential optimization

Various sequential optimization strategies can be found in the literature. All of them implement the same schema:

1. Generate an initial design.
2. Fit a surrogate model.
3. Locate new design points based on a certain criterion.
4. Update the surrogate model with the new points.
5. Repeat steps 3 and 4 until a reasonable stopping criterion is fulfilled.

One main difference between the optimization algorithms is the way that they locate the updating points in Step 3.

One possible method to search for the optimum is to restrict the design space sequentially. Examples can be found in Wang (2003), Osio and Amon (1996) and Wang and Simpson (2004). In each step they restrict the space based on the surrogate model and spread new points inside the restricted space. A drawback is that they restrict the design irreversibly and do not account for model uncertainty, which entails that they may only find a local optimum if the fitted model is not appropriate. A theorem by Törn and Žilinskas (1987, Ch.1.2) states, that in order to guarantee convergence to the global optimum of a continuous function, the algorithm must be able to converge to every point in the parameter space. According to Jones (2001) this means, that any method that shall really converge globally must "have a feature that forces it to pay attention to parts of the space that have been relatively unexplored and, from time to time, go back and sample in these regions".

As a consequence, several optimization strategies locate new design points based on criteria that take the model uncertainty into account. The "Taxonomy of Global Optimization Methods" by Jones (2001) gives a good overview of such strategies. The list below gives four examples that compete with each other.

- The efficient global optimization algorithm (EGO) introduced by Jones et al. (1998) maximizes the expected improvement of the candidates in the design space.
- Kushner (1964) maximizes the probability of improvement.
- Gutmann (2001) maximizes the credibility of the hypothesis that a value $f^*$ is the global optimum.
- Cox and John (1997) minimizes a statistical lower bound of the surrogate model.

Due to its good properties regarding efficiency and global convergence, the EGO algorithm by Jones has become very popular. The other algorithms mentioned above have not been as much accepted as EGO nor as advanced. We will concentrate on the EGO algorithm, since it is widely accepted especially among engineers. Also several multivariate enhancements already exist. However, the new approach seizes also the ideas of Cox and John.

Another group of sequential optimization algorithms are the so-called evolutionary algorithms (EA's). Updating points are generated with a selection and mutation strategy. The best point already found is selected. A mutation of these points is taken as the new updating point. The mutation includes random components, that either shift the algorithm towards the target region or explore an unknown region. EA's are known as simple and efficient algorithms. The research on this area happens side by side with the research on the optimization strategies around EGO. None of the strategies can be generally preferred to the other. Even some combinations of both approaches exist (e.g. Jeong and Obayashi, 2005). As already said, in this thesis we concentrate on the ideas of EGO.

## 2.2 Space-filling designs

Space-filling designs, sometimes also called exploratory designs, originate from the field of computer experiments. As a consequence, they underly specific principles. Santner et al. (2003, Ch.5.1.2) formulate the following two:

1. In computer experiments repeated observations at the same parameter setting yield identical responses. Designs should not take more than one observation at any parameter setting.

2. The uncertainty about the exact functional form of the relationship between the parameters and the responses in computer experiments imply that the designs should provide information about the whole experimental region to allow us to fit a variety of models.

Primarily implied by the second principle, space-filling designs have the advantage that they do not assume any special underlying model. The points of the design are selected using optimality criteria, such that they are spread evenly over the whole region of parameter settings. The initial designs in a sequential optimization procedure are usually space-filling designs. In what follows some popular space-filling designs are introduced in more detail. A comprehensive review of space-filling designs can be found in Santner et al. (2003, Ch.5).

### 2.2.1 Latin-Hypercube design

The Latin-Hypercube design (LHDs) is generated by Latin-Hypercube sampling. A design with $n$ points for $p$ independently distributed parameters is obtained as follows. First, the $p$-dimensional parameter space is partitioned into $n^p$ so-called cells. This is achieved by dividing the domain of each design parameter into $n$ intervals. The Cartesian products of those intervals gives the cells. A subset of $n$ cells is selected such that the projections of the centers of the cells onto each axis are uniformly spread across the axis. Finally, one design point is chosen randomly from each selected cell.

The explained algorithm does not guarantee to get an LHD which is intuitively space-filling over the full experimental region. In the two-dimensional space the diagonal may for example be an LHD design. To guarantee the design to be space-filling additional optimality criteria have to be applied. Extensions of the LHD are e.g. introduced in Morris and Mitchell (1995), Handcock (1991) or Owen (1992).

### 2.2.2 Minimax and maximin design

Very popular space-filling designs are the Minimax and Maximin design. They are using the $p^{th}$ order distance (also named Minkowsky metric) to select the design points. The $p^{th}$ order distance is defined by

$$\rho_p(x_i, x_j) = \left[ \sum_{k=1}^{b} |x_{ik} - x_{jk}|^p \right]^{\frac{1}{p}} \tag{2.1}$$

for $p \geq 1$. It measures the distance between two points $x_i, x_j$ from a grid of points $\mathbf{X} \subset \mathbb{R}^b$ representing the parameter space $\chi$. For $p = 1$ the $p^{th}$ order distance becomes the rectangular distance, also called city block metric, absolute distance, Manhattan distance or Taxicab distance. For $p = 2$ it becomes the Euclidian distance.

If the $p^{th}$ order distance is minimized

$$\min_{x_1, x_2 \in A} \rho_p(x_1, x_2) \tag{2.2}$$

one obtains a measure of closeness of any pair of points $x_1, x_2$ in a set of design points $A \subset \mathbf{X}$.

A Maximin distance design ($A_{Mm}$) maximizes the measure of closeness:

$$\min_{x_1, x_2 \in A_{Mm}} \rho_p(x_1, x_2) = \max_{A \subset \mathbf{X}} \min_{x_1, x_2 \in A} \rho_p(x_1, x_2). \tag{2.3}$$

By maximizing the minimum distance the Maximin design guarantees that no two points in the design are too close.

In contrast, a Minimax design ($A_{mM}$) aims to select the design points in such a way that every point in $\mathbf{X}$ is close to some point in $A$. It is formally defined as

$$\min_{A \subset \mathbf{X}} \max_{x \in \mathbf{X}} \rho_p(x, A) = \max_{x \in \mathbf{X}} \rho_p(x, A_{mM}). \tag{2.4}$$

The $p^{th}$ order distance $\rho_p(x, A)$ between a point $x$ and a set of points $A$ is thereby defined as

$$\rho_p(x, A) = \min_{x_i \in A} \rho_p(x, x_i). \tag{2.5}$$

Usually, the Euclidian distance ($p = 2$) is used for the Minimax or Maximin design. In the literature also designs using other distance measures like the average distance criterion function or the average projection design criterion can be found (cf. Santner et al., 2003, Ch.5.3).

### 2.2.3 Uniform coverage design

The uniform coverage design, also called U-optimal design, minimizes the deviation of a design from the uniform distribution. It is assumed that the parameter space $\chi$ is a $p$-dimensional hyper rectangular $\times_{i=1}^{p}[a_i, b_i]$ and $x \in \chi$ is a random variable with uniform distribution function

$$F(x) = \prod_{i=1}^{p} \left( \frac{x_i - a_i}{b_i - a_i} \right). \tag{2.6}$$

Let $A = \{x_1, ..., x_n\}$ be any design. The empirical distribution for the points in $A$ and one fixed point $x_0 = (x_{01}, ..., x_{0p})$ from the parameter space is

$$F_n(x_0) = \frac{1}{n} \sum_{i=1}^{n} I\{x_{i1} \leq x_{01}, ..., x_{id} \leq x_{0p}\}, \tag{2.7}$$

where $I\{.\}$ is the characteristic function. Hence, the distance $|F_n(x) - F(x)|$ is the discrepancy between one design point and the uniform distribution. A design $A_{unif}$ is a uniform coverage design if it minimizes the discrepancy between any design $A$ and the uniform distribution, i.e.

$$A_{unif} = \min_{A \subset \chi} \sup_{x \in \chi} |F_n(x) - F(x)|. \tag{2.8}$$

Since the uniform coverage criterion needs extensive computation time, the parameter space is usually discretized. Fang et al. (2000) describe two algorithms to generate the design for a discretized parameter space. The resulting design then is nearly uniform coverage and can be determined within short computation time. However, (nearly) uniform coverage designs need not be intuitively space-filling like LHDs.

### 2.2.4 Coffee-house design

The coffee-house design by Müller (2001, Ch.21) is a space-filling design with a very simple construction rule. Müller's idea is to generate the design subsequently just like people select their table in a Viennese coffee-house, which gives the name of the design. According to a popular Austrian novel upon arrival a new customer chooses that table in the coffee-house that is the most remote from all other customers already present in the coffee-house. In accordance to this principle the coffee-house design is generated as follows. Let $\chi$ be the parameter space. First the two points $x_i$ and $x_{i'}$ that have the maximum distance among all pairs in $\chi$ are chosen as initial two point design

$$A_2 = \{x_1, x_2\} = \arg \max_{x_i, x_{i'} \in \chi} \|x_i - x_{i'}\|.$$

Then additional points $x_i$ are added to the design $A_2$ subsequently, generating $A_{i+1} = A_i \cup x_{i+1}$, where

$$x_{i+1} = \arg \max_{x_0 \in \chi} \min_{x_i \in A_i} \|x_i - x_0\|,$$

which is the point that maximizes the minimum distance to all points in $A_i$. The construction of a coffee-house design is simple and quick. It asymptotically shares desirable properties of maximin distance designs.

### 2.2.5 Discussion of the different space-filling designs

Of course, many other space-filling designs than presented exist. Because of their complexity they are not considered here. Due to their simple construction Maximin and Latin-Hypercube designs are the most popular designs. However, LHDs become much more difficult when multiple constraints restrict the design space and may intuitively not be space-filling. Maximin designs avoid small distances between the design points which leads to the drawback that for a small number of design points they tend to locate the selected points near the boundary of the parameter region. Hence, Maximin and LHD designs often are not appropriate for practical applications. Minimax designs do not have this drawback, but are much harder to compute than Maximin designs. The computation time needed to generate them is so extensive that they are rarely used. Nearly uniform coverage and coffee-house designs have similar properties to the Minimax designs, but do not need extensive computation time. The combination of different criteria is a way to find a design that is simple to construct but still intuitively space-filling. One example is the Maximin LHD by Morris and Mitchell (1995). In a first step, the set of all LHDs with a design point in the middle of the space with the required size $n$ is generated. In a second step, that LHD is chosen that maximizes the minimum distance between two design points.

## 2.3 Kriging models

The initial design in sequential optimization is usually generated to be space-fillingly to assure that the whole design space is explored uniformly and no functional form of the relationship is assumed a priori. On the basis of the initial design a surrogate model is fitted that is updated sequentially during the optimization process. The surrogate model should now be likewise flexible as the design is. Furthermore, it should be able to model complex response surfaces data-faithfully and give information on the uncertainty of the model.

The Kriging model, also known as stochastic process model or spatial regression, provides such a model (Santner et al. (2003,Ch.2), Cressie (1993, Ch.3), Sacks et al. (1989)). Originally, it was introduced in the 1950s by Krige and forms the basis of geostatistics. At the end of the 1980s, Kriging techniques were applied to deterministic computer experiments, which became known under the name DACE (Design and Analysis of Computer Experiments). Nowadays, DACE or Kriging in general is widely used as surrogate model in statistical optimization. The great advantage of Kriging is that it permits realistic predictions of the response surface in the entire design space based on a relatively small number of design points by incorporating information about their spatial dependencies. The model differentiates between large-scale dependency modeled through the deterministic mean structure and small-scale dependency modeled through the correlation structure. Formally the model is defined as

$$y(x_i) = \mu(x_i) + Z(x_i), \text{ for } \; i = 1, ..., n \; , \tag{2.9}$$

where

- $x_i$ denotes the $i$-th sampled design point from the $p$-dimensional parameter space $\chi \in \mathbb{R}^p$.
- $\mu(x_i)$ defines a deterministic trend component with $E[y(x_i)] = \mu(x_i) = \sum\limits_{j=1}^{p} f_j(x_i)\beta_j$, where $f_1(.), ..., f_p(.)$ are known regression functions and $\beta = (\beta_1, ..., \beta_p)$ are unknown regression coefficients.
- $Z(x_i)$ is a realization of a second-order stationary stochastic process $Z = \{Z(x_i) : x_i \in \chi\}$ with $E[Z(x_i)] = 0$ and $Cov[Z(x_i), Z(x_j)] = \sigma_Z^2 R[x_i - x_j]$ for $i, j = 1, .., n$, where $R[x_i - x_j] = Corr[Z(x_i), Z(x_j)]$ and $\sigma_Z^2 = Var(Z(x_i))$.

Usually, further structure is imposed on $R$ to make the correlation matrix $R$ estimable and hence permit statistical inferences. The correlation function of the second-order stationary process $Z$ is assumed to depend only on the distance $h = x_i - x_j$ and a fixed number of unknown parameters $\theta = (\theta_1, ..., \theta_p)'$, i.e.
$Corr[Z(x_i), Z(x_j)] = R[h; \theta]$. Often a model from the class of nonlinear isotropic correlation functions is used to model the correlation structures of the process $Z$. The property isotropy implies that the correlation of two design points does not depend on the different directions of $h$, but only on the distance. The exponential model, the gaussian model, and the spherical model belong to this class.

A commonly used spatial correlation function in Kriging is the power exponential correlation function:

$$R[h; \theta] = exp\left\{ - \left( \frac{||h||}{\theta} \right)^s \right\}, \tag{2.10}$$

where $\theta > 0$ and $0 \leq s \leq 2$. The parameter $\theta$ determines how fast the correlation between the factors decreases. The parameter $s$ affects the smoothness of the function. If the assumption of isotropy is too restrictive, geometrically anisotropic correlation functions may be appropriate.

The anisotropic power exponential correlation function formally is:

$$R[h; \theta] = exp\left\{ - \sum_{l=1}^{p} \left( \frac{|h_l|}{\theta_l} \right)^{s_l} \right\}, \tag{2.11}$$

where $h_l = x_{il} - x_{jl}$ and $\theta_l > 0$ and $0 \leq s_l \leq 2$ for $l = 1, ..., p$. The parameters named $\theta_l$ and $s_l$ basically do have the same affect as in the isotropic case, but differentiate between the $l$ coordinate directions.

An anisotropic correlation function implies that many parameters are unknown. Sometimes the smoothness parameters $s_l$ are set to the fixed value 1 or 2. Still $\theta_l$, the unknown parameters of the deterministic trend component and $\sigma_Z^2$ are unknown. One should be aware of overfitting when fitting a Kriging model with a complex deterministic trend component and an anisotropic correlation function to a small design. Often only the intercept is chosen as the deterministic trend component then. This is known as ordinary Kriging.

### 2.3.1 Estimation of unknown parameters

The unknown parameters in the Kriging model can be estimated by the maximum likelihood (ML) method among other options. In the Kriging model the distribution of the random vector $Y^{(n)} = (Y(x_1), ..., Y(x_n))'$ is assumed to be n-dimensional multivariate normal

$$Y^{(n)} \sim N_n(\mu = F\beta, \sigma_Z^2 R), \tag{2.12}$$

where $F = (f(x_1), ..., f(x_n))'$ and $R$ is the correlation matrix of the random vector $Y^{(n)}$. Based on this distribution the log likelihood function for the vector of observations $y^{(n)}$ is

$$\begin{aligned} l(\beta, \sigma_Z) = -\frac{1}{2}\Bigg[ & n\log(2\pi) + n\log(\sigma_Z^2) + \log(\det(R)) \\ & + \frac{(y^{(n)} - F\beta)' R^{-1} (y^{(n)} - F\beta)}{\sigma_Z^2} \Bigg]. \end{aligned} \tag{2.13}$$

Differentiating the log likelihood with respect to $\beta$ and $\sigma_Z^2$, equating it to zero and using some algebra, the following ML estimators can be obtained

$$\hat{\beta} = (F'R^{-1}F)^{-1}F'R^{-1}y^{(n)} \tag{2.14}$$

$$\hat{\sigma}_Z^2 = \frac{1}{n}(y^{(n)} - F\hat{\beta})'R^{-1}(y^{(n)} - F\hat{\beta}). \tag{2.15}$$

The estimators $\hat{\beta}$ and $\hat{\sigma}_Z^2$ do assume a known correlation matrix R, which is usually not true. Let us assume the underlying structure for the correlation matrix R is unknown. The ML estimators of $\beta$ and $\sigma_Z^2$ in Equation 2.14 and 2.15 depend only on the correlation parameters $\theta$. When the ML estimators $\hat{\beta}$ and $\hat{\sigma}_Z^2$ are inserted in the log likelihood function of the Kriging model (Equation 2.13) it can be simplified to the log likelihood function

$$l(\hat{\beta}, \hat{\sigma_Z}, \theta) = -\frac{1}{2}[n\log(2\pi) + n\log(\hat{\sigma_Z}^2(\theta)) + \log(\det(R(\theta))) + n]. \tag{2.16}$$

The ML estimator of $\theta$ can then be obtained by numerically maximizing this log likelihood using e.g. the ridge-stabilized Newton-Raphson algorithm (Wolfinger et al., 1994). To estimate the parameters $\beta$ and $\sigma_Z^2$ the correlation matrix $R$ is substituted by its estimate $\hat{R} = R(\hat{\theta})$ in Equation 2.14 to 2.16.

For small samples the ML estimators of $\sigma_z^2$ and $\theta$ are known to be biased. The higher the dimension of $\beta$ the more biased are the estimators. In such a case, it is recommended to determine the restricted maximum likelihood (REML) estimates (cf. Patterson and Thompson, 1974). Without loss of generality let $rank(F)$ be the (row) rank of $F$ and $rank(F) = p$. The idea of REML is to determine a matrix $C$ with full (row) rank $n-p$ for which $CF = 0$. A transformation of the data with $C$ then results in

$CY^{(n)} \sim N(CF\beta = 0, \sigma_z^2 CR(\theta)C')$, which means the likelihood does not depend on $\beta$ anymore. The REML estimate of $\theta$ maximizes this likelihood of the transformed data $CY^{(n)}$ and can again be determined numerically. The REML estimate of $\sigma_Z^2$ becomes

$$\tilde{\sigma}_Z^2 = \frac{1}{n-p}(y^{(n)} - F\hat{\beta})'R^{-1}(y^{(n)} - F\hat{\beta}). \tag{2.17}$$

## 2.3.2 Prediction

With the help of the estimated parameters an (empirically) best linear unbiased predictor of a new point $y(x_0)$ and an (empirically) estimated prediction error can be obtained. In the case of a known correlation function, Sacks et al. (1989) prove that the best linear unbiased predictor (BLUP) for an unknown point $x_0$ given the observation vector $y^{(n)}$ is

$$\hat{y}(x_0) = f(x_0)'\hat{\beta} + r(x_0)'R^{-1}(y^{(n)} - F\hat{\beta}), \tag{2.18}$$

where $r(x_0)' = (R(x_0, x_1), ..., R(x_0, x_n))$ is the vector of correlations between the new point $x_0$ and the observations $y^{(n)}$. The prediction error is estimated by the mean squared prediction error (MSPE)

$$\begin{aligned} MSPE(\hat{y}(x_0)) &= E\{(\hat{y}(x_0) - y(x_0))|y^{(n)}\} \\ &= \sigma_Z^2(1 - r(x_0)'R^{-1}r(x_0 + a'(F'R^{-1}F)^{-1}a) \end{aligned} \tag{2.19}$$

with $a = f(x_0) - F'R^{-1}r(x_0)$ (cf. Santner et al., 2003, Ch.3). The MSPE is defined to be zero for observed points, which is intuitive for deterministic experiments.

In fact the correlation function needed in Equation 2.18 and 2.19 is usually not known and $R$ and $r(x_0)$ have to be replaced by the estimates $\hat{R}$ and $\hat{r}(x_0)$. The estimator $\hat{y}(x_0)$ is then called the empirically best linear unbiased predictor (EBLUP), although it is usually not linear and unbiased anymore. For the MSPE $\sigma_Z^2, R$ and $r(x_0)$ are replaced by $\hat{\sigma}_Z^2, \hat{R}$ and $\hat{r}(x_0)$, changing it to an empirical estimator.

For simplicity, the prediction error $MSPE(\hat{y}(x_0))$ is referred to as $s_{\hat{y}}^2(x_0))$ in the following sections.

## 2.3.3 Kriging for nondeterministic experiments

In the previous subsections the Kriging model for deterministic experiments was introduced. The model is appropriate if the design is conducted as a computer experiment, e.g. with FEM-Simulation techniques or if the experiments can be assumed as exactly repeatable. When physical experiments are conducted an error term is added to the model to consider the process variance. The model then has the form

$$y(x_i) = \mu(x_i) + Z(x_i) + \epsilon(x_i), \text{ for } \; i = 1, ..., n \; , \tag{2.20}$$

where $\mu(x_i)$ and $Z(x_i)$ are as before, and $\epsilon(x_i)$ is the additional error term. The error term is considered as a realization of a white-noise process $\mathrm{E} = \{\epsilon(x_i) : x_i \in S\}$ with

$$\begin{aligned} &E[\epsilon(x_i)] = 0 \\ &\text{and } Cov[\epsilon(x_i), \epsilon(x_j)] = \begin{cases} \sigma_\epsilon^2 & ,\text{if } x_i = x_j \\ 0 & ,\text{if } x_i \neq x_j \end{cases} \end{aligned} \tag{2.21}$$

The covariance matrix of $y(x)$ now is

$$V = \sigma_Z^2 R + \sigma_\epsilon^2 I = \sigma_\epsilon^2 \left( \frac{\sigma_Z^2}{\sigma_\epsilon^2} R + I \right). \tag{2.22}$$

Analogously to the deterministic case, estimators for $\hat{\beta}$ and $\sigma_\epsilon^2$ can be obtained by maximizing the likelihood function. The log likelihood has the same form as in the deterministic case, only $R$ has to be substituted with $W = \frac{\sigma_Z^2}{\sigma_\epsilon^2} R + I$. Likewise, the BLUP can still be obtained with Eq. 2.18 when $R$ is substituted with $W$. The MSPE for the nondeterministic case has the form

$$MSPE(\hat{y}(x_0)) = \sigma_Z^2 (1 + \frac{\sigma_\epsilon^2}{\sigma_Z^2} - r(x_0)' W^{-1} r(x_0 + a'(F'W^{-1}F)^{-1} a), \tag{2.23}$$

with $a = f(x_0) - F'W^{-1}r(x_0)$ (cf. O'Connell and Wolfinger, 1997). Note, that this MSPE is not zero for observed points.

## 2.3.4 Model validation techniques

The Kriging model is a very flexible approach, that permits various different model specifications, e.g. different trend components, different spatial correlation structures, isotropy or anisotropy and so on. Some goodness of fit measures may help to choose the best model specification. The Akaike criterion (AIC) and the Schwarz criterion (BIC) are well-known goodness of fit measures. They are based on the deviance

$$L = -2log(l*),$$

where $l*$ is the maximized likelihood value of the model. The AIC criterion as defined by Akaike (1974) is

$$AIC = L + 2u, \tag{2.24}$$

where $u$ is the number of estimated parameters. The smaller the AIC is, the better is the model. The summation of $2u$ to the deviance $L$ penalizes a large number of parameters and aims to prevent overfitting. The BIC criterion was defined by Schwarz (1975). It penalizes large number of parameters depending on the number of design points $n$ that are used to fit the model. Formally it is

$$BIC = L + log(n)u. \tag{2.25}$$

A comparison of the two criteria shows that for $n > 7$ BIC harder penalizes a large number of parameters than AIC.

Jones et al. (1998) suggest to use several diagnostic plots to validate the fitted model based on the leave-one-out cross validation method. Suppose the model was fitted with the help of observations $y^{(n)} = [y(x_1), ..., y(x_n)]$. Each of the $y(x_i)$ is eliminated one at a time and the model is fitted to the remaining observations. Then the deleted observation $y(x_i)$ is predicted from the model fitted with the remaining data. These predictions are plotted against their respective observations. For a good model, the points in the plot are close to the 45° line. If the model is valid, it also holds approximately that

$$\frac{y(x_i) - \hat{y}_{(-i)}(x_i)}{\sqrt{s^2_{\hat{y}_{(-i)}}(x_i)}} \sim N(0,1) \text{ for } \; 1 \leq i \leq n, \tag{2.26}$$

where $\hat{y}_{(-i)}(x_i)$ is the prediction of point $x_i$ from the model fitted without $y(x_i)$ and $s^2_{\hat{y}_{(-i)}}(x_i)$ the corresponding prediction error. Jones calls this quantity the standardized cross-validated residual (SCVR) and states that the SCVR values should roughly be in the interval [-3,3] for a valid model due to their normal distribution. Additionally, the SCVR's can be visualized in a Q-Q plot (SCVR's against standard normal quantiles) and should be close to the 45° line there, too.

The cross-validation technique can also be used to compare the prediction abilities of different models. The smaller the mean squared cross-validation error (MSCVE)

$$MSCVE = \frac{1}{n}\sum_{i=1}^{n}(y(x_i) - \hat{y}_{(-i)}(x_i))^2 \tag{2.27}$$

is, the better are the predictions of the model.

## 2.4 Sequential design optimization (SDO) by Cox and John

Cox and John (1997) introduced a sequential sampling method to find a global minimum. Starting with a small space-filling design they stepwise fit a Kriging model (cf. Section 2.3). New design points are generated based on the standard prediction error provided by the Kriging model. Let $\mathbf{X} = (x_i)_{i=1,\dots,l}$ be a grid of candidate points representing the parameter space $\chi$. Let further $\hat{y}(x_i)$ be the predictions and $s^2(x_i)$ the mean squared prediction errors provided by the Kriging model and $\hat{\sigma}_Z^2$ the estimation of the variance of the stochastic component $Z(x)$ of the Kriging model as defined in Section 2.3. Cox and John determine a lower confidence bound of the predictions

$$lcb(x_i) = \hat{y}(x_i) - \kappa\hat{\sigma_Z}^2\sqrt{s^2(x_i)} \tag{2.28}$$

for some specified $\kappa$. The candidate $x_i$ that minimizes $lcb(x_i)$ is chosen as the next updating point. The algorithm stops when a user-defined maximum number of experiments is reached or when the current observed minimum $y_{min}$ is smaller than the minimum of the lower confidence bounds, i.e.

$$y_{min} < \min_{i=1,..,l}(lcb(x_i)). \tag{2.29}$$

Cox and John suggest setting $\kappa = 2$ or $\kappa = 2.5$. This method has been reviewed by Jones (2001) using $\kappa = 5$ as a conservative choice for $\kappa$. He shows that the method fails to find the global optimum. He states that the success of the method strongly depends on the choice of $\kappa$. Jones suggests enhancing the method by using several confidence boundaries with different $\kappa$'s simultaneously. Each confidence boundary with a different $\kappa$ is minimized by another point, which results in different possible optima. All points resulting from minimizing the confidence boundaries are then clustered to form a small number of updating points which are used to refine the Kriging model. He writes that this "appears to be a highly promising approach". Cox and John's algorithm has not been used or advanced, probably due to the popularity of the EGO algorithm. However, the virtual observations used in the new approach (cf. Section 3.1) are very similar to the confidence boundaries from Cox and John's SDO algorithm. The mtEGO algorithm also uses the idea of Jones to consider several confidence boundaries simultaneously.

## 2.5 The EGO-algorithm

The efficient global optimization algorithm (EGO) was published by Jones et al. (1998). First, an initial space-filling design is generated. The number of design points should be as small as possible, but should enable to fit a surrogate that reflects the true relationship at least roughly. Jones et al. propose to use 10 points per influencing parameter. Then a Kriging model is fitted. A so-called expected improvement criterion is evaluated for each candidate in the design space. The point that maximizes the criterion is the new updating point. The model and the expected improvement are refined then with the new points. This refinement is repeated until some lower boundary for the expected improvement value is reached. The following subsection gives details on the expected improvement criterion.

### 2.5.1 The expected improvement criterion

The expected improvement criterion was introduced by Schonlau (1997). It balances exploitation from the meta-model where the prediction is optimal with the need for exploration where the uncertainty of the metamodel is high. Without loss of generality, let us assume our problem is to find the minimum of the objective $Y(x)$ given an $n$ point initial design with responses $y^{(n)} = (y(x_1), .., y(x_n))'$. Let $\hat{y}(x)$ be the prediction of the surrogate model fitted to the design for each possible point $x$ in the design space and let $s^2(x)$ be the corresponding mean squared prediction prediction errors. Further, it is assumed that for a new point $x$ the response $Y(x)$ given $y^{(n)}$ is normally distributed with mean equal to the prediction and variance equal to the prediction error, hence $Y(x)|y^{(n)} \sim N(\hat{y}(x), s^2_{\hat{y}(x)})$. We neglect the fact that the parameters $\hat{y}(x)$ and $s^2_{\hat{y}(x)}$ are estimated and that a t-distribution would be more appropriate here. To determine the capability of $x$ to be the minimum regarding the currently observed local minimum $y_{min}$ a probabilistically-based improvement function is defined by

$$I(x) = \begin{cases} y_{min} - Y(x) & , \text{ if } Y(x) < y_{min} \\ 0 & , \text{ if } Y(x) \geq y_{min} \end{cases} \tag{2.30}$$

In an intuitively appealing way, the improvement function increases the smaller $Y(x)$ is than the current minimum $y_{min}$, and is 0 for values $Y(x)$ greater or equal than the current minimum. Hence, the improvement function $I(x)$ only measures how close $Y(x)$ is to the target $y_{min}$, the model uncertainty is not taken into account so far. The conditional density of the predictions $f_{Y(x)|y^{(n)}}$ can be seen as a measure of uncertainty of the fitted model.

Therefore, the conditional expectation of $I(x)$, given the observations $y^{(n)}$, is determined as

$$E[I(x)|y^{(n)}] = \int_{-\infty}^{y_{min}} (y_{min} - z) f_{Y(x)|y^{(n)}}(z) dz. \tag{2.31}$$

The conditional expectation $E[I(x)|y^{(n)}]$ is called the expected improvement. Due to the assumption that $Y(x)|y^{(n)} \sim N(\hat{y}(x), s^2_{\hat{y}(x)})$, the expected improvement can be transformed to the closed form

$$E[I(x)|y^{(n)}] = (y_{min} - \hat{y}(x))\Phi\left(\frac{y_{min} - \hat{y}(x)}{s_{\hat{y}(x)}}\right) + s_{\hat{y}(x)}\phi\left(\frac{y_{min} - \hat{y}(x)}{s_{\hat{y}(x)}}\right), \tag{2.32}$$

where $\phi()$ and $\Phi()$ are the density and distribution function of the standard normal distribution (a proof is given in Henkenjohann, 2006). The first term is the improvement of $y_{min}$ multiplied by the probability that this improvement is actually achieved. The second term is the prediction error multiplied by the probability that $y$ is smaller than $y_{min}$. Hence, the expected improvement is large in areas where the improvement $(y_{min} - z)$ is likely to be large, i.e. the prediction is close to the optimum, or where the model uncertainty is high. Particularly, it gets large if the prediction is close to the optimum and uncertain. The expected improvement gets 0 for points that are observed, if the prediction error is 0 in those points, which is true for computer-experiments.

The calculation of the expected improvement and the updating of the model is repeated until a stopping criteria is reached. Originally, Schonlau (1997) suggests stopping if the expected improvement becomes smaller than a fixed threshold value, e.g. 0.001. Since the expected improvement is dependent of the size of the observed quality characteristics, Jones et al. (1998) suggest stopping if the expected improvement becomes smaller than 1% of the currently observed optimum. Henkenjohann et al. (2005) propose that the threshold value should be 1% of the range of the observations $|y_{min} - y_{max}|$. They argue that if a quality characteristic is enlarged with an additive constant $c = 100$, the intercept of the surrogate model changes. The expected improvement does not change then, but a threshold value that depends on the currently observed optimum does.

In Jones et al. (1998) a generalized version of the expected improvement is introduced as follows

$$E[I^g(x)|y^{(n)}] = \int_{-\infty}^{y_{min}} (y_{min} - z)^g f_{Y(x)|y^{(n)}}(z) dz. \tag{2.33}$$

The exponent $g$ enables the user to weigh how important areas of high uncertainty are and enables to control between a local or global search. The larger g is chosen the more global the search is. Different sizes of $g$ and also cyclic schemes for $g$ have been discussed in Sasena et al. (2000). For the generalized expected improvement the same stopping rules as for the original version can be used. They now refer to $(E[I^g(x)|y^{(n)}])^{1/g}$ instead of $E[I(x)|y^{(n)}]$.

### 2.5.2 Recent developments regarding EGO

The EGO-algorithm is designed for single-objective maximization (resp. minimization) problems without constraints and without noise. Many versions and extensions have been published recently.

Hawe and Sykulsky (2007) published a so-called hybrid-algorithm with different phases and a cooling scheme to gain more efficiency and accuracy compared with the original EGO for standard situations. Sóbester et al. (2005), Sasena (2002) and Forrester et al. (2006) extended the EGO algorithm to enable the optimization of constrained objectives. Williams et al. (2000), Queipo et al. (2002), Huang et al. (2006) or Villemonteix et al. (2009) give examples how to adapt or modify EGO to cope with noisy objectives for robust optimization. All approaches mentioned above are designed to optimize only one objective. However, for many fields like mechanical engineering it is substantial to optimize several objectives simultaneously. Multi-objective optimization leads to the concept of pareto-optimality introduced by Pareto (1896). A realization of $b$ objectives $\mathbf{Y}(x_0) = (y_1(x_0), ..., y_b(x_0))$ is said to be pareto-optimal if there is no other realization $\mathbf{Y}(x) = (y_1(x), ..., y_b(x)),\ x \in \chi$, that improves the quality of one objective without decreasing the quality of at least one other objective. That means the realization $\mathbf{Y}(x_0)$ dominates all other realizations $\mathbf{Y}(x)$ and is, hence, a non-dominated pareto solution. The set of all non-dominated pareto solutions is called the pareto front.

The extensions of the EGO algorithm to the multi-objective case split into two groups in the literature. Hawe and Sykulsky (2004) and Jeong and Obayashi (2005) optimize the objectives separately as long as possible to get to know the pareto front and then determine one pareto-optimum as a best compromise for the present application. In contrast, the algorithms ParEGO by Knowles (2005), HyPerModel by Turner et al. (2007) and ASOP by Henkenjohann et al. (2005) transform the multi-objective problem into a single-objective problem first and proceed with a single-objective sequential algorithm. They only search for one pareto-optimum, but not for the whole pareto front. The transformation is done differently. ParEGO uses scalarizing Tchebycheff-functions, HyPerModel uses a criterion that is a weighted sum of four different criteria on proximity, weight, slope magnitude and convergence metrics and ASOP uses the concept of desirabilities. All approaches to multivariate optimization are designed for minimization or maximization of the objectives. To the best of our knowledge, an algorithm that enables multivariate optimization towards a special target value is not available yet. Since the concept of desirabilities provides handsome features for the target-value optimization problem, our approach adopts and extends the ideas of ASOP by Henkenjohann et al. (2005).

Some of the previously mentioned references make use of hybrid, cooling or cyclic schemes to yield faster and more precise results. Sasena et al. (2000), Sóbester et al. (2005), Turner et al. (2007), Hawe and Sykulsky (2007) vary their criterion-parameters regarding a special cooling or cyclic scheme reminiscent of simulated annealing. They start with a global search that becomes more local with each step. Jones (2001) presents an enhancement for different sequential optimization strategies that suggests to work in a hybrid way. A set of different parameterizations of one infill sampling criterion is used simultaneously which results in a set of updating points. The obtained results are clustered to few essential updating points. In each single step maximum information is obtained. The concept of using different parameterizations simultaneously and cluster the obtained results has great importance for the new strategy mtEGO.

## 2.6 The concept of desirability

The concept of desirability was introduced by Harrington (1965) for multi-objective optimization in industrial quality management. Harrington's idea is to transform the $b$-dimensional multi-objective problem into a single-objective problem. First, each single objective $y$ is transformed to the scale-free interval $[0, 1]$ with a so-called desirability function $d(y)$. After the transformation of all objectives $\mathbf{Y} = (y_1, ..., y_m)$ with desirability functions to gain comparable scales and comparable targets, they are aggregated to a joint desirability index $DI(\mathbf{Y}) = DI(y_1, ...y_m)$. The joint optimum of the multi-objective problem can be found by optimizing the desirability index. The specification of a desirability function needs experts knowledge about the problem that is to be optimized. Particularly, information is needed about the target and how strong the non-optimality has to be penalized. In the following, different types of desirability functions and desirability indices with their corresponding features are described.

### 2.6.1 Types of desirability functions

Desirability functions can be grouped into one-sided and two-sided functions. One-sided functions map quality characteristics to $[0, 1]$ that are to be minimized or maximized. If the optimum of a quality characteristic is a certain target value, we need a two-sided desirability function. For both types, a desirability value of 0 corresponds to an unacceptable quality. The desirability values get larger, the better the quality is, slowly approaching 1. A desirability value of 1 indicates the perfect quality, the target is exactly met.

For the one-sided case Harrington defines the desirability function as

$$d(y) = \exp(-\exp(-(b_0 + b_1 y))), \text{ with } b_0, b_1 \in \mathbb{R}. \tag{2.34}$$

The parameters $b_0$ and $b_1$ explicitly define the kurtosis of the function. The expert should provide two points of $y$ and their associated values of $d(y)$ to determine $b_0$ and $b_1$ with a system of two equations.

To specify the two-sided desirability function of Harrington an upper and a lower specification limit ($USL$ and $LSL$) are required that lie symmetrically around the target value and which are associated with a desirability of $1/e$. The two-sided desirability function has the form

$$d(y) = \exp\left(-\left|\frac{y - \frac{USL+LSL}{2}}{\frac{USL-LSL}{2}}\right|^{\nu}\right) \text{ with } \nu > 0. \tag{2.35}$$

The parameter $\nu$ determines the kurtosis of the function and has to be chosen such that it meets the expert's preferences. The larger $\nu$ is, the less the function penalizes deviations from the target. Figure 2.1 shows some exemplary Harrington desirability functions.

Derringer and Suich (1980) introduce desirability functions that are defined piecewise and also enable asymmetric forms. The one-sided Derringer-Suich desirability function

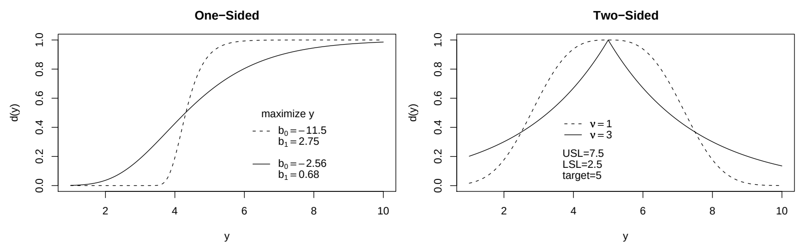


Figure 2.1: Examples of Harrington desirability functions: one-sided (left) and two-sided (right)

for a maximization problem has the form

$$d(y) = \begin{cases} 0 & \text{,if } y < LSL \\ \left(\frac{y-LSL}{T-LSL}\right)^l & \text{,if } LSL \leq y \leq T \\ 1 & \text{,if } y > T \end{cases}, \tag{2.36}$$

where $T$ is the target value. It is advisable to use this function, if a precise target $T$ and a certain lower specification limit $LSL$ are known. The $USL$ marks the specification limit where all values of $y$ greater than $T$ realize no additional benefit and values smaller than $LSL$ result in an unacceptable quality. The parameter $l$ again specifies the kurtosis of the function and hence influences how strong deviations from the target are penalized. For a minimization problem the desirability function is specified likewise.

Derringer and Suich also provide a two-sided desirability function for target-value problems in the following form

$$d(y) = \begin{cases} 0 & \text{,if } y < LSL \\ \left(\frac{y-LSL}{T-LSL}\right)^l & \text{,if } LSL \leq y \leq T \\ \left(\frac{y-USL}{T-USL}\right)^r & \text{,if } T < y \leq USL \\ 0 & \text{,if } y > USL \end{cases} \tag{2.37}$$

Here, the lower and upper specification limits $LSL$ and $USL$ represent those values of $y$, where the deviation from $T$ is that large that an unacceptable quality is reached. The desirability gets 0 outside these limits. With the help of the parameters $l$ and $r$ the shape of the function below and above the target value are determined. Specifying $l$ and $r$ differently, this desirability function can be asymmetric. This is very useful for situations where a deviation below the target is worse than exceeding the target or vice versa. For example the optimization problem from the motivating example in Section 1.1 can be transformed to two-sided asymmetric desirabilities to penalize sheet thicking adequately. Figure 2.2 shows examples of Derringer-Suich desirability functions.

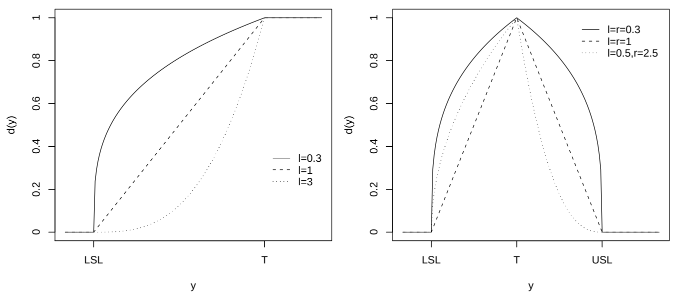


Figure 2.2: Examples of Derringer-Suich desirability functions: one-sided (left) and two-sided (right)

Further specifications of desirability functions can be found in Govaerts and Le Bailly de Tilleghem (2005). They develop functions based on the cumulative distribution function of the standard normal distribution to provide smoother and differentiable functions. Asymmetric specifications are however permitted.

## 2.6.2 Types of desirability indices

To evaluate the joint quality of $b$ objectives at the same time, Harrington (1965) suggests to aggregate the desirabilities to a desirability index. He defines the desirability index as the geometric mean of the single desirabilities $d_j(y_j),\ j = 1, ..., b$

$$DI(\mathbf{Y}) = \prod_{j=1}^{b} d_j(y_j)^{1/b}. \tag{2.38}$$

A desirability index $DI(Y)$ close to 1 indicates that the observation $\mathbf{Y} = (y_1, .., y_b)'$ has an overall good quality, whereas $DI(\mathbf{Y}) = 0$ indicates an unacceptable quality result. Notice, that if one of the single desirabilities is 0, i.e. one of the objectives has unacceptable quality, the overall quality is unacceptable, too. Figure 2.3 shows contours of two desirability functions and their joint desirability index. The joint optimum can easily be seen from the right contour plot.

In Trautmann and Weihs (2004) it was shown that a point $x_{opt}$ that is determined by optimizing the geometric mean of desirability functions is pareto-optimal. This makes the concept of desirability very appealing for multi-objective optimization.

In some cases, it may be appropriate to use the weighted geometric mean as the desirability index to allocate different importance to the quality characteristics. The

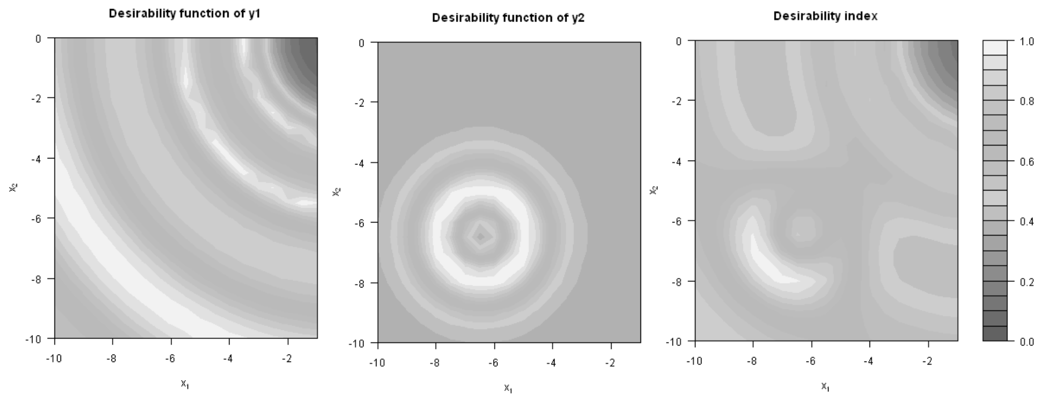


Figure 2.3: Two exemplary desirability functions and their joint desirability index

index then has the form

$$DI(\mathbf{Y}) = \prod_{j=1}^{b} d_j(y_j)^{g_j}, \text{ with } \sum_{j=1}^{b} g_j = 1. \tag{2.39}$$

There exist also other versions of desirability indices, e.g. $DI = \min_{j=1,\ldots,b} d_j$ by Kim and Lin (2000). These versions play a minor role for optimization, since they omit a lot of information and pareto-optimality is not given.

## 2.6.3 Distribution of desirability functions and indices

Optimization using the concept of desirability usually starts with a statistically planned design. A model is fitted, which is used to predict the desirability index for each point in the parameter space. Finally, the optimum is found by maximizing the predicted desirability index. The prediction error from the fitted model is ignored when optimizing the desirability index. A simulation study published by Steuer (2005) showed that the results may be largely wrong. The found optimum was often lying far away from the actual optimum. Sometimes the true desirability of the found optimum was even 0. Therefore, it may be wise to regard the predicted desirability index as a random variable and to optimize their estimated expectation instead.

Weber and Weihs (2003) derive distributions of different desirability functions. Assuming the quality characteristic to be normally distributed with expectation $\mu$ and variance $\sigma^2$, the distribution of the Harrington desirability functions $d$ can be derived applying the density transformation theorem multiple times.

The distribution of the one-sided Harrington desirability function is the double log-normal distribution with density

$$f(d) = \frac{-1}{\sqrt{2\pi}\tilde{\sigma}\log(d)d} \exp\left[-\frac{1}{2\tilde{\sigma}^2}(\log(-\log(d)) - \tilde{\mu})^2\right], \tag{2.40}$$

where $\tilde{\mu} = -(b_0 + b_1\mu)$ and $\tilde{\sigma}^2 = b_1^2\sigma^2$. The distribution of the two-sided specification is given by the density

$$f(d) = c\left[\exp\left(\frac{-((-\log(d))^{1/\nu} - \tilde{\mu})^2}{2\tilde{\sigma}^2}\right) + \exp\left(\frac{-((-\log(d))^{1/\nu} + \tilde{\mu})^2}{2\tilde{\sigma}^2}\right)\right] \tag{2.41}$$

where

- $c = \frac{(-\log(d))^{\frac{1}{\nu}-1}}{\sqrt{2\pi}\tilde{\sigma}d\nu}$
- $\tilde{\mu} = \frac{2}{USL-LSL}\mu + \frac{USL+LSL}{USL-LSL}$
- and $\tilde{\sigma}^2 = \left(\frac{2}{USL-LSL}\right)^2\sigma^2$.

The derivation of the distributions for Derringer-Suich desirability functions is much more difficult. Only for the special case with $r = l = 1$ a distribution can be determined in the same manner (cf. Steuer, 2005). In the one-sided case for a the-smaller-the-better quality characteristic the density has the form

$$f(d) = \begin{cases} \Phi\left(\frac{LSL-\mu}{\sigma}\right) & \text{, if } d = 0 \\ \frac{T-LSL}{\sigma}\phi\left(\frac{LSL+d(T-LSL)-\mu}{\sigma}\right) & \text{, if } d \in (0,1) \\ 1 - \Phi\left(\frac{T-\mu}{\sigma}\right) & \text{, if } d = 1 \end{cases} \tag{2.42}$$

where $\phi$ is the density and $\Phi$ the distribution function of the standard normal distribution. The distribution of the two-sided Derringer-Suich desirability with $r = l = 1$ has the density

$$f(d) = \begin{cases} \Phi\left(\frac{LSL-\mu}{\sigma}\right) + 1 - \Phi\left(\frac{USL-\mu}{\sigma}\right) & \text{, if } d = 0 \\ \frac{T-LSL}{\sigma}\phi(q_1) + \frac{USL-T}{\sigma}\phi(q_2) & \text{, if } d \in (0,1) \\ 0 & \text{, if } d = 1 \end{cases} \tag{2.43}$$

where $q_1 = \frac{LSL+d(T-LSL)-\mu}{\sigma}$ and $q_2 = \frac{USL-d(USL-T)-\mu}{\sigma}$.

The derivation of exact distribution or density functions for the geometric mean as desirability index is very complex and only possible for some special cases. Trautmann and Weihs (2006) derived the exact distribution for the bivariate case of two-sided

Harrington desirabilities with $\nu = 1$ and approximative distributions for one-sided desirabilities. Based on large simulation studies, Steuer (2005) yields information about some properties of the distribution of desirability indices with Derringer-Suich desirabilities, but also no closed forms are given. In Govaerts and Le Bailly de Tilleghem (2005) density functions for their own desirability functions are given for independent objectives.

## 2.7 Multivariate expected improvement using desirabilities

As already mentioned in Section 2.5.2 the expected improvement criterion has been extended for the multivariate case in different ways, where the variant of Henkenjohann et al. (2005) is the most appealing version for our problem. Henkenjohann et al. develop a multivariate expected improvement criterion based on the concept of desirabilities. The predictions of the fitted surrogate models are transformed to desirabilities and combined in one desirability index to proceed with the univariate expected improvement criterion. The calculation of the univariate expected improvement does not change much. In Equation 2.30 to 2.33 the univariate objective $Y(x)$ has to be replaced by the desirability index $DI(x)$ as follows. Assume we have a $b$-dimensional multivariate optimization problem. Let $DI$ be the desirability index of the $b$ desirability functions $d_j,\ j = 1, ..., b$, defined for the $b$ objectives. Let further $DI_{max}$ be the maximum desirability index within the current observations $y_1^{(n)}, ..., y_b^{(n)}$. The multivariate improvement function based on the desirability index then has the form

$$I_M^g(x) = \begin{cases} (DI(x) - DI_{max})^g & \text{, if } DI(x) > DI_{max} \\ 0 & \text{, if } DI(x) \leq DI_{max} \end{cases} \tag{2.44}$$

Taking the conditional expectation leads to the multivariate expected improvement

$$E[I_M^g(x)|y_1^{(n)}, ..., y_b^{(n)}] = \int\limits_{DI_{max}}^{1} (z - DI_{max})^g f_{DI(x)|y_1^{(n)},...,y_b^{(n)}}(z)dz, \tag{2.45}$$

where $f_{DI(x)|y_1^{(n)},...,y_b^{(n)}}$ is the conditional density of the desirability index $DI(x)$ given the observations $y_1^{(n)}, ..., y_b^{(n)}$.

The stopping criteria for this multivariate expected improvement depends on the currently observed maximum desirability index $DI_{max}$. The contour plots of the desir-

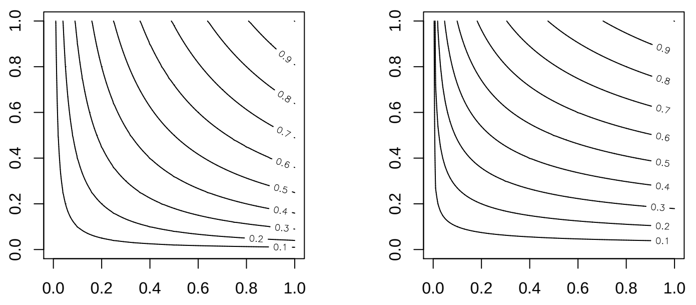


Figure 2.4: Contour plots of the desirability index with weights for the geometric mean (left: $g_1 = g_2 = 0.5$, right: $g_1 = 0.3, g_2 = 0.7$) (from Henkenjohann, 2006)

ability indices in Figure 2.4 show that the lines are roughly parallel only for $DI_{max} > 0.6$ and only then the relation of the value of the expected improvement is relatively constant to its actual improvement of the current optimum. Therefore, only if $DI_{max} > 0.6$ the process can be stopped if a certain multiple of the expected improvement is smaller than 1% of $DI_{max}$, otherwise the decision should depend on the location of the distribution of the desirability index.

In general, the strategy works with every kind of desirability function and desirability index, and can hence also be used with a target-value problem. However, it is difficult to obtain the conditional density of the desirability index $f_{DI(x)|y_1^{(n)},\ldots,y_b^{(n)}}$. This poses a great challenge, especially for two-sided desirability functions or Derringer-Suich desirability functions since they are only piecewise differentiable. Henkenjohann et al. (2005) could only give the conditional density for the special case of Harrington's one-sided desirabilities $d_j(x)$ and the geometric mean as index. Assuming $\mathbf{Y}(x) \sim N(\hat{y}(x), \hat{\Sigma}(x))$, the conditional density function of $DI(x) = d_j(x)$ has the form

$$f(DI(x)) = \frac{(-1)^b}{\sqrt{(2\pi)^b|\tilde{\Sigma}(x)| \prod_{j=1}^{I} log(d_j(x))d_j(x)}} e^{-\frac{1}{2}a'\tilde{\Sigma}(x)^- a} \tag{2.46}$$

where $a = (log(-log(DI(x))) - \tilde{\mu}(x))$ with $\tilde{\mu}(x) = b_0 + b_1\hat{y}(x)$ and $\tilde{\Sigma}(x) = b_1'\hat{\Sigma}(x)b_1$.

Henkenjohann (2006) also discuss the use of a Monte-Carlo simulation of the conditional density function and show that it is computationally not feasible to do so.

## 2.8 Clustering methods

Some multivariate optimization algorithms work in a hybrid way, several different infill sampling criteria or parameterizations of one criterion are evaluated simultaneously. All updating points that are developed simultaneously are reduced to few good updating points using simple methods from cluster analysis. Since the new approach also is a hybrid strategy, the idea of cluster analysis and some algorithms are introduced.

Cluster analysis is the generic name for a wide variety of techniques that are used to empirically group entities from an unclassified data set into homogeneous subgroups, so-called clusters. The points are grouped on the basis of their similarities, such that the distance between any two points in the cluster is less than the distance between any point in the cluster and any point not in it. This makes the clusters as homogeneous as possible.

### 2.8.1 Distance measures

The recognition of entities as similar or dissimilar is the basis of the process of classification. A clustering algorithm usually starts with the calculation of a matrix of similarities or distances between the entities. If the data is binary, i.e. the variables are of the type 'absence' or 'presence', similarity coefficients are used. They measure the relationship between the two entities, given the values of a set of $k$ variables common to both. In general, they take values between 0 and 1. For examples of similarity coefficients see Everitt (1980, Ch.2.3).

For quantitative data the similarity between two entities $x_i$ and $x_j$ with $k$ variables each is measured by distance metrics. The most commonly used distance measure in clustering methods is the $p^{th}$ order distance $\rho_p(x_i, x_j)$ as defined in Equation 2.1. Usually, the Euclidian metric ($p = 2$) is used. Since it is badly affected by changing the scale of a variable, often the entities $x_i = (x_{ik})_{k=1,..,v}$ are replaced by the standardized variable $(z_{ik})_{k=1,...,v} = (x_{ik}/\sigma_k)_{k=1,...,v}$, where $\sigma_k$ is the standard deviation of the $k$th variable. Another interesting distance measure that is also used in some clustering methods is the Mahalanobis distance

$$\rho(x_i, x_j) = (x_i - x_j)'\Sigma^{-1}(x_i - x_j). \tag{2.47}$$

Thereby, $\Sigma$ is the pooled within group variance-covariance matrix. The Mahalanobis distance has the advantage that it permits correlation between variables.

To evaluate the similarity between clusters instead of entities, the same metrics can be used by substituting the inter-individual coefficients $x$ with group means $\overline{x}$ in Equation 2.1 and 2.47. For other concepts of distance measures used in clustering methods we refer to Aldenderfer and Blashfield (1984, Ch.2).

### 2.8.2 Clustering algorithms

Since the field of application for clustering methods is huge, a large number of different algorithms have been developed. Everitt (1980, Ch.3) gives a comprehensive review of existing algorithms, where he classifies the algorithms into the five families of techniques given below. Other good reviews of clustering methods can be found in Aldenderfer and Blashfield (1984, Ch.2) or Cormack (1971).

**Hierarchical techniques** form the clusters stepwise by classifying the classes themselves into groups at different levels to form a tree. The direction of classification can be agglomerative or divisive, depending whether it is started with several groups with one entity each or one group with all entities.

**Optimization techniques** form the clusters by optimizing a certain clustering criterion.

**Density or mode-seeking techniques** form clusters by searching for regions with a relatively dense concentration of entities.

**Clumping techniques** form eventually overlapping clusters.

Hierarchical agglomerative methods are the most popular and simple ones. Among the hierarchical agglomerative methods the most popular methods are known as single-linkage, complete-linkage, and average-linkage. The basic procedure for all of them is the following. Suppose the set of $n$ entities $\{x_1, ..., x_n\}$ to be classified. Initially each entity forms its own cluster. In any particular step the methods merge those two currently existing clusters $G_a$ and $G_b$ that have the smallest distance $D(G_a, G_b)$, where $D(G_a, G_b)$ varies for the different methods. Each fusion decreases the number of clusters by one, until all clusters are merged into one large group.

Let us specify the distance $D(G_a, G_b)$ for the mentioned methods. Suppose $\rho(x_i, x_j)$ denotes any choice of distance measure, as presented in subsection 2.8.1.

**Single linkage method (nearest neighbor method)** In each step those clusters are merged that have the smallest distance between their nearest members. Hence, the distance between the groups is defined as the distance between their closest members, formally:

$$D(G_a, G_b) = \min_{x_i \in G_a, x_j \in G_b} \rho(x_i, x_j). \tag{2.48}$$

The single linkage rule states that it is enough that at least one member of the existing cluster is of the same level of similarity as the cluster that is to be merged.

**Complete linkage method (furthest neighbor method)** As the second name indicates, this method is exactly the opposite of the single linkage method. The distance between groups is defined as the distance between their most remote pair of entities:

$$D(G_a, G_b) = \max_{x_i \in G_a, x_j \in G_b} \rho(x_i, x_j). \tag{2.49}$$

In contrast to single linkage, the idea of this rule is that any candidate that is included into an existing cluster must be within a certain level of similarity to all members of that cluster.

**Average linkage method** This method was proposed as a compromise to the extremes of single and complete linkage. There are a number of variants of the method. Each essentially computes an average of the similarity of any candidate under consideration with all candidates in the existing cluster and, subsequently, merges the candidates to that cluster if a given level of similarity is achieved using the average value. The most commonly used variant of average linkage computes the arithmetic mean of similarities among the candidates.

$$D(G_a, G_b) = \frac{1}{n_a \cdot n_b} \cdot \sum_{x_i \in G_a} \sum_{x_j \in G_b} \rho(x_i, x_j). \tag{2.50}$$

Other variants are designed to calculate the similarity between the centroids of two clusters that might be merged, often named centroid method, i.e.

$$D(G_a, G_b) = \rho(\overline{x_a}, \overline{x_b}), \text{ where } \overline{x_a} = \frac{1}{n_a} \cdot \sum_{x_i \in G_a} x_i \text{ and } \overline{x_b} = \frac{1}{n_b} \cdot \sum_{x_j \in G_b} x_j. \tag{2.51}$$

The sequence of successive fusions of the entities can be represented visually by a tree-diagram, usually called dendrogram (see Figure 2.5). Each step where a pair of cases is merged is represented as a branch in this tree. At the lowest level, all entities

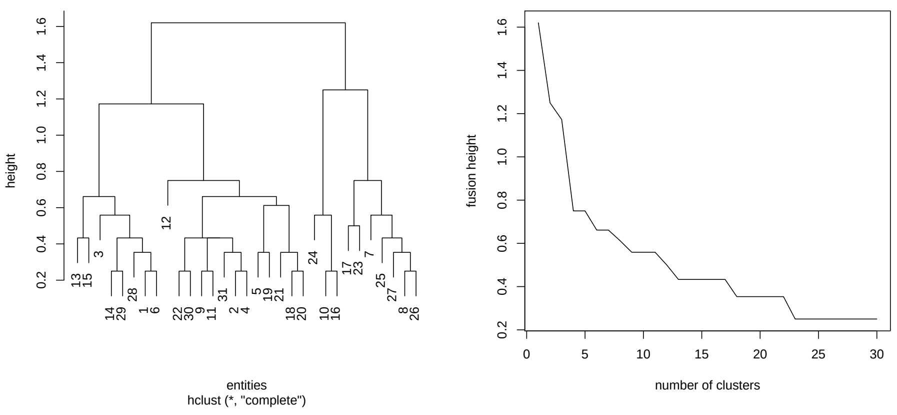


Figure 2.5: Exemplary dendrogram (left) and scree plot (right) for Complete Linkage

are independent. At the highest level, all entities are merged into one large cluster. In the dendrogram the height of a horizontal line is the distance $\rho$ between them, where $\rho$ varies between the different cluster methods as described before.

### 2.8.3 The optimal number of clusters and validation techniques

The nested structure of a dendrogram suggests that many different groups may be present in the data. A yet unsolved problem of cluster analysis is the definition of the optimal number of groups in this data. The reason is the lack of a suitable null hypothesis and the complex nature of multivariate sampling distributions. A heuristical approach is to look at the dendrogram and to search for a point, where adding another cluster would decrease the homogeneity of the clusters strongly. Branches in the dendrogram that are merged late, i.e. large vertical lines, state that their corresponding clusters are inhomogeneous. A way to find such stages more easily is to plot the height of fusion against the number of clusters. Analogous to the scree test of factor analysis, a good partition of the data can be found to the left of a marked ‘flattening’. Such a marked ‘flattening’ in the graph suggests that no new information is constituted by the following fusions of clusters. A jump in the graph implies that two relatively dissimilar clusters have been merged. Thus, the optimal number of cluster should lie to the right of a jump. Figure 2.5 shows a dendrogram and its corresponding scree plot. Here, a good choice would be to take four clusters.

Note, that the validation of cluster analysis is difficult due to the same problems as the choice of the optimal number. In Jain and Dubes (1988, Ch.4) and Aldenderfer and Blashfield (1984, Ch.4) some approaches are suggested, but not used here. In sequential optimization clustering is used to reduce the number of potential updating points. Miss-classification does not cause serious problems. Only the number of optimization iterations that is needed to find the global optimum may raise.

# 3 mtEGO - A novel approach to multi-objective target-value optimization

In this chapter we present a novel multivariate global optimization algorithm for target-value problems. As the procedure of the new algorithm follows the concepts of EGO (for details on EGO see Section 2.5) it is named mtEGO ($\underline{\text{m}}$ultivariate $\underline{\text{t}}$arget-value EGO). We start with a small initial space-filling design and fit a Kriging model as surrogate model. This model is refined stepwise by adding new points until a stopping criterion is reached. The infill sampling criterion that decides on the updating points is a variant of the expected improvement criterion. The novelty of the new variant is its ability to account for target-value problems using transformations of the objectives to two-sided desirabilities (for details on the concept of desirabilities see Section 2.6).

Henkenjohann et al. (2005) introduce a multivariate expected improvement criterion using the concept of desirability (Section 2.7). They transform the multi-objective problem to a single-objective problem with the help of a desirability index and then use the expected improvement criterion as introduced by Jones et al. (1998). The criterion needs to incorporate information about the model uncertainty which is now reached using the conditional density of the vector of desirabilities. This conditional density is known in a closed form only for some very special cases (cf. Section 2.6.3). Therefore, the approach by Henkenjohann et al. is restricted to one-sided Harrington desirabilities and can not be used straightforward with two-sided desirabilities. For length of computation time, a Monte-Carlo Simulation of the distribution is also not feasible. However, the approach of Henkenjohann et al. gave the basic idea for the new algorithm. The new algorithm also uses desirability functions and indices to transform the multi-objective problem into a single-objective one. We also keep up the idea that the conditional density of the vector of desirabilities must be incorporated into the algorithm to regard the original uncertainty of the surrogate after all the transformations are done. In contrast to Henkenjohann et al., we do not need the conditional density of the vector of desirabilities in a closed form or a precise estimation of it. It is replaced by a very rough approximation using so-called virtual observations and a combination and clustering strategy to find appropriate updating points.

## 3.1 Virtual observations

Instead of deriving or fully simulating the conditional distribution of the desirability vector, we calculate a very rough approximation of the uncertainty distribution of the untransformed predictions by virtual observations. Similarly to Cox and John (1997) in Section 2.4, we construct virtual observations by means of $(1-\alpha)\%$ confidence intervals for the predictions from the fitted surrogate model using the corresponding prediction errors. Suppose $\hat{y}(x)$ is the prediction of $y(x)$ from the surrogate and $s_{\hat{y}}^2(x)$ is the prediction error. Assuming the predictions of the surrogate model are approximately normally distributed, which is true for Kriging models, the $(1-\alpha)\%$ confidence interval of the prediction $\hat{y}$ can be determined by

$$[\hat{y}(x) - s_{\hat{y}}^2(x) \cdot t_{1-\alpha/2,df} \ , \ \hat{y}(x) + s_{\hat{y}}^2(x) \cdot t_{1-\alpha/2,df}, ] \tag{3.1}$$

where $t_{1-\alpha/2,df}$ is the $(1-\alpha/2)-$quantile of the t-distribution with $df$ degrees of freedom.

The left hand side of Figure 3.1 shows the 99.5% confidence interval for one exemplary predicted point $(\hat{y}_1(x_0), \hat{y}_2(x_0))$. The prediction, the lower and the upper confidence boundary are determined for each objective $y_1$ and $y_2$. Matching each of the three values for objective $y_1$ with all three values from objective $y_2$ gives nine points, including the original prediction, that are displayed in the figure. Each of these nine points represents one possible true result $(y_1(x_0), y_2(x_0))$ and is therefore called a virtual observation. The set of nine virtual observations gives a very rough impression of the impact of the model uncertainty on the predictions.

To get a better impression, virtual observations for several levels of different $(1-\alpha)\%$ confidence intervals can be used simultaneously. The right hand side of Figure 3.1 shows virtual observations for three levels $\alpha \in \{0.1, 0.05, 0.005\}$, resulting in 90%, 95% and 99.5% confidence intervals. Of course a large number of levels would yield in a full Monte-Carlo simulation of the model uncertainty distribution. But for $a$ confidence levels and $b$ objectives $a^b$ virtual observations have to be determined for each predicted

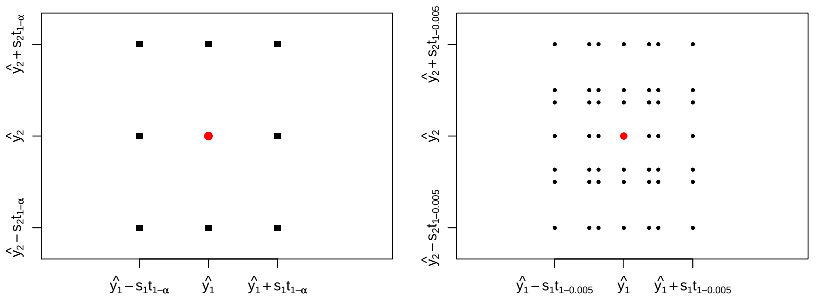


Figure 3.1: Virtual observations for $\alpha = 0.005$ (left) and $\alpha \in \{0.1, 0.05, 0.005\}$ (right)

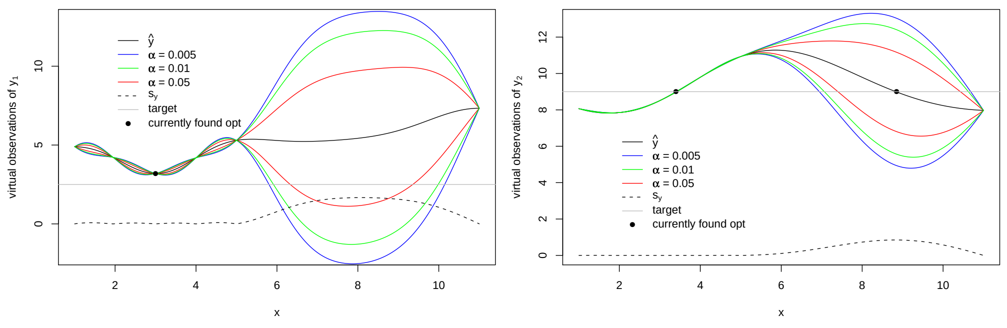


Figure 3.2: Exemplary virtual observation vectors for two $\alpha$-levels and three objectives $y_1$ (left) and $y_2$ (right)

point. The following procedure becomes slow and computationally expensive for larger values of $a$ and $b$.

For the optimization of a whole parameter space, we are not interested in virtual observations for only one predicted point, but in the vectors of virtual observations for the whole parameter space. For better illustration, suppose there are two objectives $y_1$ and $y_2$ and only one influencing parameter $x$. Figure 3.2 shows virtual observations of three confidence interval levels for the univariate parameter space $x = (1.0, 1.1, 1.2, ..., 11)'$. The predictions are quite certain in the interval [1,5] and very uncertain in the interval [5,11]. The target is represented by the horizontal line. The current local optimum for $y_1$ lies in the space with small model uncertainty. While the predictions (solid black line) do not indicate any potential for an optimum in the uncertain area, the virtual observations (colored lines) reveal a high potential of containing global optima in different places. If we take one vector of virtual observations the point where it meets the target or is at least the closest to the target can possibly be one true optimum

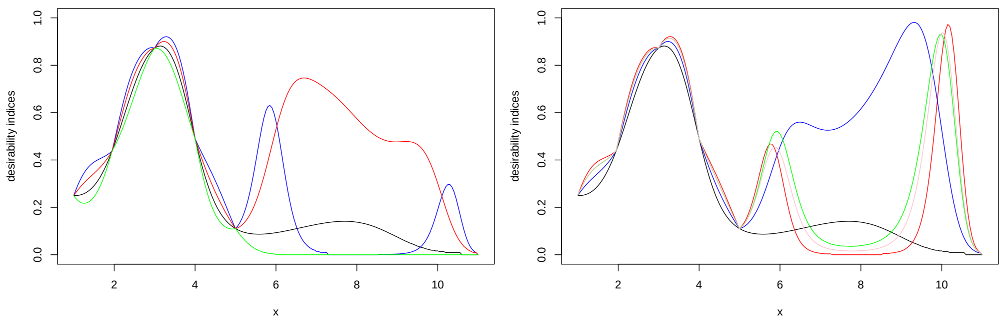


Figure 3.3: Exemplary resulting vectors of desirability indices

within the confidence level of this virtual observation. The set of all optima of the single virtual observation vectors thus represents the impact of the model uncertainty on the possible optimum. The same applies to objective $y_2$. The new information from the virtual observations for each single objective has to be combined to find the multivariate global objective. We use the concept of desirabilities to transform the problem into a univariate one. All vectors of virtual observations are transformed using beforehand specified (two-sided) desirability functions, resulting in desirability vectors. The desirability vectors of all objectives are cross-combined with each other and aggregated to desirability indices, such that the index is computed for each virtual observation displayed on the right hand side of Figure 3.1. Hence, the result are $a^b$ different desirability index values for every parameter setting $x$. Figure 3.3 shows some resulting desirability indices for the example. The left hand side of the figure shows some indices having their maximum very close to the currently found optimum. The right hand side of the figure shows indices that have their maximum in another region than the currently found optimum. The single desirabilities for the certain combination of virtual observations that leads to one of these indices are both large in this region. Hence, the points maximizing these indices have a high potential for a joint global optimum. Each desirability index is then transformed to an improvement vector, as introduced by Jones et al. (1998), resulting in one improvement vector per desirability index. Figure 3.4 shows the improvement vectors that result from the desirability indices displayed in Figure 3.3. The peaks in these plots represent the potential regions for a global optimum.

The expected improvement could now be estimated taking the arithmetic mean of all improvement values that have been determined through the cross-combination of

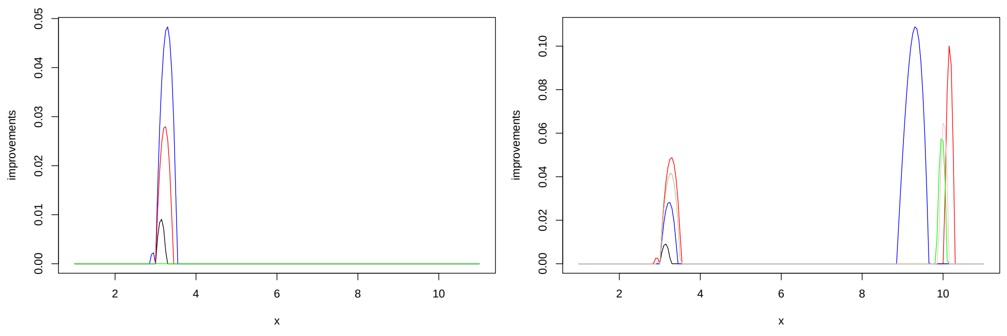


Figure 3.4: The improvement vectors resulting from the desirability indices from Figure 3.3

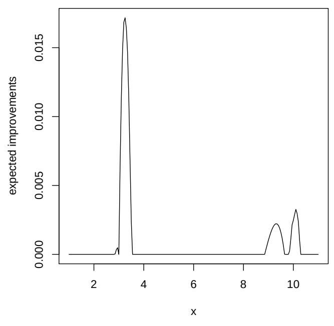


Figure 3.5: The expected improvement vector

virtual observations for each observation. The resulting vector of average improvements then is a rough estimation of the expected improvement. The maximum of the estimated expected improvement would be the updating point. But taking the average masks all the information about the tails of the distribution that we could gain by constructing the virtual observations. For the example used in Figures 3.2 to 3.4 the estimated expected improvement vector, i.e. the average of all improvement vectors, results in the curve displayed in Figure 3.5. Since 20 from all 25 indices look similar to the left hand side of Figure 3.4, the peak close to the currently found optimum is dominating such that the expected improvement also has its peak close to the currently found optimum. The information from the five indices on the right hand side of the figure is masked. Unfortunately, these are the important indices showing possible other optima in the uncertain areas. Only if we use only very small $\alpha$ levels there would be a chance that the expected improvement gets larger in an uncertain area than in the local optimum. However, this would be a very inappropriate parameterization.

In order to solve that problem, our heuristic proceeds with the separate improvement vectors, like other hybrid sequential algorithms do and "average" the information only in the end using hierarchical clustering. We determine the parameter settings $x$ that maximize the improvement vectors and thus get a set of points that we call the candidates for the global optimum. The set of candidates represents the impact of the model uncertainty on the possible location of the global optimum. In order to minimize the number of candidate points and since some of the candidates often lie close to each other or occur several times, we use standard hierarchical clustering to determine groups of candidate points. Finally, a representative point from each clustered group is taken as updating point to refine the model. A precise step-by-step instruction of the algorithm is given in the next subsection. Note, that without taking the average of the several improvements the name "expected improvement" is formally wrong, since no expectation of the improvements is determined.

## 3.2 The mtEGO algorithm step-by-step

In the following we give a detailed step-by-step introduction to the new mtEGO approach. The algorithm follows the same scheme like most of the sequential optimization procedures. The single steps are

1. Generate and run an initial space-filling design.
2. Fit a surrogate model that provides predictions and normally distributed prediction errors. Preferably a Kriging model.
3. Locate new design points based on the observations and the fitted surrogate model.
4. Evaluate the stopping criterion.
5. Update the surrogate model with the new design points and repeat Step 2 to 4 until the stopping criterion is fulfilled.

Suppose we have a multi-objective problem with $b$ objectives $y_1, y_2, ..., y_b$ and $p$ influencing parameters. Let $G \subset \chi$ be a set of $q$ feasible parameter settings representing the parameter space $\chi \subset \mathbb{R}^p$. Let further be $A$ the $n$ point initial design determined in Step 1, where $A \subset G$. Based on the design $A$, a Kriging model is fitted in Step 2 for each objective separately providing the algorithm with $b$ predictions and prediction errors for all parameter settings $x \in G$. Let $\hat{y}_1(x), \hat{y}_2(x), ...\hat{y}_b(x), \forall x \in G$ be the predictions and $s^2_{\hat{y}_1}(x), s^2_{\hat{y}_2}(x), ..., s^2_{\hat{y}_b}(x), \forall x \in G$ be the prediction errors, that are estimated by the empirical BLUP and MSPE in the Kriging Model (cf. Section 2.3.2). The location of new design points in Step 3 now is different from other existing algorithms. It can be partitioned in three phases: (A) estimation of the rough impression of the distribution, (B) determination of candidates for the global optimum, (C) reduction of the candidates to a small set of updating points. Each of these phases consists of several steps, that are explained in the following. Therefore, for each objective $y_1, y_2, ..., y_b$ a separate desirability function $d^{(1)}(.), d^{(2)}(.), ...., d^{(b)}(.)$ must be specified and a desirability index $DI[d^{(1)}, d^{(2)}, ..., d^{(b)}]$ (usually the geometric mean) must be selected.

**Phase 3.A:**

This phase is used to determine a small number of virtual observations to get a rough impression of the uncertainty distribution of the surrogate model.

3.A.1 The levels of the confidence intervals $\{\alpha_1, \alpha_2, ...., \alpha_a\}$ with $\alpha_i \in (0, 1]$ are chosen, depending on how local or global the location of the new updating points shall be. The effect of different choices of $\alpha$ is demonstrated in Section 3.5 and a reasonable parameterization for the whole set of $\alpha$ levels is discussed as a result of the simulation study in Section 4.2.

3.A.2 The vectors of virtual observations for each vector of predictions $\hat{y}_1(x), \hat{y}_2(x), ..., \hat{y}_b(x), \forall x \in G$ are determined according to the following schema:

$$\begin{aligned}
&\tilde{y}_1(x) = \hat{y}_1(x),\\
&\tilde{y}_{11+}(x) = \hat{y}_1(x) + t_{1-\alpha_1/2,df} \cdot \sqrt{s^2_{\hat{y}_1}(x)}, \quad \tilde{y}_{11-}(x) = \hat{y}_1(x) - t_{1-\alpha_1/2,df} \cdot \sqrt{s^2_{\hat{y}_1}(x)},\\
&\tilde{y}_{12+}(x) = \hat{y}_1(x) + t_{1-\alpha_2/2,df} \cdot \sqrt{s^2_{\hat{y}_1}(x)}, \quad \tilde{y}_{12-}(x) = \hat{y}_1(x) - t_{1-\alpha_2/2,df} \cdot \sqrt{s^2_{\hat{y}_1}(x)}\\
&...\\
&\tilde{y}_{1a+}(x) = \hat{y}_1(x) + t_{1-\alpha_a/2,df} \cdot \sqrt{s^2_{\hat{y}_1}(x)}, \quad \tilde{y}_{1a-}(x) = \hat{y}_1(x) + t_{1-\alpha_a/2,df} \cdot \sqrt{s^2_{\hat{y}_1}(x)}\\
&\tilde{y}_2(x) = \hat{y}_2(x),\\
&\tilde{y}_{21+}(x) = \hat{y}_2(x) + t_{1-\alpha_1/2,df} \cdot \sqrt{s^2_{\hat{y}_2}(x)}, \quad \tilde{y}_{21-}(x) = \hat{y}_2(x) - t_{1-\alpha_1/2,df} \cdot \sqrt{s^2_{\hat{y}_2}(x)},\\
&\tilde{y}_{22+}(x) = \hat{y}_2(x) + t_{1-\alpha_2/2,df} \cdot \sqrt{s^2_{\hat{y}_2}(x)}, \quad \tilde{y}_{22-}(x) = \hat{y}_2(x) - t_{1-\alpha_2/2,df} \cdot \sqrt{s^2_{\hat{y}_2}(x)}\\
&...\\
&\tilde{y}_{2a+}(x) = \hat{y}_2(x) + t_{1-\alpha_a/2,df} \cdot \sqrt{s^2_{\hat{y}_2}(x)}, \quad \tilde{y}_{2a-}(x) = \hat{y}_2(x) + t_{1-\alpha_a/2,df} \cdot \sqrt{s^2_{\hat{y}_2}(x)}\\
&...
\end{aligned}$$

$\tilde{y}_b(x) = \hat{y}_b(x)$,
$\tilde{y}_{b1+}(x) = \hat{y}_b(x) + t_{1-\alpha_1/2,df} \cdot \sqrt{s^2_{\hat{y}_b}(x)}$, $\tilde{y}_{b1-}(x) = \hat{y}_b(x) - t_{1-\alpha_1/2,df} \cdot \sqrt{s^2_{\hat{y}_b}(x)}$,
$\tilde{y}_{b2+}(x) = \hat{y}_b(x) + t_{1-\alpha_2/2,df} \cdot \sqrt{s^2_{\hat{y}_b}(x)}$, $\tilde{y}_{b2-}(x) = \hat{y}_b(x) - t_{1-\alpha_2/2,df} \cdot \sqrt{s^2_{\hat{y}_b}(x)}$
...
$\tilde{y}_{ba+}(x) = \hat{y}_b(x) + t_{1-\alpha_a/2,df} \cdot \sqrt{s^2_{\hat{y}_b}(x)}$, $\tilde{y}_{ba-}(x) = \hat{y}_b(x) + t_{1-\alpha_a/2,df} \cdot \sqrt{s^2_{\hat{y}_b}(x)}$
Altogether, we get $(2ba + b)$ vectors of virtual observations.

3.A.3 Each of the vectors from 3.A.2 is transformed into desirabilities according to the corresponding desirability functions $d^{(1)}, d^{(2)}, ...., d^{(b)}$.
$d_1(x) = d^{(1)}(\tilde{y}_1(x))$,
$d_{11+}(x) = d^{(1)}(\tilde{y}_{11+}(x))$, $d_{11-}(x) = d^{(1)}(\tilde{y}_{11-}(x))$
$d_{12+}(x) = d^{(1)}(\tilde{y}_{12+}(x))$, $d_{12-}(x) = d^{(1)}(\tilde{y}_{12-}(x))$
...
$d_{1a+}(x) = d^{(1)}(\tilde{y}_{1a+}(x))$, $d_{1a-}(x) = d^{(1)}(\tilde{y}_{1a-}(x))$
$d_2(x) = d^{(2)}(\tilde{y}_2(x))$,
$d_{21+}(x) = d^{(2)}(\tilde{y}_{21+}(x))$, $d_{21-}(x) = d^{(2)}(\tilde{y}_{21-}(x))$
$d_{22+}(x) = d^{(2)}(\tilde{y}_{22+}(x))$, $d_{22-}(x) = d^{(2)}(\tilde{y}_{22-}(x))$
...
$d_{2a+}(x) = d^{(2)}(\tilde{y}_{2a+}(x))$, $d_{2a-}(x) = d^{(2)}(\tilde{y}_{2a-}(x))$
...
$d_b(x) = d^{(b)}(\tilde{y}_b(x))$,
$d_{b1+}(x) = d^{(b)}(\tilde{y}_{b1+}(x))$, $d_{b1-}(x) = d^{(b)}(\tilde{y}_{b1-}(x))$
$d_{b2+}(x) = d^{(b)}(\tilde{y}_{b2+}(x))$, $d_{b2-}(x) = d^{(b)}(\tilde{y}_{b2-}(x))$
...
$d_{ba+}(x) = d^{(b)}(\tilde{y}_{ba+}(x))$ $d_{ba-}(x) = d^{(b)}(\tilde{y}_{ba-}(x))$
for all $x \in G$. The result of 3.A.3 are $(2ba + b)$ vectors of desirabilities.

3.A.4 The desirability vectors of 3.A.3 are aggregated to desirability indices. Therefore, one desirability vector per objective is selected from the set in 3.A.3 and then aggregated to the desirability index using the (weighted) geometric mean. Each desirability vector belonging to $y_1$ is combined with each belonging to $y_2$ with each from $y_3$ ... with each from $y_b$. The $b$ desirability vectors to be combined to one index are selected according to the following cross-combination scheme (for simplicity we use the short notation $d_{b1+}$ for the desirability vector $d_{b1+}(x), \forall x \in G$):

$DI[d_1, d_2, ..., d_b]$,

$DI[d_{11+}, d_2, ..., d_b]$, $DI[d_{11-}, d_2, ..., d_b]$, ... $DI[d_{1a+}, d_2, ..., d_b]$, $DI[d_{1a-}, d_2, ..., d_b]$

$DI[d_1, d_{21+}, ..., d_b]$, $DI[d_1, d_{21-}, ..., d_b]$, ... $DI[d_1, d_{2a+}, ..., d_b]$, $DI[d_1, d_{2a-}, ..., d_b]$

...

$DI[d_1, d_2, ..., d_{b1+}]$, $DI[d_1, d_2, ..., d_{b1-}]$, ... $DI[d_1, d_2, ..., d_{ba+}]$, $DI[d_1, d_2, ..., d_{ba-}]$

$DI[d_{11+}, d_{21+}, ..., d_b]$, $DI[d_{11-}, d_{21+}, ..., d_b]$, ... $DI[d_{1a+}, d_{21+}, ..., d_b]$, $DI[d_{1a-}, d_{21+}, ..., d_b]$

$DI[d_{11+}, d_{21-}, ..., d_b]$, $DI[d_{11-}, d_{21-}, ..., d_b]$, ... $DI[d_{1a+}, d_{21-}, ..., d_b]$, $DI[d_{1a-}, d_{21-}, ..., d_b]$

$DI[d_{11+}, d_{22+}, ..., d_b]$, $DI[d_{11-}, d_{22+}, ..., d_b]$, ... $DI[d_{1a+}, d_{22+}, ..., d_b]$, $DI[d_{1a-}, d_{22+}, ..., d_b]$

$DI[d_{11+}, d_{22-}, ..., d_b]$, $DI[d_{11-}, d_{22-}, ..., d_b]$, ... $DI[d_{1a+}, d_{22-}, ..., d_b]$, $DI[d_{1a-}, d_{22-}, ..., d_b]$

...

$DI[d_{11+}, d_{2a+}, ..., d_b]$, $DI[d_{11-}, d_{2a+}, ..., d_b]$, ... $DI[d_{1a+}, d_{2a+}, ..., d_b]$, $DI[d_{1a-}, d_{2a+}, ..., d_b]$

$DI[d_{11+}, d_{2a-}, ..., d_b]$, $DI[d_{11-}, d_{2a-}, ..., d_b]$, ... $DI[d_{1a+}, d_{2a-}, ..., d_b]$, $DI[d_{1a-}, d_{2a-}, ..., d_b]$

...

$DI[d_{11+}, d_{21+}, ..., d_{b1+}]$, $DI[d_{11-}, d_{21+}, ..., d_{b1+}]$, ... $DI[d_{1a+}, d_{21+}, ..., d_{b1+}]$, $DI[d_{1a-}, d_{21+}, ..., d_{b1+}]$

$DI[d_{11+}, d_{21-}, ..., d_{b1-}]$, $DI[d_{11-}, d_{21-}, ..., d_{b1-}]$, ... $DI[d_{1a+}, d_{21-}, ..., d_{b1-}]$, $DI[d_{1a-}, d_{21-}, ..., d_{b1-}]$

$DI[d_{11+}, d_{22+}, ..., d_{b1+}]$, $DI[d_{11-}, d_{22+}, ..., d_{b1+}]$, ... $DI[d_{1a+}, d_{22+}, ..., d_{b1+}]$, $DI[d_{1a-}, d_{22+}, ..., d_{b1+}]$

$DI[d_{11+}, d_{22-}, ..., d_{b1-}]$, $DI[d_{11-}, d_{22-}, ..., d_{b1-}]$, ... $DI[d_{1a+}, d_{22-}, ..., d_{b1-}]$, $DI[d_{1a-}, d_{22-}, ..., d_{b1-}]$

...

$DI[d_{11+}, d_{2a+}, ..., d_{b1+}]$, $DI[d_{11-}, d_{2a+}, ..., d_{b1+}]$, ... $DI[d_{1a+}, d_{2a+}, ..., d_{b1+}]$, $DI[d_{1a-}, d_{2a+}, ..., d_{b1+}]$

$DI[d_{11+}, d_{2a-}, ..., d_{b1-}]$, $DI[d_{11-}, d_{2a-}, ..., d_{b1-}]$, ... $DI[d_{1a+}, d_{2a-}, ..., d_{b1-}]$, $DI[d_{1a-}, d_{2a-}, ..., d_{b1-}]$

$DI[d_{11+}, d_{21+}, ..., d_{b2+}]$, $DI[d_{11-}, d_{21+}, ..., d_{b2+}]$, ... $DI[d_{1a+}, d_{21+}, ..., d_{b2+}]$, $DI[d_{1a-}, d_{21+}, ..., d_{b2+}]$

$DI[d_{11+}, d_{21-}, ..., d_{b2-}]$, $DI[d_{11-}, d_{21-}, ..., d_{b2-}]$, ... $DI[d_{1a+}, d_{21-}, ..., d_{b2-}]$, $DI[d_{1a-}, d_{21-}, ..., d_{b2-}]$

$DI[d_{11+}, d_{22+}, ..., d_{b2+}]$, $DI[d_{11-}, d_{22+}, ..., d_{b2+}]$, ... $DI[d_{1a+}, d_{22+}, ..., d_{b2+}]$, $DI[d_{1a-}, d_{22+}, ..., d_{b2+}]$

$DI[d_{11+}, d_{22-}, ..., d_{b2-}]$, $DI[d_{11-}, d_{22-}, ..., d_{b2-}]$, ... $DI[d_{1a+}, d_{22-}, ..., d_{b2-}]$, $DI[d_{1a-}, d_{22-}, ..., d_{b2-}]$

...

$DI[d_{11+}, d_{2a+}, ..., d_{ba+}]$, $DI[d_{11-}, d_{2a+}, ..., d_{ba+}]$, ... $DI[d_{1a+}, d_{2a+}, ..., d_{ba+}]$, $DI[d_{1a-}, d_{2a+}, ..., d_{ba+}]$

$DI[d_{11+}, d_{2a-}, ..., d_{ba-}]$, $DI[d_{11-}, d_{2a-}, ..., d_{ba-}]$, ... $DI[d_{1a+}, d_{2a-}, ..., d_{ba-}]$, $DI[d_{1a-}, d_{2a-}, ..., d_{ba-}]$

In total, there exist $(2a + 1)^b$ desirability indices, where each one represents a possible true desirability index within the uncertainty of the models. The desirability indices still are vectors depending on $x$ and should formally be noted as $DI[d_{1a-}(x), d_{2a-}(x), ..., d_{ba-}(x)], \forall x \in G$. This dependency is important to remember in the next phase but is now hidden in the short notation.

**Phase 3.B:**
Having a rough impression how the desirability index looks like within the whole range of uncertainty, potential candidates for the global optimum are determined.

3.B.1 Each vector of desirability indices from 3.A.4 is transformed into a vector of improvement values using Equation 2.44 with $g = 1$.

$$I(x) = \begin{cases} (DI(x) - DI_{max}) & \text{, if } DI(x) > DI_{max} \\ 0 & \text{, if } DI(x) \leq DI_{max} \end{cases} .$$

(Using another value of $g$ is not reasonable since the choice of the $\alpha$ levels enables to control how local or global the search will be. If $g$ would be varied too, the parameterization of $\alpha$ becomes difficult.)
The currently maximum observed desirability index here is

$$DI_{max} = \max_{x \in A} DI[d^{(1)}(y_1(x)), ..., d^{(b)}(y_b(x))] \tag{3.2}$$

and $DI(x)$ is substituted subsequently by all desirability index vectors from 3.A.4. For the exemplary desirability index $DI(x) = DI[d_{11-}(x), d_2(x), ..., d_b(x)]$ the improvement is hence calculated as follows:

$$I_{DI[d_{11-},d_2,...,d_b]}(x) = \begin{cases} (DI[d_{11-}(x), d_2(x), ..., d_b(x)] - DI_{max}) \\ \quad \text{, if } DI[d_{11-}(x), d_2(x), ..., d_b(x)] > DI_{max} \\ 0, \quad \text{else} \end{cases} \tag{3.3}$$

Determining the improvements for all desirability indices from 3.A.4 thus results in the vectors of improvements
$I_{DI[d_1,d_2,...,d_b]}(x)$,
$I_{DI[d_{11+},d_2,...,d_b]}(x)$, $I_{DI[d_{11-},d_2,...,d_b]}(x)$,
...
$I_{DI[d_{1a+},d_{2a-},...,d_{ba-}]}(x)$, $I_{DI[d_{1a-},d_{2a-},...,d_{ba-}]}(x)$
In total, there exist $(2a+1)^b$ vectors of improvement, where each one represents a possible true vector of improvement within the uncertainty of the models.

3.B.2 For each vector of improvement $I_{DI}(x))$ from 3.B.1 the point $x$ is determined that maximizes it

$$x_c = \underset{x \in G}{\text{argmax}}\, I_{DI}(x)). \tag{3.4}$$

The resulting maximizing points $x_c$ are candidates for the potential true optimum and are collected in the set of candidate points $C$. If several points maximize an improvement vector, all of them are added to the set $C$, unless all values are 0. If all values for one improvement vector are 0 this states that no point can improve the currently found optimum for this possible true result. If all improvement vectors are 0 for all parameter settings $x$, the set $C$ is empty, which means that the currently found optimum can not be further improved at all and the algorithm stops.

**Phase 3.C:**
Reduction of the set of candidates to a small set of updating points.

3.C.1 First, the point $x^*$ with the best predictions with the current model fit, i.e.

$$x^* = \underset{x_i \in \{x_1,...,x_q\}}{\text{argmax}} \; DI[d^{(1)}(\hat{y}_1(x)), ..., d^{(b)}(\hat{y}_b(x))] \tag{3.5}$$

is determined. If $x^*$ is not included in the design yet, $x^*$ is included in the candidate set $C$ determined in 3.B.2, since it is the suggested candidate of the above mentioned index (a proof can be found in Appendix A.1). The point $x^*$ then becomes the first updating point and is excluded from the candidate set $C$. The remaining points are stored in the set $C^*$.
If $x^*$ is already included in the design, all improvements are 0 for the index $DI[d^{(1)}(\hat{y}_1(x)), ..., d^{(b)}(\hat{y}_b(x))]$ and no candidate is suggested then. Hence, $C^* = C$ in this case (proofed in Appendix A.1).
This procedure ensures that the currently best predicted point is an updating point and is used to refine the surrogate model, if it is not already available.

3.C.2 The remaining set of candidates $C^*$ is divided into groups with candidates that lie close to each other using any hierarchical clustering method described in Section 2.8.2. The number of groups defines the number of updating points and can be influenced directly by choosing the point where the dendrogram is cut. How to find the appropriate number of clusters automatically is described in Section 3.3.

3.C.3 For all groups formed in 3.C.2 the center points are chosen as representative candidate points. The center point $x_{cent}$ of group $G = \{x_1, ..., x_o\}$ $(x_1, ..., x_o \in C^*)$ is that point from the group that minimizes the Euclidian distances to the rest of the points within the group

$$x_{cent} = \underset{\substack{x_i \in G \\ i=1,\ldots,o}}{\operatorname{argmin}} \sum_{j=1}^{o} \left( \sum_{k=1}^{p} (x_{ik} - x_{jk})^2 \right)^{1/2} . \tag{3.6}$$

If there is more than one point minimizing the distances, we choose one of them at random. The representative points and eventually the point $x^*$ from 3.C.1 are the final updating points.

In Step 4 the stopping criterion, which is introduced in the next subsection, is evaluated. If it is not reached yet, the updating points determined in 3.C.3 are added to the design and the model is refined. mtEGO then starts over again with Step 2 to Step 4 until the stopping criterion is reached.

## 3.3 Choice of the number of clusters in the candidate set

As described in Section 2.8.3, the choice of the number of clusters is not trivial and a naturally right number of clusters does not exist. Since the number of clusters defines the number of updating points and we do not want to add a mass of points, we try to add as few points as needed without missing interesting regions by merging too different clusters. Section 2.8.3 describes how a scree graph can help finding a good cutting point. However, using the scree graph also always is a subjective choice and needs experience in using mtEGO to choose an appropriate number of clusters. We therefore present a criterion that chooses the number of clusters automatically for users that are not experienced with mtEGO. The criterion was developed using a kind of teach-back. In a first study mtEGO was used for different $\alpha$ levels and test problems and the number of clusters was chosen by an experienced user. Then different automatized criteria were developed and the criterion that chooses (almost) the same numbers of cluster as the expert is now presented. But as every automatization, a manual choice on the basis of long experience can yield in much better results. Therefore, it is sensible to supervise the scree graph, the dendrogram and the progress of the candidate set to get familiar with the decisions and detect eventually bad decisions in exceptional cases. In some cases one may prefer a larger number of updating points, since the model fit is bad. Or vice versa, one wants to choose only few clusters since the model is very precise and the progress of the candidates concentrates more and more on a certain region that obviously has to be the region of the optimum.

The basis of the criterion is the idea to choose the number of clusters where the scree graph has a marked ‘flattening’. To detect a ‘flattening’ automatically we need to know where the decrease changes the most. Suppose we have $n$ candidates and we build clusters by taking the least homogenous entity out of the cluster in each step starting with one group containing all clusters. Let $H(i), i = 1, ..., n$, be the fusion height when the $i$-th cluster is formed. $H(i)$ is the monotone decreasing function that is also displayed in the scree graph. The first derivative

$$H'(i) = H(i) - H(i-1), i = 2, ..., n,$$

gives the decrease of the fusion height between the fusion of the $(i-1)$-th and $(i)$-th cluster. The second derivative

$$H''(i) = H'(i+1) - H'(i) = H(i+1) - 2H(i) + H(i-1), i = 2, ..., n-1,$$

then gives the change in decrease of the fusion height between the fusion of the $(i-1)$-th and $i$-th cluster compared to the fusion of the $i$-th to the $(i+1)$-th cluster. The number of clusters $i$ that maximizes $H''(i)$ is hence the point with the greatest marked ‘flattening’ in the scree graph (see Figure 3.6).

Often the fusion height starts decreasing very strongly and flattens slowly before the first marked ’flattening’ can be seen in the scree graph. $H''(i)$ is then very large in the beginning although we would not see a marked ‘flattening’ in the graph. In Figure 3.7 for example $H''(i)$ is maximum for two groups, but the flattening and the dendrogram suggest to choose five clusters. To avoid choosing too few groups in such cases, we search the number of groups $i$ that maximizes $H''(i)$ under the condition that $H(i) < 0.5 \cdot H(1)$. With the new condition the criterion correctly chooses five groups in the example.

In the implementation of mtEGO, that is attached in Appendix A.6, the user is also allowed to limit the number of groups by definition of a maximum and a minimum. For the simulation study presented in Section 4 the limit is set to 10 groups at maximum, since we see no sense in adding more points than twice the dimension of the parameter space in each cycle. Few updating points per cycle most probably result in many cycles, which is not desirable in high dimensions due to long computing times per cycle. Note, that for the criterion presented here five candidates at minimum are needed to compute

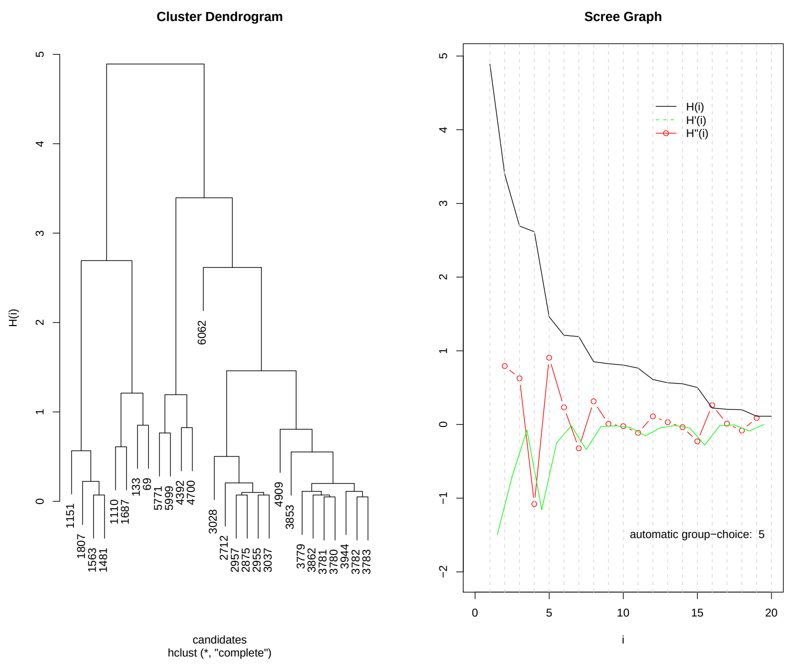


Figure 3.6: Exemplary automatized choice of number of clusters for dendrogram and scree plot

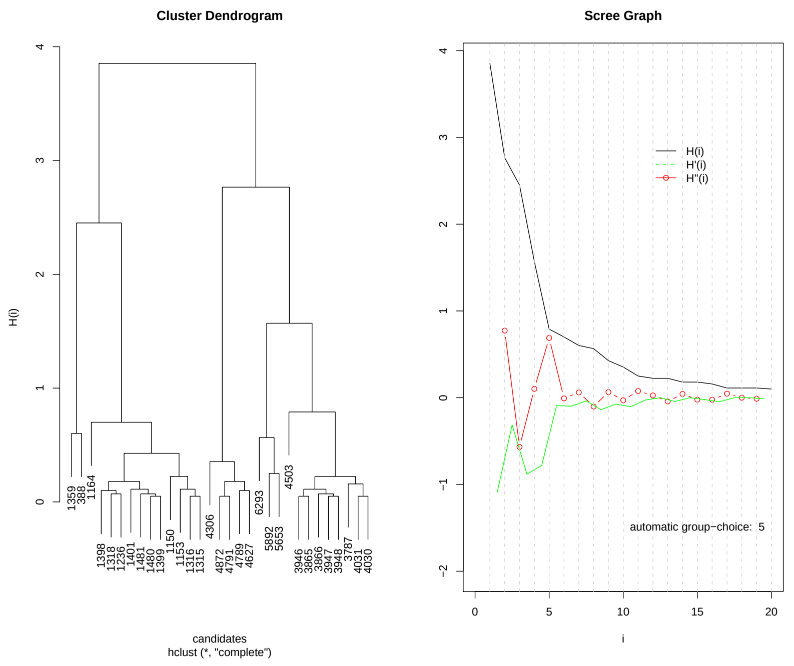


Figure 3.7: Example for automatized choice of number where too few groups must be avoided

$H''(i)$ and decide about the maximum. If less candidates are available, all can be taken as separate groups, unless some obvious cluster are present.

## 3.4 Stopping criterion for the new approach

Jones et al. (1998) use the maximum value of the expected improvement as stopping criterion. If the expected improvement gets too small their algorithm stops. As described in Section 3.2 the new heuristic uses a cross-combination of the desirability vectors from all virtual observations. Therefore, we do not have one maximum EI. The new stopping criterion has to take all indices into account. A good basis for a stopping criterion is to take the maximum of all improvement vectors determined in Step 3.B.2 and to choose the maximum among those maximums, formally:

$$maximax = \max_{\forall\ I_{DI}\ \text{in 3.B.2}} \left( \max_{\forall x \in \chi}\ I_{DI}(x) \right). \tag{3.7}$$

*maximax* is determined by the improvements, that depend on the desirability index, which depends only on the predictions. If the predictions only get more precise (i.e. the prediction error gets smaller) but change very slightly between two optimization steps, *maximax* changes only very slightly. An appropriate stopping criterion should take into account, that the predictions got better in the sense of precision although the predictions did not get better regarding the optimization targets. The prediction error should be taken into account. Therefore, we determine the maximum prediction error that is present in the current surrogate models, i.e. the maximum among the maximums of the standard prediction errors for each objective $(y_1, ..., y_b)$, formally:

$$maxerrmax = \max_{i=1,...,b} \left( \max_{\forall x \in \chi}\ s^2_{y_i}(x) \right). \tag{3.8}$$

The maximum improvement value present in the optimization *maximax* is then multiplied with the maximum available prediction error *maxerrmax* resulting in the stopping criterion:

$$relmaximax = maximax \cdot maxerrmax. \tag{3.9}$$

The algorithm may be stopped, if *relmaximax* reaches 0, i.e. no point, not even a virtual observation, reaches better results then the currently found. However, depending on the $\alpha$ levels it may need many points and iterations until 0 is reached. Therefore

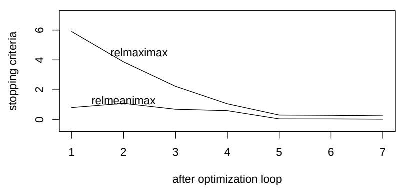


Figure 3.8: Exemplary progress of the stopping criteria

we propose to observe the progress of the criterion and stop if it saturates at a level close to 0, where the meaning of ‘close’ depends on the size of *relmaximax* after the first optimization loop. If it starts with a value of approximately 6, 0.2 can be seen as close to 0. If it starts with values of approximately 0.5, a saturation of 0.2 must be very clear to stop the optimization. Figure 3.8 shows a saturating progress that indicates to stop after the sixth optimization loop. Note, that there is no guarantee that *relmaximax* is monotonically decreasing. If the surrogate model is changed after adding updating points to the design because it was false in some regions, this may cause that *relmaximax* enlarges before it decreases again.

Sometimes, it makes sense to supervise also the progress of the average of the maximum improvements weighted by the prediction errors,

$$relmeanimax = 1/a^b \cdot \sum_{\forall\ I_{DI}\ \text{ in } 3.B.2} \left( \max_{\forall x \in \chi} I_{DI}(x) \right) \cdot maxerrmax, \tag{3.10}$$

where $a$ is the number of $\alpha$ levels used and $b$ is the number of objectives. Just like for *relmaximax*, the algorithm can be stopped if *relmeanimax* approaches 0 or saturates close to 0. For the same reason as *relmaximax*, the criterion *relmeanimax* is not necessarily monotonically decreasing.

The two criteria *relmaximax* and *relmeanimax* together give a good overview of the current capability of improvement of the optimization procedure. If *relmeanimax* already saturates close to 0, but *relmaximax* does not, this means that for the majority of desirability indices no more improvements can be reached. Only one or few indices are then able to improve the results and it is wise to check which confidence levels are responsible for this effect. If those are rather narrow intervals one could stop if the optimum satisfies the users quality requirements. If those are very wide intervals, we advise to proceed, since they may reveal a new optimum in a still uncertain region.

Additionally, it makes sense to supervise the progress of the set of candidates, the current best predicted point $x^*$ and the current observed optimum $x_{curopt}$. If the set of candidates concentrates around $x^*$ and $x^* = x_{curopt}$ at least for two iterations, the algorithm can be stopped, independently from the criterion $relmaximax$. Usually, $relmaximax$ then saturates too, but if $relmaximax$ is already very small in the first optimization cycle a saturation sometimes is hard to recognize. In such a case the supervision of the progress is a helpful tool.

Due to Step 3.C.1 of the algorithm the predicted optimum is always verified in the next optimization cycle. If the stopping criterion is reached, it is not necessary to verify the found optimum once again. This step is also very important to avoid that the algorithm may get stuck in a predicted optimum $x^*$ that is actually false or badly predicted. Because of Step 3.C.1 the predicted optimum will be an updating point in the next optimization cycle and will be corrected then. If the prediction is correct and no other better point is found, $x^*$ becomes $x_{curopt}$ after the next optimization cycle.

## 3.5 The effect of a certain $\alpha$ level on mtEGO

The choice of the $\alpha$ levels used during the optimization with mtEGO has of course an impact on the optimization process. With the help of the parameterization of the set of $\alpha$ levels in Step 3.A.1 of mtEGO the user may control how global or local the search is. The combination of different $\alpha$ levels that are used simultaneously enables mtEGO to search globally and locally at the same time. Nevertheless, one should know in general the effect of a certain $\alpha$ level to be able to choose the vector. In the following we try to illustrate the effect that a different size of one $\alpha$ level has. A guideline on the appropriate choice of the whole set of $\alpha$ levels is developed in the simulation study in Section 4.2.

In general, an optimization is more global the more attention is given to uncertain areas. Jones et al. (1998) use the factor $g$ in the generalized expected improvement criterion to control how much weight is given to the uncertainty distribution (cf. Section 2.5). In mtEGO the level of the $(1-\alpha)\%$ confidence interval controls how much attention is given to the uncertain areas. Figure 3.9 demonstrates how the level affects the search. The 20% confidence interval (i.e. $\alpha = 0.8$ virtual observations) has narrow confidence boundaries above and below the predictions even in the uncertain area $[5, 10]$. The corresponding desirability is small in the whole uncertain interval $[5, 10]$ and largest for $x = 3.05$, which is very close to the currently found optimum at $x = 3$.

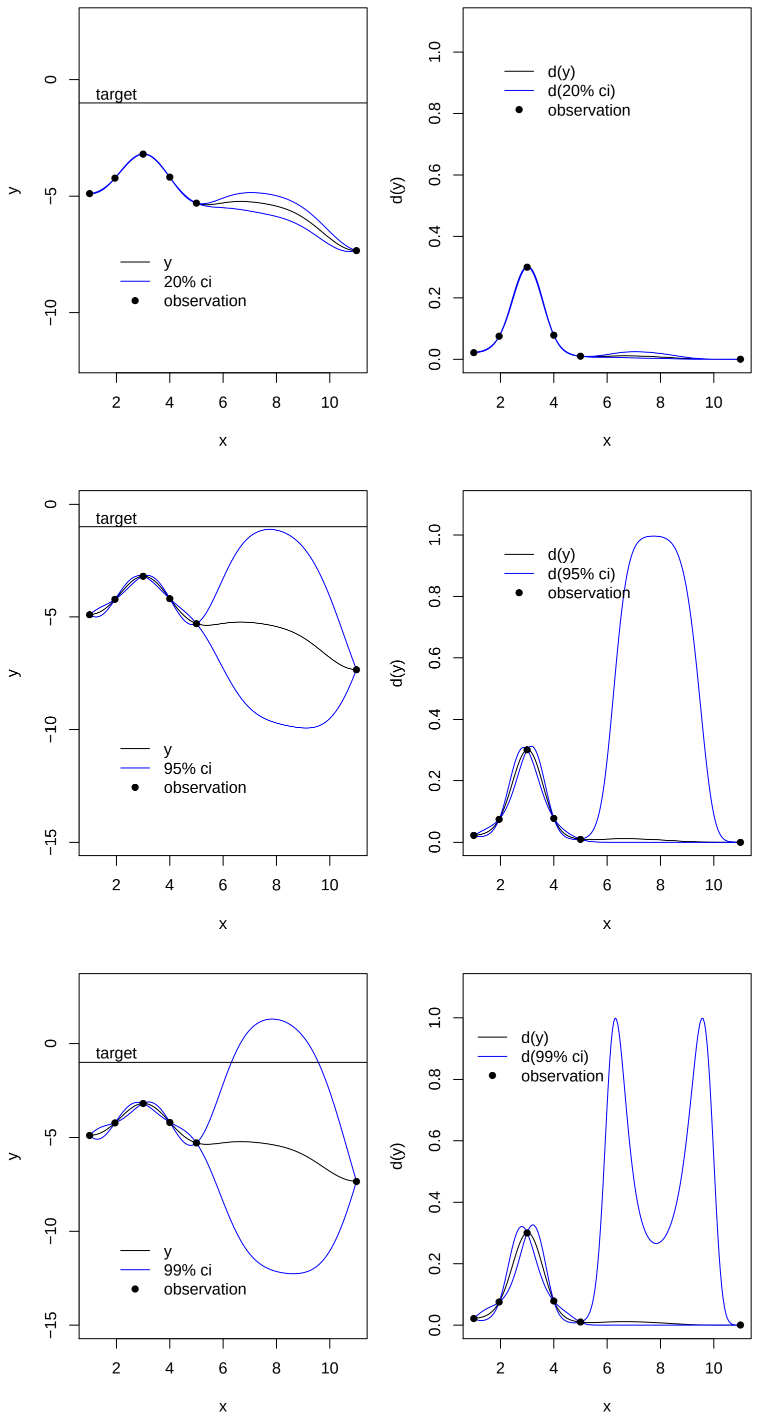


Figure 3.9: The effect of a certain choice of a $(1-\alpha)\%$ confidence interval (ci) on the optimization

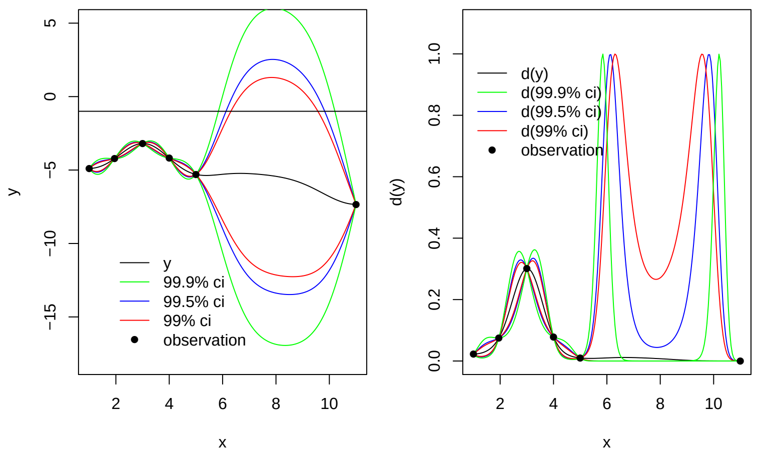


Figure 3.10: The effect of too small choices of $\alpha$ levels on the optimization

Suppose, we do a multi-objective optimization with mtEGO using only this $\alpha = 0.8$ level. We cross-combine the $\alpha = 0.8$ level virtual observations with all other $\alpha = 0.8$ level virtual observations from all other objectives. The resulting indices will most probably be maximum close to the currently found optimum. The suggested candidates will hence lie close to the currently found optimum. The optimization is very local then. In contrast, the 95% confidence interval, which is the $\alpha = 0.05$ virtual observation, forms wide boundaries around the predictions in the uncertain area $[5, 10]$. The upper boundary almost touches the target. Now, the desirability of the upper boundary is maximum inside the uncertain space. Using 99% confidence intervals, i.e. a small $\alpha = 0.01$ level virtual observation, the upper boundary even exceeds the target and hence the desirability reaches 1 in the two points where the target is met. The desirability around the currently found optimum at $x = 3$ is much smaller than in the uncertain area. Suppose again we do a multi-objective optimization with mtEGO using only this $\alpha = 0.01$ level. We then cross-combine the $\alpha = 0.01$ level virtual observations with all other $\alpha = 0.01$ level virtual observations from all other objectives. The resulting indices have maxima in the uncertain area then and suggest candidates there. The search is more global now.

Figure 3.10 demonstrates a problem that occurs if too small $\alpha$ values are used for mtEGO. If one compares the 99%, the 99.5%, and the 99.9% confidence interval, one can see that the points $x$ where the boundaries meet the target are lying closer to the observations the wider the confidence interval is. This means the smaller $\alpha$ is chosen the closer the suggested candidates for the optimum lie to the already available observations. If only very small $\alpha$ values are used for an optimization with mtEGO the suggested candidates and, hence, also the updating points will all lie next to the already observed design points. We call this phenomenon clumping of the updating points. Clumping is only desirable around those observations with high desirability, around the other observations it is an undesired effect. An updating point lying very close to an existing observation with small desirability yields no benefit and is a waste of resources and time.

Summarizing the previous demonstration one can say that using mtEGO with small $\alpha$ levels (i.e. wide confidence intervals) yields a global search and using large $\alpha$ levels (i.e. narrow confidence intervals) yields a local search. The idea of mtEGO is to use several $\alpha$ levels simultaneously, which enables to combine small and large $\alpha$ levels to account for a global and local search at the same time. As already mentioned, the appropriate number and combination of $\alpha$ levels is studied with several simulations in Section 4.2.

## 3.6 Implementation

The new algorithm mtEGO has been implemented with the software **R** (Version 2.8.1), a state-of-the-art, freely available statistical software package. For details on **R** we refer to the manual by the R Development Core Team (2005). Since the implementation of mtEGO is not published, the code is given in Appendix A.6. The function `mtEGO()` performs Phase 3.A and 3.B of the algorithm and gives the set of candidates. Furthermore, the function already gives the values of the stopping criterion. The functions `elimcuropt()`, `grouping()` and `updates()` implement Phase 3.C of the algorithm. The Kriging model is fitted with the procedure `proc mixed` of the SAS software, Version 9.1. We preferred using the SAS software to the implementations of the Kriging model in the software **R**, because of the availability and simple usage of anisotropic correlation functions. An exemplary code for the fit of the Kriging model can also be found in the appendix. We also give the function `mtEGOimp()` for the improved version of mtEGO, but only the two-sided case is implemented. Also other functions in **R** are given that are only subroutines needed for the main functions mentioned above.

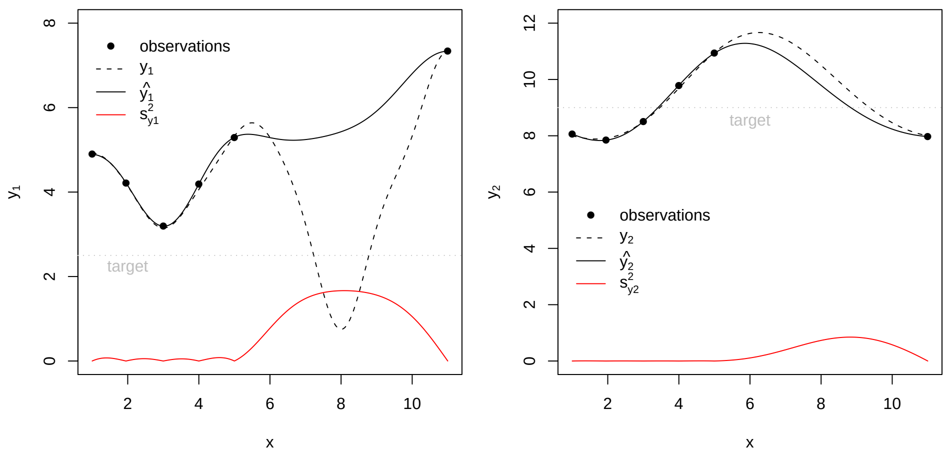


Figure 3.11: Plots of the true underlying relationship and the initial design and fitted model

## 3.7 Step-by-Step Example

In the following we present a simple but illustrative optimization example using mtEGO. The example has only one influencing parameter and two objectives. Figure 3.11 shows the true relationship for the two objectives $y_1$ and $y_2$ (dashed lines), the initial design points (black dots) and the fitted Kriging models $\hat{y_1}$ and $\hat{y_2}$ (solid black line). For a better understanding how the heuristic works, we chose the initial points unbalanced, although we strongly recommend to cover the parameter space in a space-filling way in real applications. Let the optimization target value for $y_1$ be 2.5 and 9 for $y_2$. Two-sided symmetric Harrington desirability functions with the following specification limits are used during the optimization (cf. Section 2.6 and Equation 2.35 for the exact formulae of $d(y_1)$ and $d(y_2)$):

| | target value | $LSL$ | $USL$ | $\nu$ |
|---|---|---|---|---|
| $y_1$ | 2.5 | 1 | 4 | 2 |
| $y_2$ | 9 | 7 | 11 | 2 |

The desirability index is the unweighted geometric mean of the single desirabilities. Figure 3.12 shows the resulting desirabilities for the two objectives and its desirability index of the true underlying relationship. The desirability of objective $y_1$ shows one local and two global optima. The desirability of objective $y_2$ has two global optima. The joint desirability index finally has one local optimum at 3.15, one local optimum at 7.3 and a global optimum that is not much larger than the first local optimum at 8.8.

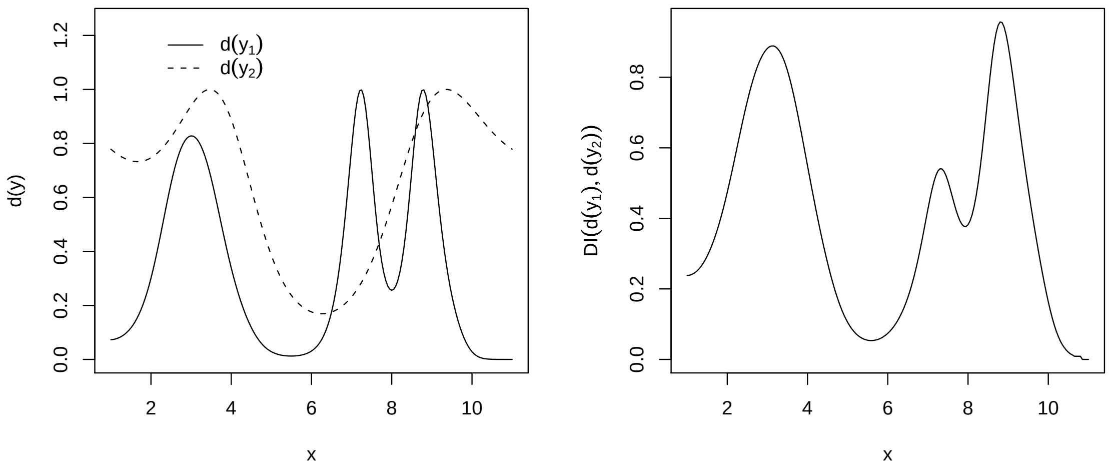


Figure 3.12: Plots of the desirabilities and the desirability index

The initially fitted model in Figure 3.11 indicates that the large local optimum at 3.15 is already found but the other two optima lie in the uncertain area of the parameter space and should be found by mtEGO.

We choose the vector $\alpha \in \{\alpha_1 = 0.005, \alpha_2 = 0.01, \alpha_3 = 0.05\}$ and start constructing the virtual observations and their corresponding desirabilities and indices. Figure 3.13 shows the pure predictions $\hat{y}_1$ and $\hat{y}_2$ (black lines) and the corresponding virtual observations $\hat{y} + t_{1-0.005/2,df} \cdot s^2$ (dashed blue), $\hat{y} - t_{1-0.005/2,df} \cdot s^2$ (dotted blue),
$\hat{y} + t_{1-0.01/2,df} \cdot s^2$ (dashed green), $\hat{y} - t_{1-0.01/2,df} \cdot s^2$ (dotted green),
$\hat{y} + t_{1-0.05/2,df} \cdot s^2$ (dashed red), $\hat{y} - t_{1-0.05/2,df} \cdot s^2$ (dotted red).

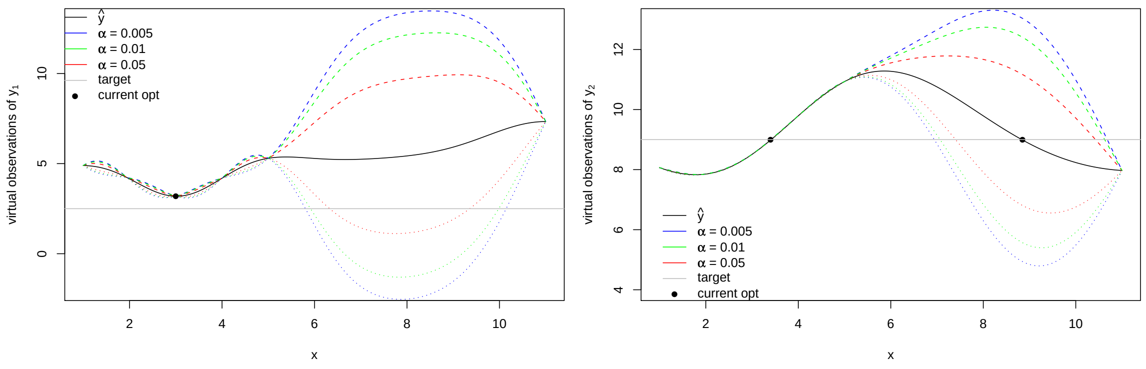


Figure 3.13: Plots of the virtual observations for $y_1$ and $y_2$

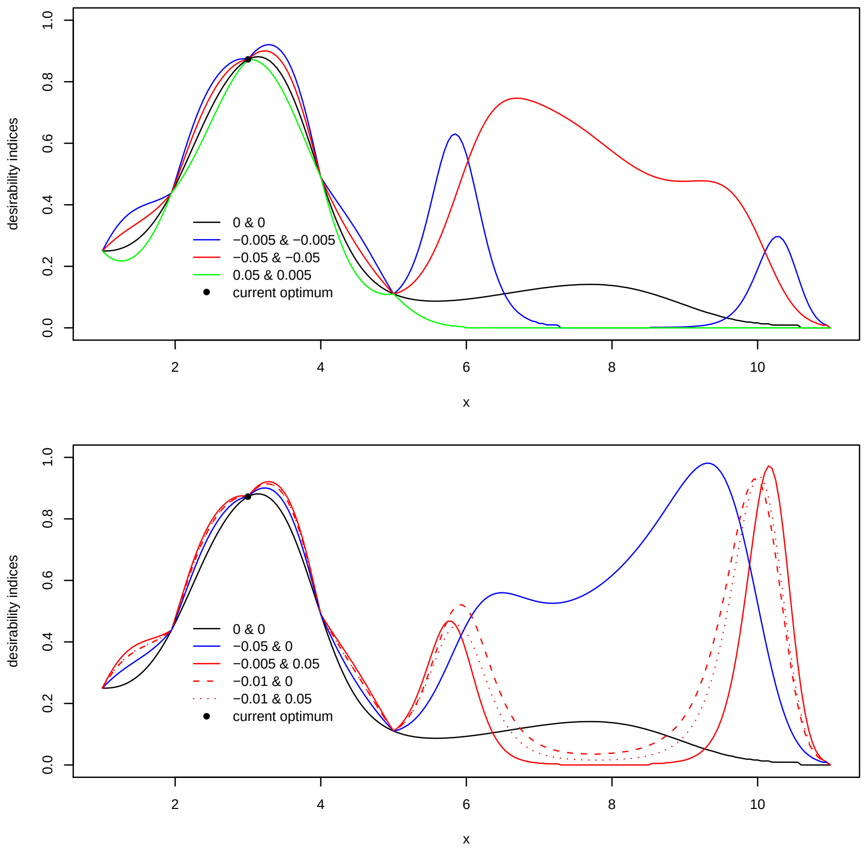


Figure 3.14: Desirability indices for some exemplary pairs of virtual observations with the annotated $\alpha$ level

The curves of the six vectors of virtual observations approach or even meet the target in the interval $[5, 10]$, indicating that there is a possibility for an optimum within the corresponding $(1 - \alpha)\%$ confidence level.

Let us now transform the virtual observations to desirabilities and combine them into desirability indices. Since there are seven vectors for each of the objectives $y_1$ and $y_2$ the complete cross-combination results in 49 different desirability indices (according to Step 3.A.4 in Section 3.2). All indices are similar in the interval $[1, 5]$, where the prediction error is small and the currently observed optimum is found. Most of the indices look like the upper part of Figure 3.14 and have one large peak close to the already known local optimum and only smaller peaks in the uncertain area $[5, 10]$. Only four indices are special and therefore illustrated in the lower part of Figure 3.14. For the annotated combinations of $\alpha$ level virtual observations, the desirability indices get new peaks in the interval $[5, 10]$ that are higher than the peak of the local optimum

| # indices suggested | suggested candidate | Clustering Group | Representative = updating point |
|---|---|---|---|
| 7 | 3.15 | currently predicted optimum | 3.15 |
| 2 | 3.05 | 1 | |
| 11 | 3.25 | 1 | 3.25 |
| 6 | 3.30 | 1 | |
| 1 | 9.30 | 2 | 9.30 |
| 1 | 9.95 | 3 | |
| 1 | 10.00 | 3 | 10.00 |
| 1 | 10.15 | 3 | |
| 19 | NA | no point is better than current optimum | – |

Table 3.1: Clustering of the points with maximum improvement

around $x = 3$. When calculating the improvements for all 49 indices, the indices with the largest peak around the local optimum (displayed in the upper plot) will become maximum around 3 and hence also suggest updating points around 3, which is the region of the local optimum. But for the four other indices the improvements each have their maxima in different points and yield in different updating points as shown in Table 3.1. Altogether, we have a set of eight suggested candidates which partially lie close to each other. After excluding the currently predicted optimum, which is $x = 3.15$, from the set of candidates according to Step 3.C.1 of mtEGO, we use the complete linkage clustering method to reduce the remaining candidates to really interesting updating points. We form three groups (cf. Table 3.1) and determine the representatives for each group. Finally, we have four updating points that are added to the initial design.

After updating the fitted model the whole heuristic starts from afresh. In the next step, the suggested candidates are {9.00, 9.05, 6.00, 6.20, 6.05, 8.80, 8.90, 8.95}. After clustering, three updating points at $\{6.05, 8.90, 9.05\}$ are added to the design. The design has 13 points now. In the third iteration of the heuristic, only $x = 8.8$ is suggested as a candidate, which is the true global optimum. After refining the model with $x = 8.8$ all improvements are 0, also for all virtual observations in the fourth iteration. The global optimum has been found correctly within four steps and altogether 14 points.

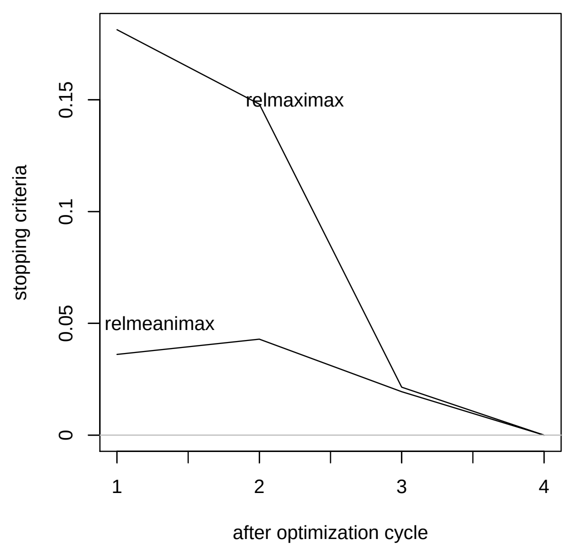


Figure 3.15: Progress of the stopping criteria during the optimization procedure

Figure 3.15 additionally shows the progress of the stopping criteria for this example. It decreases monotonically reaching 0 after mtEGO has been run the fourth time. Figure 3.16 shows the progress of the surrogate model during the updating process. It can be seen how well the procedure approaches the true relationships with quite few points.

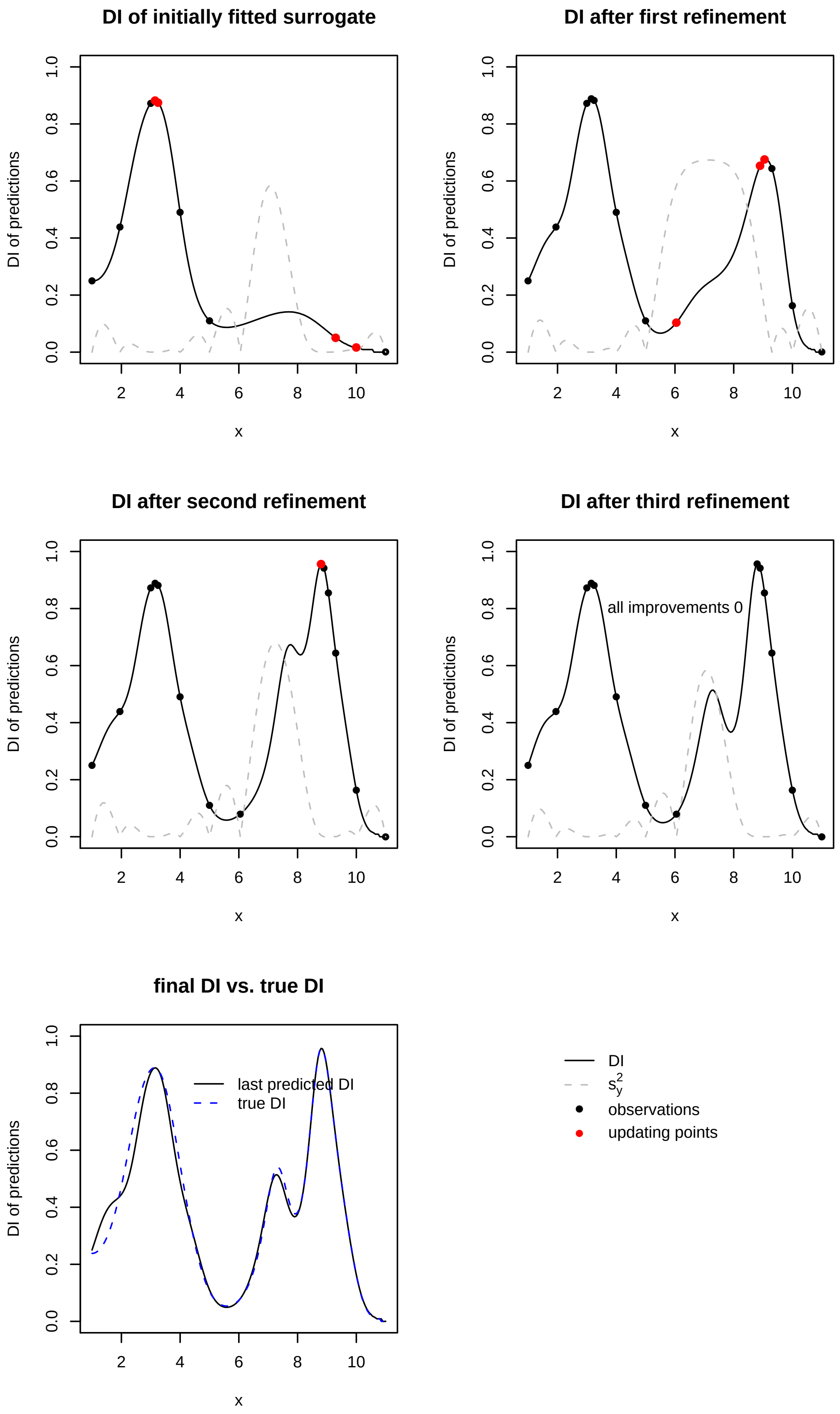


Figure 3.16: The exemplary progress of the surrogate models during the optimization procedure

# 4 Simulation study

In this chapter we present a simulation study on the performance of the mtEGO approach. The algorithm is evaluated for a set of seven different test problems. The main question of the study is whether the global optima are found. Additionally, we are interested in how many steps and how many points are needed to find the optima. Another question of interest is how different choices of $\alpha$ levels influence the optimization process. As a conclusion, a guideline on a reasonable parameterization of the confidence levels is given. The results of the study are then compared to two simple brute force methods. Further, some successful optimizations of advanced problems with different target structures are shown exemplarily for one selected parameterization. Finally, limitations of the mtEGO strategy are discussed briefly.

## 4.1 Introduction of the test problems

Special test suites for the evaluation of multivariate target-value optimization problems are not available in the literature. The test functions for this simulation study are mainly those Knowles (2005) uses to evaluate his ParEGO approach. They are originally coming from Deb et al. (2001), Van Veldhuizen and Lamont (1999), and Okabe et al. (2004) and have been introduced for testing evolutionary algorithms for the whole spectrum of complexity of multivariate minimization problems. Among the test problems some have a simple diagonal pareto front, one has a spiral shaped front or similar complex structures. However, when a special target value is defined for those test suites instead of using them as minimization problems, the complexity of the structure of the global optimum is reduced to quite simple structures, e.g. exactly one point in the parameter space. Therefore, our simulation study does not cover the whole spectrum of complexity of applications. To give a really representative study completely new test suites would have to be developed.

The test problems of this simulation study are introduced in the following. Note, that we use two-sided Harrington desirability functions and the geometric mean as desirability index for all test problems. Section 4.4 presents the applicability of mtEGO using other desirability functions and the weighted geometric mean as desirability index.

**MOSI** is a new creation in the spirit of Sasena et al. (2000). The problem consists of two objectives and only one influencing design variable. The functions of the objectives are

$$\begin{aligned} y_1 &= -\sin(x) - \exp(x/100) + 10 \\ y_2 &= \sin(x) + 2 \cdot \exp(x/10) \end{aligned} \tag{4.1}$$

where $x \in [0, 10]$. The target value optimization problem is defined by Harrington two-sided desirability functions specified with the parameters

| | target value | $LSL$ | $USL$ | $\nu$ |
|---|---|---|---|---|
| $y_1$ | 8.5 | 6.5 | 10.5 | 1 |
| $y_2$ | 4 | 6 | 2 | 1 |

.

The joint global optimum of the two objectives can be found at $x = 6.47$. Figure 4.1 illustrates the objectives and the joint desirability index.

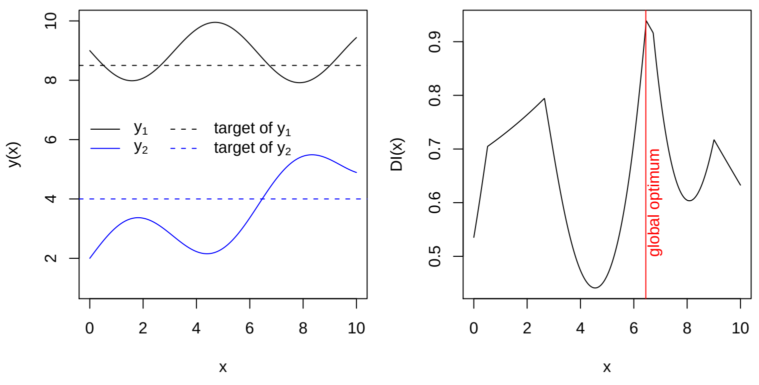


Figure 4.1: Plot of the objectives and desirability index of test problem MOSI

**VLMOP2** has two influencing parameters $x_1, x_2$ and two objectives $y_1, y_2$. It was introduced by Van Veldhuizen and Lamont (1999). The objectives are defined as

$$\begin{aligned} y_1 &= 1 - \exp\left(-\sum_{i=1,2} (x_i - 1/\sqrt{2})^2\right) \\ y_2 &= 1 - \exp\left(-\sum_{i=1,2} (x_i + 1/\sqrt{2})^2\right) \end{aligned} \tag{4.2}$$

for $x_1, x_2 \in [-2, 2]$. The target value problem is specified by two-sided Harrington desirability functions with the parameters

| | target value | $LSL$ | $USL$ | $\nu$ |
|---|---|---|---|---|
| $y_1$ | 0.5 | 0.3 | 0.7 | 2 |
| $y_2$ | 0.5 | 0.3 | 0.7 | 2 |

.

Figure 4.2 shows the contours of the objectives and the desirability index. The global optimum lies in the coordinates $(0, 0)$ surrounded by large areas with desirability index with value 0. mtEGO should find the optimum straight forward, without searching too much in those areas with value 0.

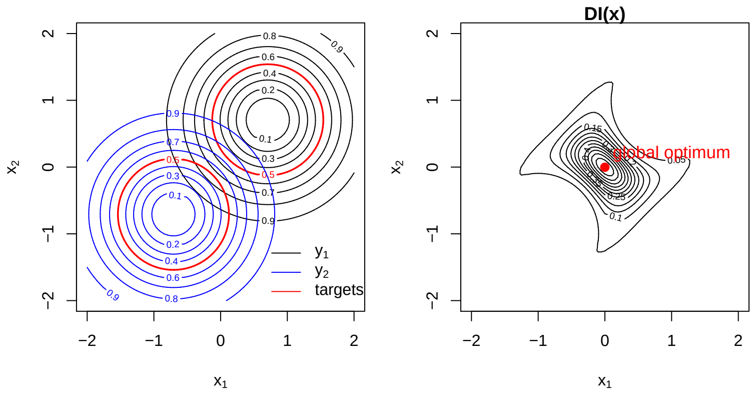


Figure 4.2: Contours of the objectives (left) and desirability index (right) of test problem VLMOP2

**VLMOP3** is a three dimensional test problem introduced by Van Veldhuizen and Lamont (1999). The objectives are defined as

$$\begin{aligned} y_1 &= 0.5(x_1^2 + x_2^2) + \sin(x_1^2 + x_2^2) \\ y_2 &= \tfrac{(3x_1 - 2x_2 + 4)^2}{8} + \tfrac{(x_1 - x_2 + 1)^2}{27} + 15 \\ y_3 &= \tfrac{1}{x_1^2 + x_2^2 + 1} - 1.1\exp(-x_1^2 - x_2^2) \end{aligned} \tag{4.3}$$

for $x_1, x_2 \in [-3, 3]$. The target value problem is specified by two-sided Harrington desirability functions with the parameters

| | target value | $LSL$ | $USL$ | $\nu$ |
|---|---|---|---|---|
| $y_1$ | 4 | 2 | 6 | 2 |
| $y_2$ | 30 | 25 | 35 | 2 |
| $y_3$ | 0.15 | 0.1 | 0.2 | 2 |

.

Figure 4.3 shows the contours of the three objectives and their joint desirability index. The contour of the desirability index shows two global optima in the coordinates $(0.6, -2.5)$ and $(2.5, 0.4)$, one local optimum in the upper left corner and a large region with desirability value 0. mtEGO should find both optima and also give attention to the area around the local optimum.

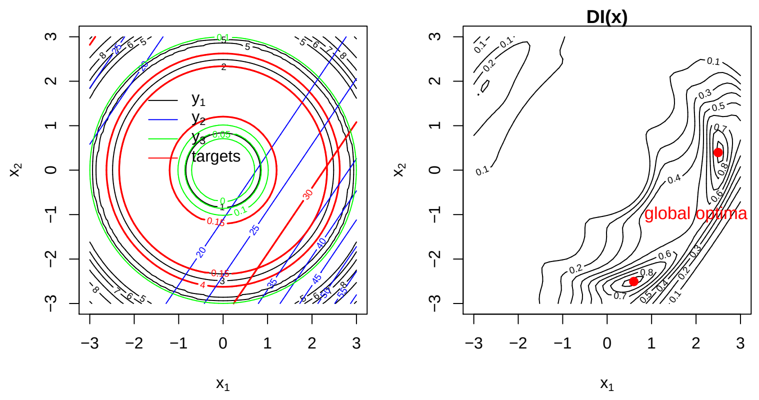


Figure 4.3: Contours of the objectives (left) and desirability index (right) of test problem VLMOP3

**VLMOP3Bound** is a shifted version of VLMOP3. The objectives and desirability functions are defined as before. Now, the parameter space is the rectangular with $x_1 \in [-3, 2.5]$ and $x_2 \in [-2, 3.5]$. We shifted the parameter space such that for this test problem VLMOP3Bound only one global optimum is available in $(2.5, 0.4)$, which lies exactly on the right boundary of the parameter space. Testing VLMOP3Bound is important to see how the heuristic behaves if an optimum lies exactly on the boundary. It is also possible to compare how the efficiency of the heuristic differs between the case when an optimum lies inside the parameter space as for VLMOP3 versus lying on the boundary.

**DTLZ1** is the first of many test problems published by Deb et al. (2001). All test problems introduced there are designed for any $n$ design variables and $M$ objectives.

The objectives are defined as

$$
\begin{aligned}
&y_1 = \tfrac{1}{2}x_1x_2...x_{M-1}(1+g)\\
&y_2 = \tfrac{1}{2}x_1x_2...(1-x_{M-1})(1+g)\\
&...\\
&y_{M-1} = \tfrac{1}{2}x_1(1-x_2)(1+g)\\
&y_M = \tfrac{1}{2}(1-x_1)(1+g)\\
&\text{where} \quad g = 100\left[n + \sum_{i=2,...,n}(x_i-0.5)^2 - \cos(20\pi(x_i-0.5))\right]\\
&\text{and } x_i \in [0,1], i = 1,...,n.
\end{aligned}
\tag{4.4}
$$

For better illustration, we use the DTLZ1 test function with only two design variables and two objectives. The two-sided Harrington desirability functions of the two objectives are specified with the parameters

| | target value | $LSL$ | $USL$ | $\nu$ |
|---|---|---|---|---|
| $y_1$ | 10 | 6 | 14 | 2 |
| $y_2$ | 2 | 0 | 4 | 2 |

.

Figure 4.4 shows the contours of the objectives and the desirability index. The problem is symmetric and has two global optima in the coordinates $(0.8, 0.1)$ and $(0.8, 0.9)$. Mainly the right half of the parameter space is of interest, the left half only contains undesired results.

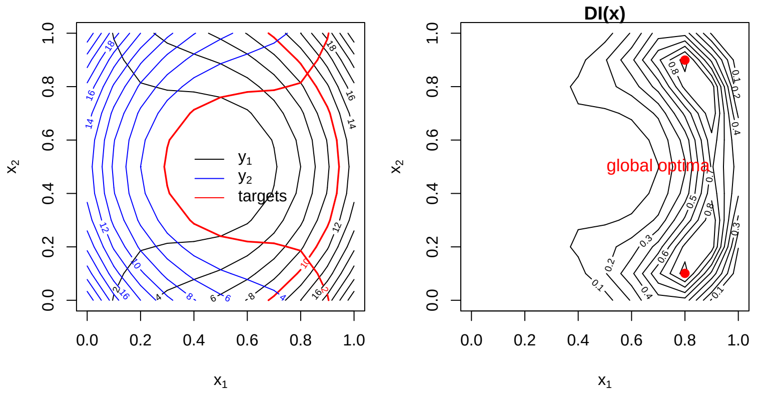


Figure 4.4: Contours of the objectives (left) and desirability index (right) of test problem DTLZ1

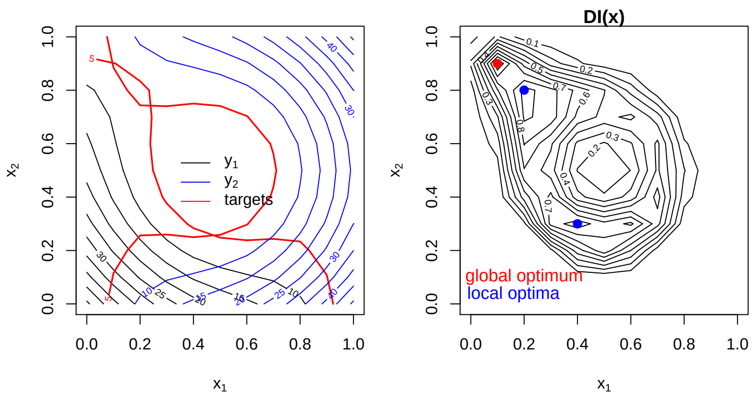


Figure 4.5: Contours of the objectives (left) and desirability index (right) of test problem DTLZ2

**DTLZ2** was published by Deb et al. (2001) designed for any $n$ design variables and $M$ objectives like DTLZ1. The objectives are defined as

$$\begin{aligned}
&y_1 = (1+g)\cos(x_1\pi/2)\cos(x_2\pi/2)\ldots\cos(x_{M-1}\pi/2)\\
&y_2 = (1+g)\cos(x_1\pi/2)\cos(x_2\pi/2)\ldots\sin(x_{M-1}\pi/2)\\
&y_3 = (1+g)\cos(x_1\pi/2)\cos(x_2\pi/2)\ldots\sin(x_{M-2}\pi/2)\\
&\ldots\\
&y_{M-1} = (1+g)\cos(x_1\pi/2)\sin(x_2\pi/2)\\
&y_M = (1+g)\sin(x_1\pi/2)\\
&\text{where}\ \ g = 100\left[n + \sum_{i=2,\ldots,n}(x_i-0.5)^2 - \cos(20\pi(x_i-0.5))\right]\\
&\text{and } x_i \in [0,1], i = 1,\ldots,n.
\end{aligned} \tag{4.5}$$

In our study we use the DTLZ2 test function with only two design variables and two objectives. The two-sided Harrington desirability functions are specified with the parameters

| | target value | $LSL$ | $USL$ | $\nu$ |
|---|---|---|---|---|
| $y_1$ | 5 | 2 | 8 | 2 |
| $y_2$ | 5 | 2 | 8 | 2 |

The global optimum lies in the coordinates $(0.1, 0.9)$, hence very near to the left upper corner of the parameter space (cf. Figure 4.5). The contour of the desirability index also shows two large local optima and one of it lies very close to the global optimum. With this test problem we aim to evaluate which confidence levels are able to find a global and a local optimum and differentiate between them, when they are similar and close to each other.

**KNO** was introduced by Knowles (2005). The problem contains two objectives with two influencing parameters and the formal definition is

$$
\begin{aligned}
&y_1 = 20 - r \cdot \cos(\phi)\\
&y_2 = 20 - r \cdot \sin(\phi)\\
&\text{with } r = 9 - [3\sin(\tfrac{5}{2(x_1+x_2)^2}) + 3\sin(4(x_1+x_2)) + 5\sin(2(x_1+x_2)+2)]\\
&\text{and } \phi = \tfrac{\pi}{12(x_1-x_2+3)}
\end{aligned} \tag{4.6}
$$

for $x_1, x_2 \in [0, 3]$. The target value problem is specified by two-sided Harrington desirability functions with the parameters

| | target value | $LSL$ | $USL$ | $\nu$ |
|---|---|---|---|---|
| $y_1$ | 12 | 7 | 17 | 2 |
| $y_2$ | 19 | 16 | 22 | 2 |

.

The contour of the desirability index (Figure 4.6) shows a very regular structure. There exist two global optima in the coordinates $(1.4, 2.4)$ and $(0.4, 1.4)$ that lie on the top of two very flat long hills. It is a special challenge for optimization algorithms to find a global optimum in a flat area efficiently.

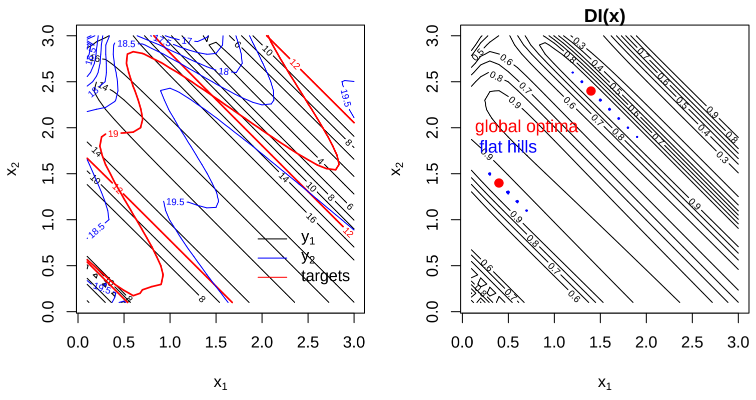


Figure 4.6: Contours of the objectives (left) and desirability index (right) of test problem KNO

**MEINE** is a newly constructed test problem with an asymmetric and skew contour of the desirability index and two almost equally good optima. The problem has three objectives and two influencing parameters. The objectives are defined as

$$
\begin{aligned}
& y_1 = x_1^3 + 0.5 * x_2^4 - 0.1 * x_2^6 \\
& y_2 = 20 - r * sin(\pi/(12 * (x_1 - x_2 + 3))) \\
& \text{with } \; r = 9 - (3 * sin(5/(2 * (x_1 + x_2)^2)) + \\
& \qquad\qquad + 3 * sin(4 * (x_1 + x_2)) + 5 * sin(2 * (x_1 + x_2) + 2)) \\
& y_3 = x_1 - x_2
\end{aligned}
\tag{4.7}
$$

where $x_1, x_2 \in [0, 3]$. To get a skew contour and almost equally sized optima the two-sided Harrington desirability functions are specified with the parameters

| | target value | $LSL$ | $USL$ | $\nu$ |
|---|---|---|---|---|
| $y_1$ | 2 | -3 | 5 | 2 |
| $y_2$ | 19 | 16 | 22 | 2 |
| $y_3$ | -0.5 | -1 | 0 | 2 |

.

The contour of the objectives and the desirability index can be found in Figure 4.7. The two optima lie in the coordinates $(0.6, 1.1)$ and $(2.2, 2.65)$, where the second point is the slightly better one.

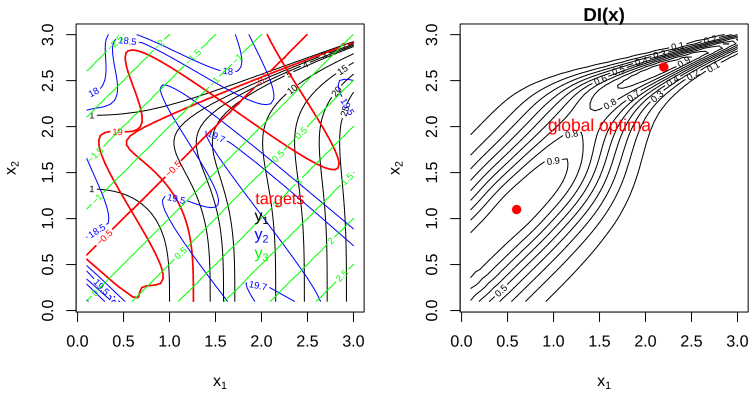


Figure 4.7: Contours of the objectives (left) and desirability index (right) of test problem MEINE

## 4.2 Simulation results

In the following, mtEGO is tested for all test problems that have been introduced in the previous section. It is parameterized with different combinations and numbers of $(1-\alpha)\%$ confidence intervals. The aim of the study is on the one hand to show that the algorithm is able to find global optima in differently complex functional relationships. On the other hand, guidelines for the parameterizations of the $\alpha$ levels are to be developed. The main question thereby is which parameterization is the most reasonable in which situation. Or does any parameterization exist that yields the overall best results. Table 4.1 shows eight different parameterizations of $\alpha$ levels that are used during the study.

The chosen parameterizations of $\alpha$ levels vary from three to six different levels that are used simultaneously in the algorithm. We did not try more levels for reasons of computation time. However, the results show good performance already for this small number of levels. Some of the parameterizations concentrate on very small $\alpha$ levels (i.e wide confidence intervals), some on large $\alpha's$ (i.e. narrow confidence intervals). Most of the chosen parameterizations combine wide, medium-sized and narrow intervals.

First of all, we optimize the very simple test problem MOSI with all eight different parameterizations shown in Table 4.1. The results show only slight differences at all. When starting with an initial 5 point design, mtEGO succeeds to find the global optimum by adding 9 to 14 updating points in 4 optimization cycles with all eight parameterizations. After the first cycle of mtEGO two to seven updating points and in the second cycle 3 to 4 points are added. The refined surrogate model after the second cycle is already that good, that for all parameterizations the global optimum is nearly found. When the third cycle of mtEGO is conducted, the stopping criterion saturates close to 0 and the heuristic stops. For all parameterizations the currently best observed point then is the global optimum. The choice of the $\alpha's$ does not effect the progress of the optimization very much for this simple test problem. Details on the progress of the optimization with the different parameterizations are given in Appendix A.5.

For all other test problems, the choice of the $\alpha$ levels does have an effect on the efficiency and even the success of the algorithm. Table 4.2 summarizes an evaluation of the results. The table gives the following evaluation criteria:

- The number of cycles that mtEGO needs until the stopping criterion is satisfied. A number of five cycles means that updating points were added to the model

| no. | $\alpha$ levels | properties |
|---|---|---|
| 1 | 0.001, 0.01, 0.05 | only wide confidence boundaries |
| 2 | 0.0001, 0.001, 0.05, 0.85 | many very wide and one narrow confidence boundaries |
| 3 | 0.05, 0.1, 0.25, 0.5, 0.75 | medium-sized confidence levels |
| 4 | 0.001, 0.01, 0.1, 0.25, 0.5 | all sizes of confidence boundaries regularly distributed |
| 5 | 0.001, 0.1, 0.5, 0.85 | very wide, medium and very narrow confidence boundaries |
| 6 | 0.001, 0.01, 0.1, 0.25, 0.5, 0.85 | levels distributed like in no.5, but more levels |
| 7 | 0.01, 0.1, 0.5, 0.9 | similar to the levels in no. 5, but smallest and largest level enlarged |
| 8 | 0.1, 0.5, 0.7, 0.8, 0.9 | many narrow confidence boundaries, no wide boundary |

Table 4.1: Different combination of $\alpha$ levels for the $(1-\alpha)$ confidence intervals that are used in the simulation study

| | parameterization → | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|---|
| VLMOP2 | # cycles | 5 | 5 | 5 | 5 | 5 | 5 | 6 | **5** |
| | # design points | 25 | 27 | 27 | 27 | 31 | 32 | 30 | **25** |
| | optimum/model | o/o | o/o | */+ | */+ | */+ | */+ | */+ | ***/+** |
| | progress candidates | + | o | ++ | + | + | + | ++ | **++** |
| | progress updatings | o | o | ++ | o | + | - | + | **++** |
| | clumping | - | - | o | - | o | - | + | **+** |
| VLMOP3 | # cycles | 5 | 6 | 4 | 6 | 4 | 5 | **5** | 5 |
| | # design points | 25 | 31 | 22 | 36 | 27 | 34 | **28** | 22 |
| | optimum/model | +o/+ | *o/+ | ++/o | *o/+ | ++/+ | *o/+ | ***+/+** | *+/+ |
| | progress candidates | o | o | ++ | + | o | + | **++** | + |
| | progress updatings | o | o | + | + | + | + | **++** | ++ |
| | clumping | - | - | o | o | + | + | **+** | + |
| VLMOP3Bound | # cycles | 8 | 7 | 6 | 7 | **5** | 7 | 7 | 5 |
| | # design points | 35 | 36 | 32 | 36 | **28** | 40 | 31 | 25 |
| | optimum/model | */+ | */o | o/o | o/+ | ***/+** | +/+ | +/+ | +/+ |
| | progress candidates | - | o | o | + | **+** | o | + | ++ |
| | progress updatings | - | o | - | o | **+** | - | o | ++ |
| | clumping | - | - | + | o | **+** | + | + | o |
| DTLZ1 | # cycles | 7 | 5 | 4 | 7 | 6 | 5 | **4** | 4 |
| | # design points | 30 | 28 | 22 | 41 | 31 | 29 | **21** | 24 |
| | optimum/model | *o/+ | *+/+ | *+/+ | **/+ | **/+ | *+/+ | ****/+** | *+/o |
| | progress candidates | - | o | ++ | + | + | + | **++** | + |
| | progress updatings | - | o | + | - | + | o | **+** | + |
| | clumping | o | + | o | o | + | o | **+** | + |
| DTLZ2 | # cycles | 4 | 5 | 4 | 3 | 5 | 6 | 5 | **4** |
| | # design points | 16 | 26 | 21 | 18 | 35 | 28 | 25 | **23** |
| | optimum/model | o/o | o/+ | o/+ | o/o | */+ | */+ | */+ | ***/+** |
| | progress candidates | - | o | + | - | + | + | + | **++** |
| | progress updatings | - | o | + | - | + | o | + | **++** |
| | clumping | - | o | + | + | + | o | o | **+** |
| KNO | # cycles | 4 | 6 | 4 | 5 | 5 | 4 | 6 | **5** |
| | # design points | 8 | 30 | 24 | 28 | 26 | 27 | 39 | **29** |
| | optimum/model | - -/o | *+/o | *o/+ | oo/+ | oo/+ | o-/o | o*/+ | ***+/+** |
| | progress candidates | - | o | o | o | o | o | + | **++** |
| | progress updatings | - | + | o | o | + | - | + | **+** |
| | clumping | o | o | - | o | o | - | + | **+** |
| MEINE | # cycles | 4 | 4 | 5 | 5 | **4** | 4 | 4 | 5 |
| | # design points | 20 | 26 | 28 | 30 | **18** | 23 | 27 | 23 |
| | optimum/model | +o/+ | *+/+ | *+/+ | ++/+ | ***+/+** | *+/+ | *+/+ | *o/+ |
| | progress candidates | o | + | ++ | ++ | **+** | o | + | + |
| | progress updatings | o | + | + | o | **++** | + | o | ++ |
| | clumping | + | + | o | + | **+** | + | o | + |

Table 4.2: Evaluation of the simulation results for the different used confidence levels ($\alpha$ levels belonging to the numbers in the headline are listed in Table 4.1)

for four times and when running Step 2 to 4 of mtEGO for the fifth time the algorithm stops without adding new updating points.

- The amount of design points that have been examined during the optimization procedure (i.e design plus all updating points). In real applications this is the number of experiments that are to be run.
- The evaluation of how precise the global optima are found and how good the overall functional relationship is represented by the final model. If the test problem has only one global optimum the first sign belongs to the optimum and the second sign to the model. A '-' indicates that the optimum is not found at all and respectively the model is far from the true relationship. '+' means the optimum is quasi found, i.e. the currently found optimum is a direct neighbor grid point of the true global optimum. Respectively, the contour plot of the model resembles the true relationship well if it is marked with '+'. 'o' is the neutral case, the currently found optimum is close to the true optimum, but not found correctly and the model is acceptable but not really good. If the global optimum is found precisely this is marked with '$*$'.
  In the case that the test problem has two global optima the first two signs belong to the evaluation of the found optima and the third sign evaluates the finally fitted model. Note, that it is possible that the algorithm finds the optimum, but the overall model fit is actually not good, e.g. when it is only precise around the global optimum.
- The progress of the distribution of the set of candidates ($C$ in Step 3.B.2 of mtEGO, see Section 3.2) and the progress of the distribution of the updating points (the points $x_{cent}$ and $x^*$ from Step 3.C.3). Both progresses are rated with '-' for bad, 'o' for acceptable, '+' for good and '++' for excellent. It is evaluated exemplarily with graphics as shown in Figure 4.8 for the test problem VLMOP3. It can be seen how the set of candidates (marked with black squares) evolves over the optimization steps and approaches the global optima more and more. This progress would be evaluated with '++'. A good progress of the candidates is important to ensure that mtEGO does not end up in a stochastic search or gets inefficient. The heuristic should concentrate more and more on the area of the global optimum, but not forget about local optima, and not examine the undesired areas too much, just as the progress in the figure does. The same

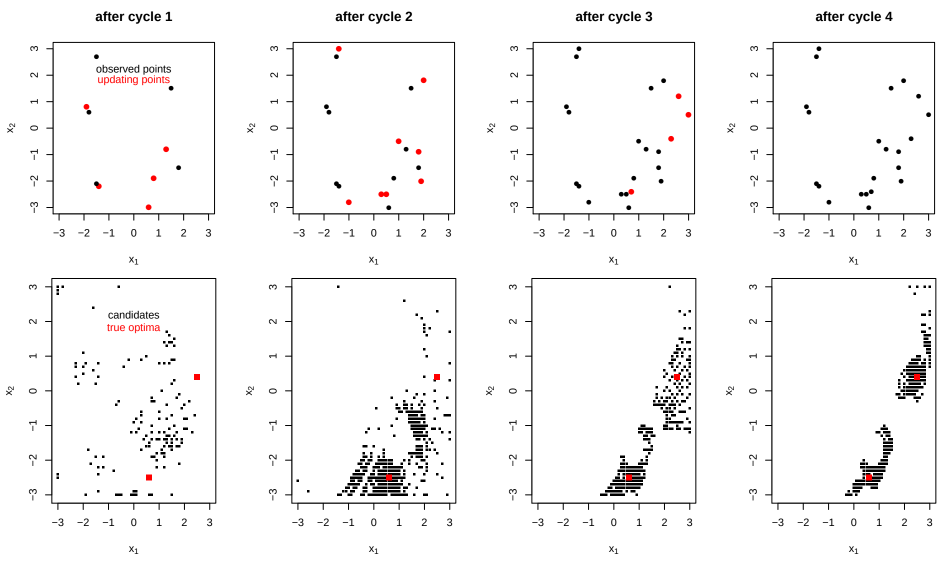


Figure 4.8: Exemplary figure to evaluate the progress of candidates and updating points for test problem VLMOP3

applies for the progress of the updating points, which are marked with red dots in the figure.

- The intensity of clumping among the updating points. It is also evaluated with plots like Figure 4.8. In some cases it happens that the algorithm suggests updating points that lie very close to the already measured points. This phenomenon is described in more detail in Section 3.5. In the area around the global optimum this is of course desirable. But if all updating points just lie next to already measured points this is a waste of time and experimental runs. An optimization process without clumping is more efficient and hence better than one where all updating points build clusters around the initial points. No clumping is assigned with '+', few clumping with 'o' and heavy clumping with '-'.

The best overall parameterization for every test problem is printed in blue, i.e. the one that has the most '+' or '∗' assigned and then needs the fewest cycles and points. In the following, we only discuss the summary of the results of the simulation study from Table 4.2. Details on the single progresses of the optimizations and all figures of the progress can be found in Appendix A.5.

Regarding the question if mtEGO is able to find the global optimum in differently complex situations Table 4.2 shows, that the heuristic works fine. For all problems the global optima could be found within limited time and a certain parameterization. mtEGO is able to deal with two objectives (e.g. VLMOP2, DTLZ1, DTLZ2) and three objectives (e.g. VLMOP3, MEINE), as well as one global optimum (e.g. VLMOP2, DTLZ2) or two global optima (e.g. DTLZ1, VLMOP3, MEINE). mtEGO has no problems with symmetric situations (e.g. VLMOP2, DTLZ1, VLMOP3) neither with asymmetric situations, (e.g. DTLZ2, MEINE). Even if the optimum lies on the boundary of the parameter space, the optimum is found. mtEGO then needs additional points and cycles to localize the optimum, see the results for VLMOP3 and VLMOP3Bound. The additional points and cycles can be explained by the fact, that the candidate the most in the middle of the cluster is chosen as updating point (cf. Step 3.C.3 in Section 3.2). The cluster containing the potential optimum on the boundary of the space, hence, must be very small to choose an updating point on the boundary. mtEGO is also able to handle test problem DTLZ2 that has two local optima that are almost as good as the global optimum. It finds the global optimum correctly among the other good regions. In contrast, mtEGO does not differentiate correctly between the almost equally sized optima of test problem MEINE. The progress of the candidates show for all parameterizations that mtEGO gives the same attention to both optima and mainly decides for the slightly worse local optimum in the end. The other situation that seems really difficult for the algorithm is the case of flat hills around the optimum, like for the test problem KNO. If the optimum is not a clear peak but only slightly better than the whole area around, mtEGO needs many updating points, many optimization cycles and a good parameterization to be able to find the global optimum precisely. For most of the parameterizations mtEGO could only approach the global optimum and stopped there, but did not find the precise point. At least the finally fitted model was in most cases acceptable, such that the user could assume from the model that the optimum is a region instead of being a precise point, which is not completely false for this test problem. If the precise global optimum shall be known, additional points and cycles would be needed. In such a situation it may be wise to stop with the approximate region region of the global optimum and then sample directly inside this region with a very local parameterization (i.e. only very narrow confidence intervals) or even another optimization strategy.

Since, mtEGO does find the global optima almost with all parameterizations, we compare the different parameterizations now. The question of interest here is, whether mtEGO works faster or better with any special parameterization for the different situations and if any parameterization can be favored. The most important result of this study is that mtEGO is able to find or at least to approach the global optimum with each parameterization of $\alpha$. None of the parameterizations found a completely false optimum, except for test problem KNO. As already mentioned, KNO is due to the flat hills a very difficult optimization situation. However, the evaluated parameterizations behave different concerning the question how fast and efficient they find the optima.

It is not surprising that a very global parameterization with mainly wide confidence intervals, e.g. number 1,2 or 4, tend to only approach the optima. The progress of the set of candidates and also the progress of the updating points is rather poor for these parameterizations. The updating points often cluster around the initial design points. If the confidence intervals are chosen only very wide, it happens that only less probable candidates for the optimum are considered which are the neighbors of the already measured points. For test problem KNO the global parameterizations number 1 and 2 are the ones that also fail to approach an acceptable model fit, showing the existence of flat hills around the optima. These parameterizations only seem important if the initial model is strongly false (for details on this problem see the explanations in Section 3.5).

Here, the parameterizations number 5,7 and 8 show the best progress of optimization and best results. Number 8 is a very local parameterization with mainly narrow intervals. Such a parameterization can find the optima very precisely, since it allows one to search locally around the currently found optimum. The drawbacks are more optimization cycles and many experiments, if the initial search direction is wrong because of a bad initial model. We can explain the success of parameterization number 8 by the goodness of the initial surrogate models. For our simulation study we used space-filling initial designs with few points (five to seven). The models were not precise, but basically o.k. None of the models modeled the true relationship completely false. Having a good surrogate, it was appropriate to search locally around the current optimum from beginning. The parameterization would not succeed that well if the initial model would be very bad. For the test problem KNO with very flat hills around the optimum, the very local parameterization number 8 is even needed to find the optimum among the many good points.

Parameterization numbers 5 and 7 use four levels regularly arranged between 0.01 and 0.9, i.e. some global and some local components are combined. It is intuitively appealing that a combination of local and global components in the parameterization of the optimization procedure yields good results. The same applies for parameterization number 3. It also uses regularly arranged $\alpha$-levels between medium-size and narrow. In our case study this parameterizations indeed also shows an overall good progress and result, but it is little worse than 5 and 7. We conclude that one really narrow confidence level is needed.

Summarizing all evaluations of the parameterizations, we come to the conclusion that a combination of wide and narrow confidence levels regularly arranged (e.g. number 5 and 7) with an emphasis on narrow confidence levels is most preferable for all situations. For applications that permit larger numbers of levels than four or five levels, our advise is to use $\alpha \in \{0.01, 0.1, 0.5\}$ and then chose the remaining levels locally. For example choose $\alpha \in \{0.01, 0.1, 0.5, 0.9\}$ as a four level parameterization. A five level parameterization could be $\alpha \in \{0.01, 0.1, 0.5, 0.7, 0.9\}$ and a six level parameterization $\alpha = 0.01, 0.1, 0.5, 0.7, 0.8, 0.9$. Only if the initial model fit is bad, e.g. the mean squared cross validation error is very large, we suggest to use smaller levels like 0.001, too.

## 4.3 Comparison of the mtEGO approach with brute force methods

The simulation study presented in the previous section shows that mtEGO is able to find global optima in various situations. Now we compare the results of the mtEGO strategy with results from two brute force methods. It is assumable that the global optima can also be found with simple designs, they only must be large enough. Let mtEGO need a certain number of experiments in total to find the optimum of a given optimization problem. It is examined here how close a one-step design with the same amount of design points approaches the optimum and whether mtEGO dominates the simple strategy or not. The results of mtEGO are therefore compared with results from a large one-step space-filling design and a random design.

First, for each test problem from the previous section we start with the same initial design as in the mtEGO simulation study and add the same number of points as the mtEGO strategy needs randomly chosen from the parameter space. The procedure is repeated 100 times for each test problem and each parameterization. Table 4.3 gives the percentage of how often mtEGO finds an optimum lying closer to the true global optimum than the optimum found by the random search, i.e. the percentage where mtEGO is superior to the random search.

The mean of the percentages given in the table shows that mtEGO is superior to the random search for six of the eight test problems. For the test problems DTLZ1 and DTLZ2 the random search yields better results in approximately 60% of the repetitions. The dominance of the random search for these two problems can easily be explained by the size of the grid representing the parameter space. DTLZ1 and DTLZ2 have a parameter space grid with only 121 points, but anyway 24 or 28 design points are observed. The examined grid portion is approximately 20%. For the other test problems the examined grid portion is always less than 5%. It is clear that the models and results for DTLZ1 and DTLZ2 respectively are, hence, very good even with the random search. Furthermore, if we have a look on the results of mtEGO for DTLZ1 and DTLZ2 in Table 4.2, we can see that the results of mtEGO were poor particularly for those parameterizations where the random search is strongly dominating. Table 4.4 compares the optima achieved by mtEGO and the random design regarding their achieved desirability. It is possible, that an optimum that lies more remote from the true optimum anyway has a better desirability index than the closer optimum. The table again gives the percentage where the results of mtEGO are better then those of the random design, i.e. where mtEGO achieves the higher desirability. The comparison regarding the achieved quality also shows the dominance of the mtEGO approach. The mean percentage where mtEGO is better than the random design even increases compared to the comparison regarding the closeness of the found optima.

The results attained with the mtEGO strategy are now compared with a large equally-sized space-filling design. Here we start with the two design points that have maximum distance in the parameter space and add subsequently the number of updating points that mtEGO needs according to the coffee-house design criterion. Since the updating points are determined according to a fixed criterion, repetitions are not needed here. Table 4.5 displays for which parameterization mtEGO dominates the equally-sized space-filling design. '+' states that mtEGO finds an optimum that lies closer to the true optimum than the optimum found by the coffee-house design, i.e. mtEGO is superior. '-' states that the coffee-house design is superior. 'o' indicates that both strategies find optima that have the same distance to the true optimum or they both find the true optimum.

As one can see in Table 4.5 mtEGO dominates the coffee-house design for the major-

| parameterization | MOSI | VLMOP2 | VLMOP3 | VLMOP3Bound | DTLZ1 | DTLZ2 | KNO | MEINE |
|---|---|---|---|---|---|---|---|---|
| 1 | 65 | 50 | 94 | 92 | 34 | 15 | 41 | 42 |
| 2 | 61 | 11 | 95 | 90 | 36 | 1 | 93 | 60 |
| 3 | 75 | 89 | 92 | 64 | 45 | 28 | 5 | 54 |
| 4 | 67 | 92 | 98 | 57 | 41 | 16 | 26 | 46 |
| 5 | 54 | 78 | 97 | 96 | 53 | 80 | 69 | 81 |
| 6 | 76 | 82 | 94 | 73 | 27 | 65 | 0 | 67 |
| 7 | 59 | 86 | 95 | 75 | 43 | 24 | 93 | 55 |
| 8 | 65 | 88 | 97 | 91 | 63 | 82 | 96 | 56 |
| mean | 65.25 | 72.00 | 95.25 | 79.75 | 42.75 | 38.88 | 52.88 | 57.63 |
| grid size | 1001 | 6561 | 3721 | 3136 | 121 | 121 | 900 | 3481 |
| mean design size | 16 | 28 | 28 | 32 | 28 | 24 | 26 | 24 |
| examined grid-portion in % | 0.016 | 0.004 | 0.008 | 0.010 | 0.231 | 0.198 | 0.029 | 0.007 |

Table 4.3: Percentage where the optimum found with mtEGO is closer to the true optimum than the optimum found with a random design

| parameterization | MOSI | VLMOP2 | VLMOP3 | VLMOP3Bound | DTLZ1 | DTLZ2 | KNO | MEINE |
|---|---|---|---|---|---|---|---|---|
| 1 | 65 | 84 | 97 | 92 | 34 | 25 | 45 | 50 |
| 2 | 61 | 48 | 98 | 90 | 36 | 14 | 95 | 60 |
| 3 | 75 | 89 | 96 | 64 | 45 | 38 | 48 | 54 |
| 4 | 67 | 92 | 99 | 57 | 41 | 26 | 52 | 46 |
| 5 | 54 | 78 | 99 | 96 | 53 | 80 | 49 | 81 |
| 6 | 76 | 82 | 97 | 73 | 27 | 65 | 51 | 67 |
| 7 | 59 | 86 | 99 | 75 | 43 | 31 | 89 | 55 |
| 8 | 65 | 88 | 99 | 91 | 63 | 82 | 99 | 56 |
| mean | 65.25 | 80.88 | 98.00 | 79.75 | 42.75 | 45.13 | 66.00 | 58.63 |
| grid size | 1001 | 6561 | 3721 | 3136 | 121 | 121 | 900 | 3481 |
| mean design size | 16 | 28 | 28 | 32 | 28 | 24 | 26 | 24 |
| examined grid-portion in % | 0.016 | 0.004 | 0.008 | 0.010 | 0.231 | 0.198 | 0.029 | 0.007 |

Table 4.4: Percentage where the achieved desirability is better for mtEGO than for the random design

| parameterization | MOSI | VLMOP2 | VLMOP3 | VLMOP3Bound | DTLZ1 | DTLZ2 | KNO | MEINE |
|---|---|---|---|---|---|---|---|---|
| 1 | + | - | + | + | o | o | + | + |
| 2 | + | - | + | + | o | - | + | + |
| 3 | + | o | + | - | o | o | - | + |
| 4 | + | o | + | - | o | o | - | + |
| 5 | + | + | + | + | o | + | + | + |
| 6 | + | + | + | + | o | + | - | + |
| 7 | + | o | + | o | o | o | + | + |
| 8 | + | o | + | + | o | + | + | + |
| mtEGO better than coffee-house | 8 | 2 | 8 | 5 | 0 | 3 | 5 | 8 |
| mtEGO and coffee-house equal | 0 | 4 | 0 | 1 | 8 | 4 | 0 | 0 |
| coffee-house better than mtEGO | 0 | 2 | 0 | 2 | 0 | 1 | 3 | 0 |
| examined portion of grid in % | 0.016 | 0.004 | 0.008 | 0.010 | 0.231 | 0.198 | 0.029 | 0.007 |

Table 4.5: Comparison of the distance of the optima obtained with mtEGO and the coffee-house design

| parameterization | MOSI | VLMOP2 | VLMOP3 | VLMOP3Bound | DTLZ1 | DTLZ2 | KNO | MEINE |
|---|---|---|---|---|---|---|---|---|
| 1 | + | - | + | + | o | o | - | + |
| 2 | + | - | + | + | o | - | + | + |
| 3 | + | o | + | - | o | o | - | + |
| 4 | + | o | + | - | o | o | - | + |
| 5 | + | + | + | + | o | + | - | + |
| 6 | + | + | + | + | o | + | - | + |
| 7 | + | o | + | o | o | o | + | + |
| 8 | + | o | + | + | o | + | + | + |
| mtEGO better than coffee-house | 8 | 2 | 8 | 5 | 0 | 3 | 3 | 8 |
| mtEGO and coffee-house equal | 0 | 4 | 0 | 1 | 8 | 4 | 0 | 0 |
| coffee-house better than mtEGO | 0 | 2 | 0 | 2 | 0 | 1 | 5 | 0 |
| examined portion of grid in % | 0.016 | 0.004 | 0.008 | 0.010 | 0.231 | 0.198 | 0.029 | 0.007 |

Table 4.6: Comparison of the achieved desirabilities with mtEGO and the coffee-house design

ity of the test problems and parameterizations. The true optimum is rarely found more precise with the coffee-house design than with mtEGO. For the test problems DTLZ1 and DTLZ2 mtEGO and the brute force method have about the same power to find the global optimum. The reason is the same as for the random search. A space-filling grid covering 20% of the parameter space grid yields very good model fits and can thus approach the global optimum very closely. The desirability index of test problem VLMOP2 has a really simple structure. Here, very few design points are sufficient to yield such a good model fit that predicts the global optimum correctly. One can say, mtEGO needs too many updating points for such a simple structured problem. However, the amount of the updating points evolves from the problem of the stopping criterion in a sequential procedure. One always needs an additional step with probably several updating points to note that an optimum has been found, whereas the large one-step space-filling design uses the given number of design points and eventually confirms the predicted optimum with only one point. However, mtEGO can not dominate the coffee-house design for VLMOP2, but it is not dominated either.

Table 4.6 gives an evaluation of mtEGO and the coffee-house design comparing the desirability of the found optima. '+' states that mtEGO finds an optimum with a larger desirability index compared to the desirability index of the optimum found by the coffee-house design. '-' states that the optimum found by the coffee-house design achieves the larger desirability index. 'o' indicates that both strategies find optima with the same desirability index. Beside for the test problem KNO the results do not differ from the comparison regarding the closeness of the optima. For the test problem KNO the coffee-house now dominates mtEGO in opposite of the comparison regarding the closeness of the optima. This test problem is indeed a really difficult problem for mtEGO due to its flat hills around the optimum. In this case the optimization using mtEGO often failed to find the optimum correctly and could only approach it. The results implicate that mtEGO tries to locate the global optimum as close as possible. The large space-filling design yields better allover model fits and apparently finds optima with larger desirability than mtEGO, even though they are located more remote from the true optimum. Summarizing the results of both comparisons for the test problem KNO, we come to the conclusion that mtEGO is not powerful and efficient for test problems with flat hills as KNO has.

For the most complex and irregular structured test problem MEINE, both brute force methods are dominated by mtEGO with respect to both criteria closeness and achieved desirability. It may be expected that mtEGO dominates the brute force methods for large and complex problems. However, due to the requirement to supervise the progress of the optimization using mtEGO we chose the test problems in the simulation study small and relatively simple.

## 4.4 Solving advanced test problems with the new heuristic

In Section 4.2 all considered objectives are transformed using two-sided Harrington desirability functions and the unweighted geometric mean as index. In this section it is shown exemplarily that mtEGO also works for other test problems, e.g. using Derringer-Suich desirabilities, mixed desirabilities (one- and two-sided), the weighted geometric mean as index and large-dimensional problems. Having already studied different parameterizations of $\alpha$ in the previous section, we now use $\alpha \in \{0.01, 0.1, 0.5, 0.9\}$ for these further problems. This was one of the best parameterizations in the case study. Except for the large dimensional example, all following test problems are based on the test problem VLMOP3 (cf. Section 4.1) and have the same parameter space and objectives. For this one test problem completely different optimization targets are defined with the help of various specifications of the desirability functions and indices. The contour of the true desirability index and the global optima are different for each example, which is illustrated in Figure 4.9. These further test problems also demonstrate the flexibility of the new heuristic. The possibility to use any kind of desirability function and index permits to specify any possible target structure for multivariate optimization problems. With the help of the following examples, we show that mtEGO is indeed able to solve them successfully.

### 4.4.1 One-sided multivariate test problem

The test problem VLMOP3 from Section 4.1 is considered here as a simple one-sided test problem. All three objectives are to be minimized. The specified desirability functions are one-sided Harrington functions with the parameters

| objective | $y_1$ | $y_2$ | $y_3$ |
|---|---|---|---|
| $b_0$ | 3 | 8 | 1.5 |
| $b_1$ | -0.8 | -0.3 | -15 |

.

Using the unweighted geometric mean as desirability index, the contour has the form displayed in Figure 4.9a and the joint minimum can be found in the coordinates $(-0.1, 0.1)$. The progress of the optimization procedure using $\alpha \in \{0.01, 0.1, 0.5, 0.9\}$ is summarized in Table 4.7.

In total mtEGO needs 30 experiments to find the optimum. Starting with a 5 point design and a bad initial model fit, the optimum is searched by adding updating points

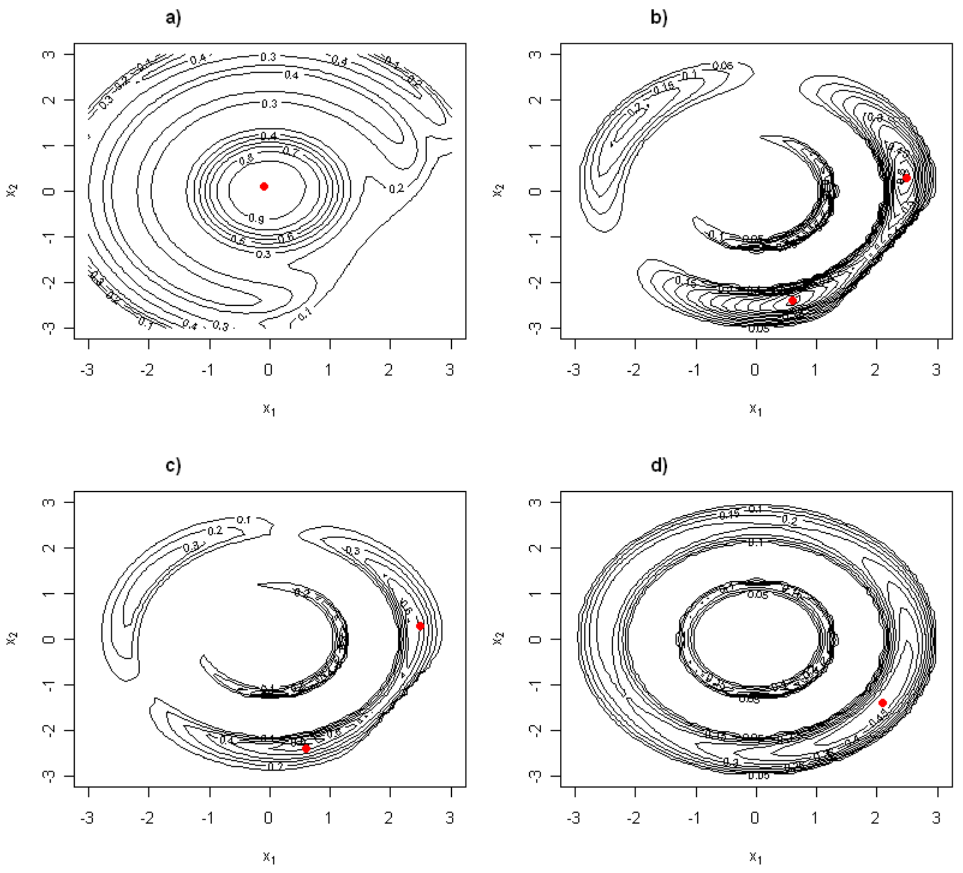


Figure 4.9: Contour plots of the true desirability indices for the advanced test problems: a) one-sided desirabilities, b) Derringer-Suich desirabilities, c) weighted geometric mean as index, d)complex mixed target structure (the red dots mark the global optima)

| cycle | # of design points | currently observed optimum | currently predicted optimum | model fit | $relmax-imax$ | global optimum found? | # of added points | location of updatings |
|---|---|---|---|---|---|---|---|---|
| 1 | 5 | (-1.5,-2.1) | (-0.2,3) | bad | 7.109 | not found | 5 | ok |
| 2 | 10 | (-1.5,-2.1) | (-0.7,2.1) | better | 4.783 | not found | 5 | ok |
| 3 | 15 | (0.5,0.4) | (0.5,0.4) | moderate | 0.553 | approaching opt | 3 | good |
| 4 | 18 | (0.2,0) | (0.2,0) | ok | 0.288 | opt close | 7 | good |
| 5 | 25 | (-0.2,0) | (-0.1,0.1) | good | 0.178 | opt close | 5 | good |
| 6 | 30 | (-0.1,0.1) | (-0.1,0.1) | good | 0.128 | opt found | stop | |

Table 4.7: Progress of the optimization of a one-sided multivariate test problem using the new heuristic

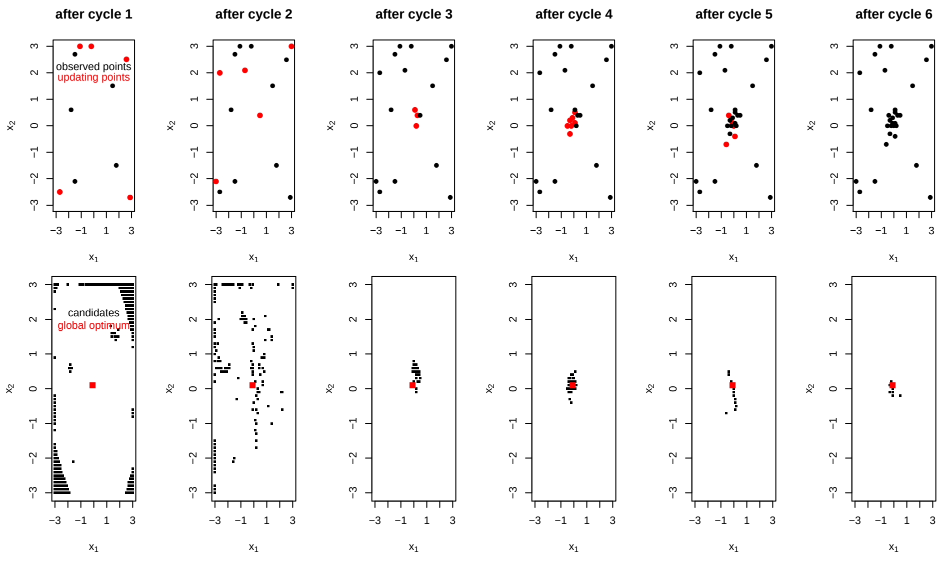


Figure 4.10: Progress of the candidates and updating points for a one-sided multivariate test problem

all over the parameter space in the beginning to improve the model quality.

In the later cycles target-oriented updating points are added. The stopping criterion decreases monotonically in every cycle. The decrease gets already small (0.288) when the optimum is almost found and starts saturating slightly in the following optimization cycle, where the global optimum is met. Hence, mtEGO also does succeed for a one-sided optimization problem. Although, we must admit that 30 experiments is quite a mass of points for a simple one-sided optimization problem. Figure 4.10 shows that mtEGO needs quite many points to locate the global optimum very precisely in the center, although it is known from cycle four where the optimum roughly is lying.

### 4.4.2 Two-sided multivariate test problem using Derringer-Suich desirability functions

In Section 4.1 the test problem VLMOP3 is considered as a target value problem using symmetric two-sided Harrington desirability functions. In the following the target value problem is specified with asymmetric two-sided Derringer-Suich desirability functions. The target and hence the global optimum are the same as in Section 4.1, while the deviations in different directions are penalized differently. The contour of the true desirability index (which is again the unweighted geometric mean) and therefore the progress of the heuristic is different than for the symmetric Harrington functions (cf. Figure 4.9b). The specification parameters of the Derringer-Suich desirability functions here are

| objective | $y_1$ | $y_2$ | $y_3$ |
|---|---|---|---|
| $target$ | 4 | 30 | 0.15 |
| $LSL$ | 0 | 15 | 0.1 |
| $USL$ | 8 | 37 | 0.17 |
| $r$ | 1 | 1 | 1 |
| $l$ | 2 | 2 | 2 |

.

The joint global optima are the parameter settings (0.6, -2.4) and (2.5, 0.3). Table 4.8 and Figure 4.11 give details on the optimization of this example using $\alpha \in \{0.01, 0.1, 0.5, 0.9\}$.

The optimization again starts with only 5 design points and a poor initial model. However, already in the third cycle the region of one of the global optima is approached. mtEGO finds the first global optimum in the fourth cycle and proceeds examining the region of the second optimum during the fifth to seventh cycle, approaching it more and more. In the eighth cycle both optima are found. The stopping criterion decreases monotonically and starts saturating in cycle five. As the saturation is not very close to 0, we do more cycles until $relmaximax$ really saturates in the eighth cycle. mtEGO succeeds to meet the target within four cycles and finds both optima in eight cycles. It needs 34 experiments in total until both optima are found and the stopping criterion saturates. However, a slight saturation can already be observed after five cycles when the first optimum is found. Hence, mtEGO is also able to handle asymmetric desirability specifications.

| cy-cle | # of design points | currently observed optimum | currently predicted optimum | model fit | $relmax-imax$ | global optima found? | # of added points | location of updatings |
|---|---|---|---|---|---|---|---|---|
| 1 | 5 | (1.8,-1.5) | (0.9,-1.7) | bad | 6.981 | not found | 6 | few clumping |
| 2 | 11 | (1.8,-1.5) | (0.7,-2.2) | better | 3.794 | not found | 5 | few clumping |
| 3 | 16 | (0.5,-2.3) | (0.6,-2.4) | moderate | 2.166 | approaching 1.opt | 5 | ok |
| 4 | 21 | (0.6,-2.4) | (2.4,-0.1) | ok | 1.356 | 1.opt found | 5 | good |
| 5 | 26 | (0.6,-2.4) | (2.4,0.2) | ok | 1.212 | approaching 2.opt | 3 | good |
| 6 | 29 | (0.6,-2.4) | (0.6,-2.4) | ok | 1.148 | 1.opt found | 2 | good |
| 7 | 31 | (2.5,0.4) | (2.5,0.3) | ok | 1.072 | 2.opt close | 3 | good |
| 8 | 34 | (2.5,0.3) | (2.5,0.3) | ok | 1.038 | both opts found | stop | |

Table 4.8: Progress of the optimization of a multivariate test problem using the new heuristic and Derringer-Suich desirabilities

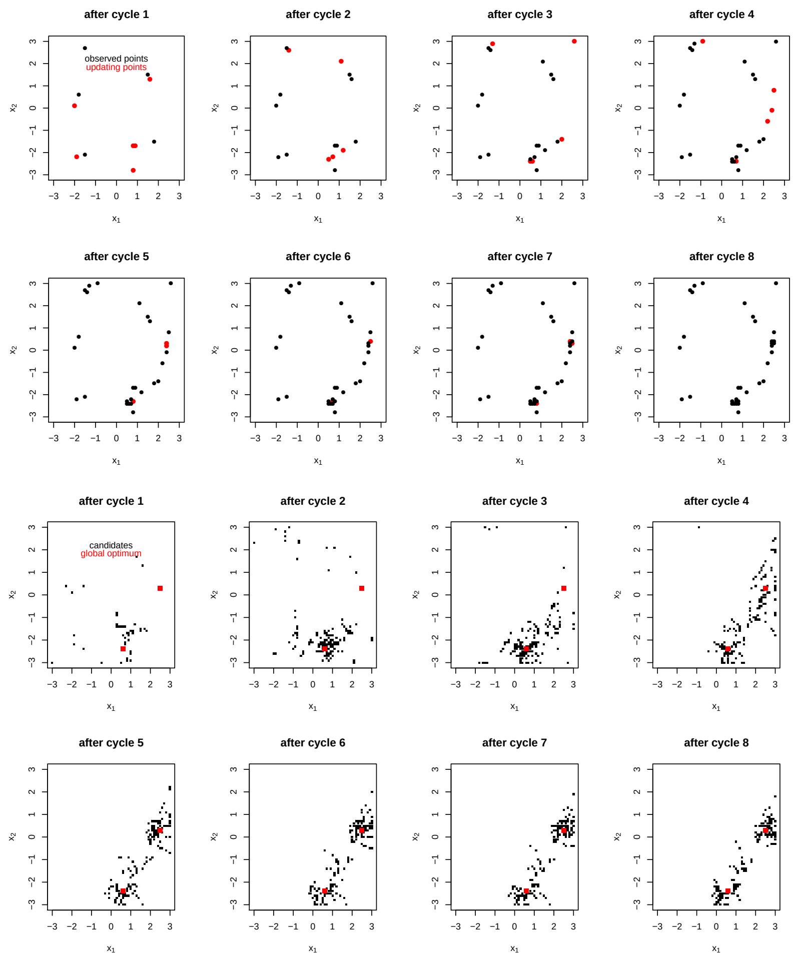


Figure 4.11: Progress of the candidates and updating points for a two-sided multivariate test problem using Derringer-suich desirabilities

### 4.4.3 Two-sided multivariate test problem with weighted objectives

In this section the target-value optimization with Derringer-Suich desirabilities from the previous subsection is varied using a weighted geometric mean as desirability index (cf. Equation 2.39). The specification of the desirability functions for the three objectives is the same as before, but they are weighted with $w_1 = 0.2, w_2 = 0.2$ and $w_3 = 0.6$ during the aggregation to one desirability index. The third objective $y_3$ is more important to be optimal in comparison with $y_1$ and $y_2$, that have the same weights. Giving $y_3$ a larger weight than the other objectives, the contour of the true desirability index (see Figure 4.9c) and also the joint optimum shift towards the target value of $y_3$. The new global optimum can be found in the coordinates (0.8, -2.2). An almost equally sized local optimum is located at (2.3, 0.4).

| cy-cle | # of design points | currently observed optimum | currently predicted optimum | model fit | $relmax-imax$ | global optimum found? | # of added points | location of updatings |
|---|---|---|---|---|---|---|---|---|
| 1 | 5 | (1.8,-1.5) | (0,-1.2) | bad | 4.902 | not found | 7 | ok |
| 2 | 12 | (1.8,-1.5) | (0.8,-3) | better | 2.462 | not found | 5 | ok |
| 3 | 17 | (1.8,-1.5) | (0.7,-2.2) | moderate | 2.13 | approaching global opt | 5 | good |
| 4 | 22 | (0.9,-2.2) | (0.8,-2.2) | moderate | 1.214 | global opt close | 5 | good |
| 5 | 27 | (0.8,-2.2) | (1.8,-0.8) | ok | 0.989 | global opt found | 4 | good |
| 6 | 31 | (0.8,-2.2) | (2.5,-0.1) | ok | 0.91 | global opt found & local opt close | stop | |

Table 4.9: Progress of the optimization of a multivariate test problem with weighted objectives

Table 4.9 shows that mtEGO also is able to optimize this target-value problem that includes the weighted geometric mean as desirability index. mtEGO successfully finds the global optimum within 31 experiments and six cycles. Figure 4.12 shows how the region around the global optimum as well as the region around the local optimum is examined. mtEGO starts approaching the global optimum almost from beginning on. The currently predicted optimum approaches the global optimum in each cycle, although the observed optimum stays at the firstly observed best point for the first three cycles. The stopping criterion saturates in the sixth cycle and, actually, the global optimum is exactly found now. The local optimum is at least roughly located.

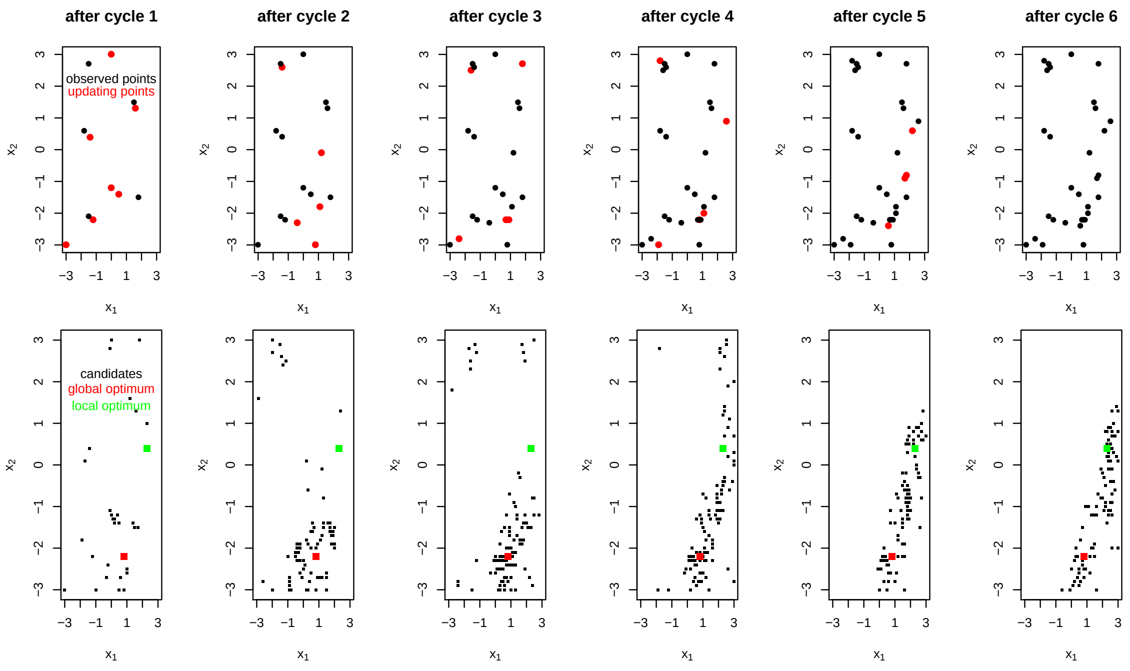


Figure 4.12: Progress of the candidates and updating points for a two-sided multivariate test problem with a weighted desirability index

### 4.4.4 A multivariate test problem with complex mixed target structure

The following test problem covers almost the whole range of complexity of the target structure of a test problem that can be handled with mtEGO. As a basis for this example the objectives of test problem VLMOP3 are used again. Different desirability functions are used to specify completely different targets for the three objectives. Objective $y_1$ is to be minimized and we use a one-sided Harrington function with $b_0 = -1.8$ and $b_1 = 0.5$. The second objective $y_2$ shall reach the target value $T = 46$ and deviations to both sides are equally bad. Hence a symmetric two-sided Harrington function with $LSL = 25, USL = 67$ and $\nu = 2$ is specified. $y_3$ also is to be optimized regards a target value ($T = 0.15$) but it is worse if the target is exceeded than if it is underrun. Therefore the specified desirability function is an asymmetric Derringer-Suich desirability function with parameters $LSL = 0.1, USL = 0.17, l = 2$ and $r = 1$. All objectives are aggregated to the desirability index shown in Figure 4.9d using the unweighted geometric mean. The resulting global optimum is the point $(2.1, -1.4)$.

| cycle | # of design points | currently observed optimum | currently predicted optimum | model fit | *relmax−imax* | global optimum found? | # of added points | location of updatings |
|---|---|---|---|---|---|---|---|---|
| 1 | 5 | (1.8,-1.5) | (1.8,-1.5) | bad | 6.978 | not found | 4 | ok |
| 2 | 9 | (1.8,-1.5) | (2.7,-1.2) | better | 3.886 | approaching opt | 9 | ok |
| 3 | 18 | (1.9,-1.8) | (1.7,-1.8) | moderate | 3.209 | approaching opt | 4 | good |
| 4 | 22 | (1.7,-1.8) | (1.5,-2) | moderate | 2.765 | not found | 5 | good |
| 5 | 27 | (1.7,-1.8) | (2,-1.5) | ok | 2.489 | approaching opt | 4 | good |
| 6 | 31 | (2,-1.5) | (2.1,-1.4) | ok | 2.183 | opt close | 5 | very good |
| 7 | 36 | (2.1,-1.4) | (2.2,-1.2) | ok | 1.944 | opt found | stop | |

Table 4.10: Progress of the optimization of a multivariate test problem with complex mixed target structure

Details of the optimization procedure for this example using $\alpha \in \{0.01, 0.1, 0.5, 0.9\}$ are given in Table 4.10. In total, mtEGO needs seven cycles and 36 experiments to find the optimum. After adding many updating points that are scattered quite regularly in the parameter space in cycle one and two, the observed optimum lies already close to the true global optimum. mtEGO then proceeds searching more locally around the observed optimum in the fourth cycle. The stopping criterion starts slightly saturating but still improves. Since the level of saturation is far from 0, we do more cycles. We stop the algorithm after seven cycles, although the stopping criterion is not clearly saturating. However, the predicted and observed optima move very little around the same point and the improvement in the stopping criterion is very little. The progress of the candidates displayed in Figure 4.13 also indicate to stop in the seventh cycle. The candidates concentrate more and more on one region. Furthermore, in the sixth cycle only two updating points are added in two different promising regions. One of the regions disappears in the seventh cycle and the other region is the already well observed region with the currently observed optimum. We therefore stop in the seventh cycle and indeed find the global optimum. Altogether, mtEGO optimized this exemplary problem with complex mixed target structure successfully, which confirms the high flexibility of the new algorithm regarding the specification of the optimization target.

### 4.4.5 A large-dimensional target optimization problem

To show that mtEGO is also able to find a global optimum in optimization problems with more dimensions than two inputs or three objectives, here also an example with five inputs and five objectives is run exemplarily with the parameterization

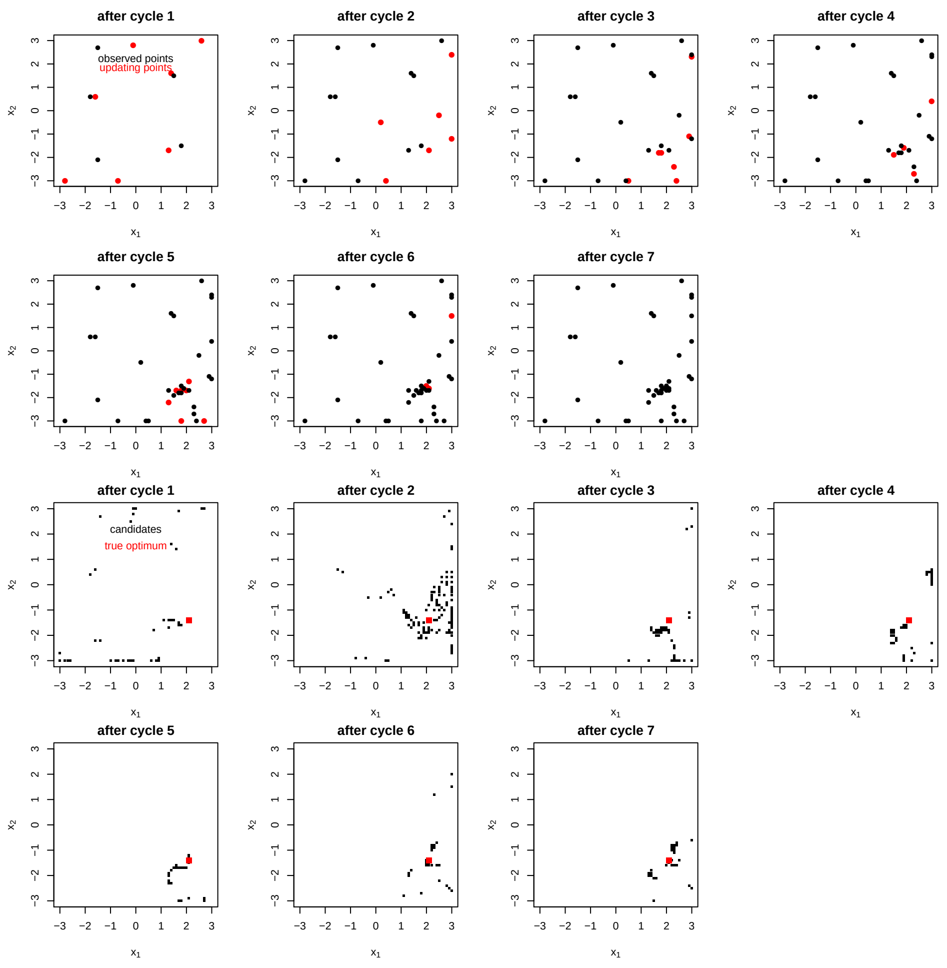


Figure 4.13: Progress of the candidates and updating points for a test problem with complex mixed target structure

$\alpha \in \{0.01, 0.1, 0.5, 0.9\}$. Different $\alpha$ parameterizations are only tested in the case study in Section 4.2 due to time reasons. One optimization cycle of this example with only four $\alpha$ levels takes already approximately two hours on a 2 GHz machine with the current implementation. The basis of this example are the objectives from test problem DTLZ2 (see Section 4.1). To restrict the complexity of this example we used two-sided Harrington desirability functions for all five objectives and the unweighted geometric mean as desirability index. The desirability functions are thereby specified with the following parameters

| | $LSL$ | target value | $USL$ | $\nu$ |
|---|---|---|---|---|
| $y_1$ | 0 | 4 | 8 | 2 |
| $y_2$ | 0 | 20 | 40 | 2 |
| $y_3$ | 0.5 | 6 | 11.5 | 2 |
| $y_4$ | 0 | 15 | 30 | 2 |
| $y_5$ | 10 | 50 | 90 | 2 |

.

This specification of desirabilities results in one global optimum that can be found in the coordinates $(0.8, 0.4, 0.2, 0.8, 0)$. The large dimensionality of the example impedes a graphical illustration of the desirability index, the model, or the location of the design points. To supervise the progress of the optimization appropriately we compare the 20 best predictions with the truly 20 best parameter settings to get an idea of the number of almost equally sized local optima and their location regarding the known global optimum.

Additionally, we use graphics like the one shown in Figure 4.14 to roughly supervise the progress of the updating points. Each colored line in the plot illustrates for one updating point its Euclidian distance to all observed design points. The upper black horizontal line shows the maximum Euclidian distance between the initial design points and the lower black line shows the Euclidian distance between the grid points of the parameter space. If all lines are very close to the upper line, the new updating points are scattered very regular in the parameter space similar to a stochastic search. This is

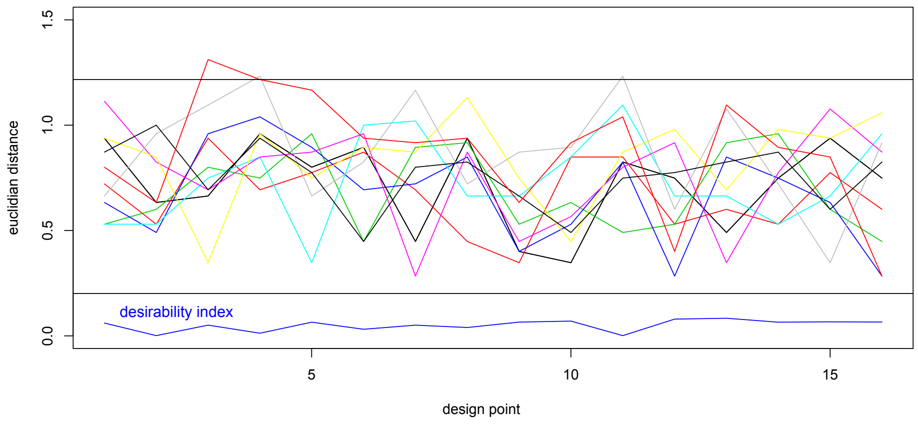


Figure 4.14: Euclidian distances between updating points and the already observed design points

only acceptable if the initial model is very bad. If all lines touch the lower line at any point, this means all updating points are lying next to an already existing point, i.e. they are clumping. Figure 4.14 shows a reasonable distribution of the updating points. The lines are far away from those design points with low desirability index (i.e. the lines are not touching the lower line there). Some lines are close to the design points 3, 5, 7, 12, 15 and 16 (i.e. they almost touch the lower line there). These design points are those with the highest desirability indices among the observations. The line of the desirability index shows peaks in these places. Hence, this figure implicates that the optimization procedure shifts towards the current optimum. Table 4.11 gives details for the progress of the optimization of this large-dimensional example. mtEGO successfully finds the global optima in eight cycles with 55 experiments in total. Note, that this is approximately the number of experiments Jones et al. suggest to use already for the initial space-filling design. Already in the second cycle mtEGO starts approaching the global optimum. In the fifth cycle the observed optimum is already lying close to the true optimum. The true optimum is found in the seventh step. mtEGO thereby starts adding many widely scattered updating points in the first cycles and less points that concentrate on the region of the global optimum are added in the later cycles. Unfortunately, the stopping criterion does not indicate to stop very clearly. It starts saturating at a level of approximately 2.3 already in the third cycle. It improves again from cycle four to seven, and saturates at a level of approximately 1 in the last step again. Usually, we would proceed the search when the saturation is uncertain and at a level larger than 1. But in the eighth cycle the observed and predicted optimum are the same, which finally indicates to stop. However, the important result is the ability of mtEGO to find a global optimum even in large dimensional problems. The supervision of the progress is a complex task in large-dimensional problems as shown above, but the example demonstrates that mtEGO searches quite straight forward and the stopping criterion also works well.

In Section 4.3 we conclude with the statement, that mtEGO clearly dominates the brute force methods for large or more complex examples. We therefore do the same comparisons using random points and the coffee-house design as in Section 4.3 for this large-dimensional example. In all simulation runs mtEGO achieves better results then the other methods, i.e. the percentage where mtEGO is better is 100 %. This example, hence, emphasizes that mtEGO is powerful especially in the case of complex optimization problems.

| cycle | # of design points | currently observed optimum | currently predicted optimum | *relmaximax* | global optimum found? | # of updating points | location of updatings |
|---|---|---|---|---|---|---|---|
| 1 | 8 | (0.8,0.8,0.4,0.4,0.2) | (0.6,0.2,0.2,0.6,1) | 6.104 | no | 7 | ok |
| 2 | 16 | (0.4,0.6,0.2,0.8,0.2) | (0.4,0.2,0.2,0.4,1) | 2.286 | approaching opt | 10 | ok (Figure 4.14) |
| 3 | 26 | (0.6,0.2,0.2,0.8,0.4) | (0.4,0,0,0.2,1) | 2.303 | approaching opt | 8 | ok |
| 4 | 34 | (0.6,0.2,0.2,0.8,0.4) | (0.6,0.4,0.2,1,0) | 2.278 | approaching opt | 7 | ok |
| 5 | 41 | (0.6,0.2,0.2,0.8,0.2) | (0.6,0.4,0.2,0.8,0) | 1.919 | opt close | 5 | ok |
| 6 | 46 | (0.6,0.4,0.2,0.8,0) | (0.8,0.4,0.2,0.8,0) | 1.223 | opt very close | 4 | ok |
| 7 | 50 | (0.8,0.4,0.2,0.8,0) | (0.2,0,0,0.2,1) | 0.944 | opt found | 5 | ok |
| 8 | 55 | (0.8,0.4,0.2,0.8,0) | (0.8,0.4,0.2,0.8,0) | 1.031 | opt found | stop | |

Table 4.11: Progress of the optimization of the large-dimensional optimization example

| # inputs | # grid points | 2 objectives | 3 objectives | 4 objectives | 5 objectives | 6 objectives |
|---|---|---|---|---|---|---|
| 3 | 1000 | 11.6s | 20.3s | 1.7m | 14.6m | 2.6h |
| 4 | 1000 | 12.9s | 21.5s | 1.7m | 14.3m | 2.7h |
| 5 | 1000 | 14.3s | 23.0s | 1.7m | 14.7m | 2.7h |
| 3 | 5000 | 57.8s | 1.7m | 8.5m | 1.2h | 12.0h |
| 4 | 5000 | 1.1m | 1.8m | 8.6m | 1.2h | 12.1h |
| 5 | 5000 | 1.23m | 1.9m | 8.7m | 1.4h | 12.1h |
| 3 | 10000 | 2.0m | 3.5m | 17.6m | 2.5h | |
| 4 | 10000 | 2.2m | 3.7m | 17.7m | 2.5h | |
| 5 | 10000 | 2.4m | 3.9m | 18.0m | 2.5h | |
| 3 | 50000 | 10.5m | 18.5m | 1.5h | 12.5h | |
| 4 | 50000 | 11.8m | 19.6m | 1.6h | 12.5h | |
| 5 | 50000 | 13.0m | 20.5m | 1.6h | 12.5h | |
| 3 | 100000 | 23.2m | 40.5m | 3.0h | | |
| 4 | 100000 | 25.6m | 42.1m | 3.1h | | |
| 5 | 100000 | 27.5m | 43.9m | 3.1h | | |
| 3 | 500000 | 3.9h | 5.7h | 19.5h | | |
| 4 | 500000 | 4.1h | 5.8h | 19.5h | | |
| 5 | 500000 | 4.9h | 6.4h | 19.5h | | |

Table 4.12: Computation time for one optimization cycle for $\alpha \in \{0.01, 0.1, 0.5, 0.9\}$

## 4.5 Limitations of the approach

During the simulation study on mtEGO, some time limitations of the current implementation (see Appendix A.6) became obvious. We define mtEGO to be infeasible when it needs more than 10 hours on a 2GHz computer with 2GB RAM for one optimization cycle. Whether mtEGO is feasible or not depends on the number of $\alpha$ levels, the number of objectives, and the size of the grid representing the parameter space, which directly depends on the number of influencing parameters and levels per parameter. Table 4.12 shows an overview of the computation time needed for one cycle of the heuristic.

We tested mtEGO for three to six objectives using four $\alpha$ levels $\{0.01, 0.1, 0.5, 0.9\}$.

For each number of objectives we determined the computation time for parameter spaces represented by grids with 1000, 5000, 10000, 50000, 100000, 500000 points. The more factor levels per input parameter, the larger becomes the grid. When four influencing factors are present with 10 levels per factor the four-dimensional parameter space is a grid with 10000 points. Having five factors with 10 levels each, the five-dimensional parameter space is a grid with 100000 grid points. The computation time of mtEGO depends almost entirely on the number of grid points and not on the number of influencing factors directly. Table 4.12 shows that the difference in computation time for different numbers of influencing parameters is marginal for the same size of parameter space grids. This means that if the dimensionality of the parameter space gets larger, maybe because new significant factors are found and the computation time shall not be extended, less levels per factor can be optimized to keep the parameter space grid the same size. We show all times for completeness, but we are mainly interested in the time differences between the different numbers of objectives. Already for four $\alpha$ levels the limitation of 10 hours is reached soon. For two and three objectives more than 500000 grid points can be optimized within 10 hours. But the heuristic needs more than 10 hours already with 50000 grid points for five objectives. For six objectives the runtime for one cycle of mtEGO is already 2.6h for 1000 grid points. One optimization cycle for seven objectives needs already 42 minutes for 125 grid points and for a 1000 grid point parameter space it would need more than one day. Fewer than four $\alpha$ levels would allow us to optimize more objectives or grid points within 10 hours of computation time. However, it is not reasonable to use fewer than three levels, since mtEGO yields either in a very local search if narrow boundaries are chosen, or it is very unprecise if wide boundaries are chosen. More $\alpha$ levels are only feasible for five objectives at maximum. Altogether, mtEGO gets very slow and infeasible for more than six objectives with the current implementation and a 2GHz machine with 2GB RAM regardless the number of $\alpha$ levels used for the heuristic.

In the following we try to explain the complexity of computation for the heuristic more theoretically. Suppose, $b$ is the number of objectives, $q$ is the number of grid points representing the $p$-dimensional parameter space and $a$ is the number of $\alpha$ levels used in the optimization. According to Section 3.2, firstly, all virtual observations are constructed in Step 3.A.2. Altogether, $2a + 1$ virtual observations per objective and per grid point are determined (i.e. $(2a + 1)bq$ virtual observations in total) Respectively, $(2a+1)bq$ arithmetic operations are conducted. The same amount of operations is needed to transform all virtual observations to desirability values in Step 3.A.3. Hence, up to now $2(2a+1)bq$ operations are needed. For $a$ confidence levels and $b$ number of objectives, the cross-combination of the desirabilites to desirability indices in Step A.3 results in $a^b$ indices. To determine the indices, one arithmetic operation is needed for each. Then all indices are transformed to improvement values, which are again $a^b$ operations. This sums up to $(2(2a+1)bq)+(2a^b)$ operations already. Finally, the candidates (i.e. the points among the $q$ grid points that maximize the improvements vectors according to Step 3.B.3) are determined, saved, clustered in groups, and a representative for each group is chosen. All those steps need computation time, too, where the runtime depends on the kind of algorithm used to find the maxima and the clustering algorithm that are used. Note, that the runtime of the clustering only depends on the number of candidates that is at maximum $a^b$ and on the dimensionality of the parameter space, but is independent of the actual number of grid points. Hence, beside the search for the maximum of the improvements and the final clustering of the candidates, the heuristic needs $(2(2a+1)bq)+(2a^d) = 4aqb+2qb+2a^b$ operations. The formula indicates the poor algorithmic efficiency and the need for a good implementation. Note, that $p$ is not present in the formula above. The time differences between different numbers of influencing factors in Table 4.12 results from time needed for saving data and matrices in the memory, which is proportional to $p$.

As a conclusion, we recommend to implement mtEGO in a more efficient way for applications with large dimensions. For example a more adequate programming language like C++ could be used. Further, the heuristic also gives the opportunity to be parallelized for grid computing. E.g. the construction of the virtual observations and desirabilities for each objective separately can be done parallel and even the determination of the different desirability indices and improvements with their maximizing points can be parallelized, since they are determined independently from each other. Additionally, a more efficient extension of mtEGO for large dimensions is introduced in the next section.

# 5 Extensions to the new approach

## 5.1 A variant for large dimensional optimization problems

In the previous chapter the new heuristic mtEGO is tested with an extensive simulation study. It is shown that mtEGO finds the global optimum reliably, but gets computationally expensive and slow for a large number of $\alpha$ levels, inputs or objectives. Here, an improved version mtEGOimp that is computationally more efficient than mtEGO is presented subsequently.

The improved version mtEGOimp basically works just like the original mtEGO algorithm from Section 3.2. The overall procedure with the five steps stays the same. The improvement of the original mtEGO algorithm takes place in Step 3.A.1. Originally, the set of $\alpha$ levels is determined and used completely in Step 3.A.2. In mtEGOimp Step 3.A.1 is extended with a pre-selection among the manually chosen $\alpha$ levels according to a simple decision rule. The suggested $\alpha$ levels are reduced to those levels that are reasonable for the specific problem that is to be optimized. mtEGOimp then proceeds with the construction of the virtual observations (i.e. Step 3.A.2) and the remaining procedure exactly like the original mtEGO approach.

Figure 5.1 motivates the idea of the reduction of the suggested $\alpha$ levels. Assume the five $\alpha$ levels $\{0.01, 0.02, 0.1, 0.25, 0.5\}$ are chosen for the optimization. The plot shows the corresponding virtual observations for one exemplary objective. The target for this objective lies below the predictions $\hat{y}$ for all univariate parameter settings $x_1$. Clearly, the upper $(1 - \alpha/2)$ confidence boundaries for all $\alpha$ levels (i.e. the dashed lines) are virtual observations that lie even more remote from the target than the predictions $\hat{y}$ do. Hence, these virtual observations do not provide any additional information about possible global optima. They are useless during the optimization procedure and we can leave them out. The same would apply to all lower confidence boundaries, if

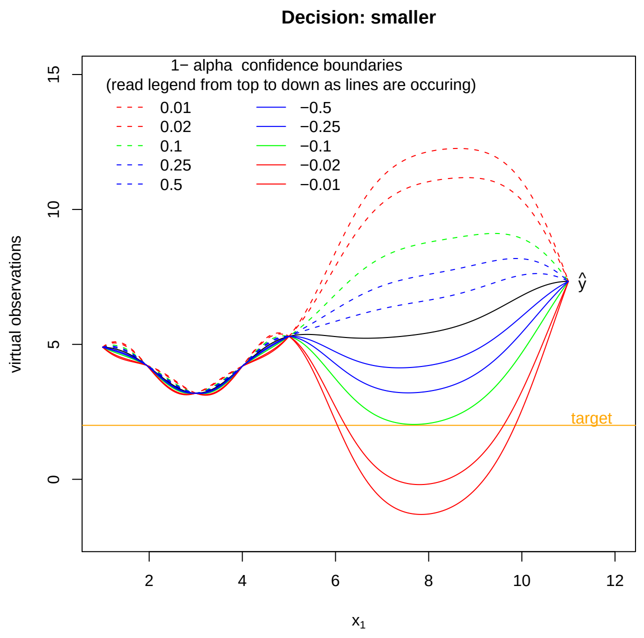


Figure 5.1: Motivation of the reduction of $\alpha$ levels to smaller $\alpha$ levels only

an objective would lie below the target for all parameter settings. But there is more potential to reduce the used $\alpha$ levels in this example. Among the solid blue and green colored virtual observations the distance is minimum to the target for almost the same parameter settings. They indicate almost the same candidates for the global optimum. Here, the solid red colored virtual observations and only one among the solid blue or green virtual observations belong to the relevant $\alpha$ levels for the optimization. All other virtual observations can be skipped.

The situation is contrary for the exemplary objective in Figure 5.2. Like for the first example all upper levels can be skipped, since the predictions all lie above the target. But here the virtual observations with small $\alpha$ levels (i.e. the wide confidence boundaries, solid red lines) and those with the medium sized $\alpha$ level (solid green line) yield nearly the same candidates. Hence, using all three virtual observations during the optimization process actually is a waste of time. Mainly the virtual observations with the large $\alpha$ levels (solid blue lines) provide new information about possible optima for this example. The optimization should be done with the $\alpha$ levels belonging to the solid blue lines and the solid green line to use the maximum information with a minimum number of $\alpha$ levels.

Figure 5.3 shows a third example where the decision is not as easy as for the first

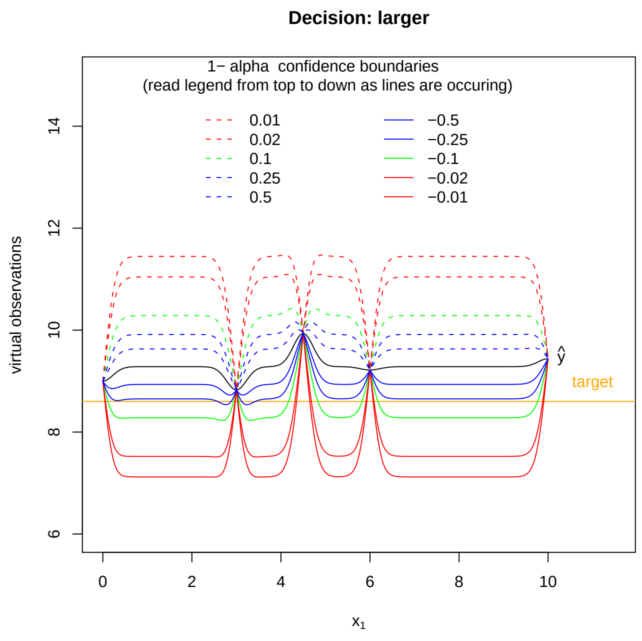


Figure 5.2: Motivation of the reduction of $\alpha$ levels to larger $\alpha$ levels only

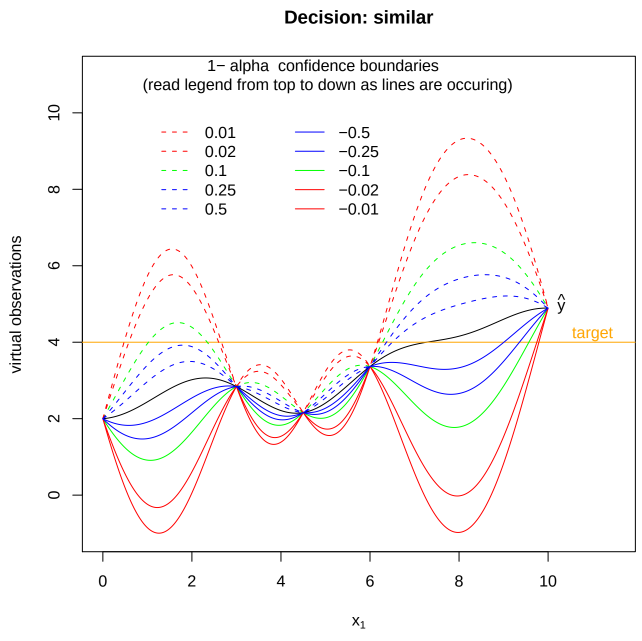


Figure 5.3: Motivation of the reduction of $\alpha$ levels to similar $\alpha$ levels only

two examples. Neither all upper, nor all lower confidence boundaries can be excluded right away, since the predictions lie below the target for small $x_1$ and then exceed it for large $x_1$. If we concentrate on the interval $[0, 3]$ the two dashed blue lines give almost the same information and the two dashed red lines give the same information. But the dashed blue, the dashed green and the dashed red lines give different information and one of each group should be used for the optimization. The solid lines are not needed in this intersection. The virtual observations in the next interval $[3, 6]$ are very narrow because of a small prediction error. Here all solid lines can be skipped and all dashed lines result in the same candidate. One level would be sufficient. In the last intersection $[6, 10]$ the virtual observations printed with dashed lines give candidates that lie very close to each other and could be represented just by one of the levels, e.g. the green colored medium sized $\alpha$ level. The virtual observations printed with solid lines also yield candidates that lie quite close to each other and could be represented only by the green medium sized $\alpha$ level. Altogether, for this third interval we need upper and lower confidence boundaries and it would be reasonable to use the $\alpha$ levels $\{0.02, 0.1, 0.25\}$, i.e. one red, the green and one blue line for the optimization.

However, the decision which $\alpha$ levels are reasonable to be used or skipped should happen automatically. The first rule is to skip all upper (lower) boundaries if the predictions are lying above (below) the target for all parameter settings $x_i \in \chi, i = 1, ..., q$, where $\chi$ is the parameter space grid. Additionally, without mentioning it yet, we used the green colored virtual observations as reference confidence boundaries in all three examples to test whether smaller, larger or similar $\alpha$ levels compared to the reference boundaries should be used. In the decision rule, we define reference confidence boundaries specified with a medium-sized $\alpha$ level $dec\alpha$ (resulting in $y_{+dec\alpha}$ and $y_{-dec\alpha}$) and a 25% decision interval around the target value $T$, i.e. $[T_{-0.25}, T_{+0.25}] = [T - 0.25T, T + 0.25T]$. The reference boundaries are compared to the decision interval $[T_{-0.25}, T_{+0.25}]$ and that for each parameter setting $x_i$ separately. If $\hat{y}(x_i)$ is larger than the target value, we take the lower reference boundary $y_{-dec\alpha}$ and check if it lies inside, below or above the interval $[T_{-0.25}, T_{+0.25}]$. If it lies inside the interval, similar $\alpha$ levels are chosen. If it lies below although the prediction lay above the target, the reference boundary is already wide enough and it is reasonable to choose larger $\alpha$ levels. If it lies above the interval the reference boundary still is far away from the target and wider boundaries which means smaller $\alpha$ levels should be included in the optimization. For predictions $\hat{y}(x_i)$ that are smaller than the target the decision is exactly vice versa.

Figure 5.4 displays the whole decision rule well-arranged. For each parameter setting $x_i$ this decision tree is evaluated. It results in a decision 'smaller', 'larger' or 'similar' for each setting $x_i$. Finally, we chose the most frequently decision among the resulting $q$ decisions.

Suppose a beforehand predefined complete set of $\alpha$ levels, e.g. $\{0.01, 0.02, 0.1, 0.25, 0.5\}$, that would have been used for the original heuristic, is split into a reference level 0.1, large levels {0.25, 0.5} and small levels {0.01, 0.02}. If the most frequent decision is 'smaller' or 'larger', the corresponding levels together with the reference level are used for the following optimization procedure. If the most frequent decision is 'similar' we use the reference level, half of the small levels (the largest among them) and half of the large levels (the smallest among them). The chosen levels are then used to determine upper and or lower confidence boundaries as virtual observations depending on the fact whether all predictions lie above, below or alternating compared to the target. Hence, for five levels the number of virtual observations for one objective is reduced from eleven to only seven or even four virtual observations. For problems with many objectives the reduction of computationally complexity is remarkable when only four instead of eleven virtual observations are cross-combined in the following steps of the optimization algorithm.

Note, that the decision rule described before and illustrated in Figure 5.4 is only valid for target value problems. For the one-sided case, i.e. minimization or maximization problems no actual target value is known and the decision rule illustrated in Figure 5.5 is used. For a better understanding of this tree, suppose the objective in Figure 5.1 is to be minimized. Obviously, all upper boundaries do not give any useful information, since they can only yield larger and thus worse values than the currently found minimum $y_{min}$ (at $x = 3$). All upper boundaries can immediately be excluded from the optimization. The reference level (solid green line) yields smaller and thus better values than the currently found minimum in the uncertain area. Each smaller $\alpha$ level (i.e. wider boundaries) yield the same minimum as the reference level then. Hence, we are not interested in the smaller levels that do not provide new information. We can concentrate on the larger $\alpha$ levels to include also the local search around the current optimum. The reference level already ensures that the algorithm searches globally. If the upper solid blue line would represent the reference level here, its minimum would lie around the currently observed minimum. This boundary lies almost completely above the current minimum. Then we are of course interested in wider boundaries (i.e.

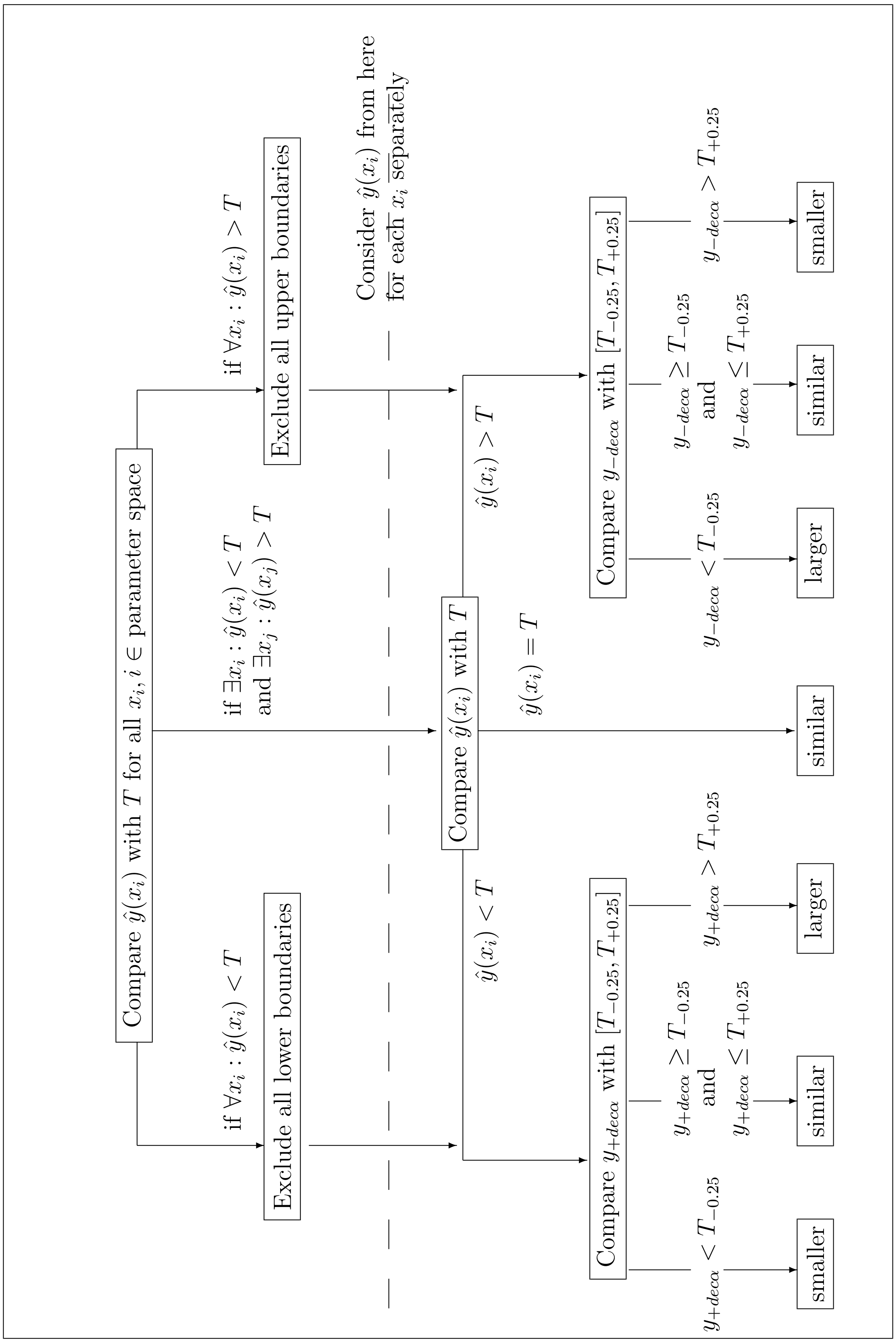

Compare $\hat{y}(x_i)$ with $T$ for all $x_i, i \in$ parameter space
if $\forall x_i : \hat{y}(x_i) > T$
if $\exists x_i : \hat{y}(x_i) < T$ and $\exists x_j : \hat{y}(x_j) > T$
if $\forall x_i : \hat{y}(x_i) < T$
Exclude all upper boundaries
Exclude all lower boundaries
Consider $\hat{y}(x_i)$ from here for each $x_i$ separately
Compare $\hat{y}(x_i)$ with $T$
$\hat{y}(x_i) > T$
$\hat{y}(x_i) = T$
$\hat{y}(x_i) < T$
Compare $y_{-dec\alpha}$ with $[T_{-0.25}, T_{+0.25}]$
$y_{-dec\alpha} > T_{+0.25}$
smaller
$y_{-dec\alpha} \geq T_{-0.25}$ and $y_{-dec\alpha} \leq T_{+0.25}$
similar
$y_{-dec\alpha} < T_{-0.25}$
larger
similar
Compare $y_{+dec\alpha}$ with $[T_{-0.25}, T_{+0.25}]$
$y_{+dec\alpha} > T_{+0.25}$
larger
$y_{+dec\alpha} \geq T_{-0.25}$ and $y_{+dec\alpha} \leq T_{+0.25}$
similar
$y_{+dec\alpha} < T_{-0.25}$
smaller

Figure 5.4: Decision tree for the question if smaller, larger or similar $\alpha$'s compared to the reference level should be taken for target-value optimization problems

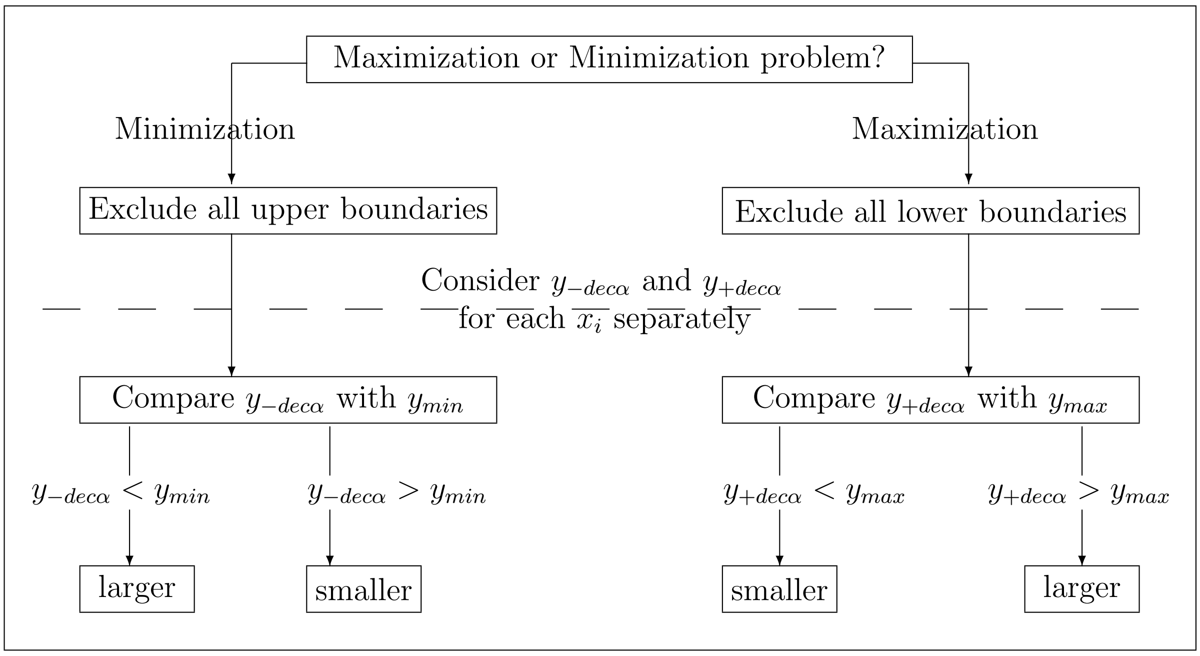

Maximization or Minimization problem?
Minimization
Maximization
Exclude all upper boundaries
Exclude all lower boundaries
Consider $y_{-dec\alpha}$ and $y_{+dec\alpha}$ for each $x_i$ separately
Compare $y_{-dec\alpha}$ with $y_{min}$
Compare $y_{+dec\alpha}$ with $y_{max}$
$y_{-dec\alpha} < y_{min}$
$y_{-dec\alpha} > y_{min}$
$y_{+dec\alpha} < y_{max}$
$y_{+dec\alpha} > y_{max}$
larger
smaller
smaller
larger

Figure 5.5: Decision tree for the question if smaller or larger $\alpha$'s compared to the reference level should be taken for one-sided optimization problems

smaller $\alpha$ levels) to ensure a global search.

If the optimization problem is a maximization problem the decision has to be exactly the opposite way round comparing the reference boundary with $y_{max}$. The decision to use 'similar' levels is not needed in the one-sided optimization problems. The nature of minimization and maximization are much simpler than the target-vale problem.

We also tried to differentiate the decisions on the $\alpha$ reduction between the separate uncertain areas. The problem here is to define the separate uncertain areas. As one can see in Figure 5.6a, this problem is trivial for the univariate parameter space. The grey vertical lines show a clear separation of the different uncertain areas between the known observations. Already for a bivariate parameter space the uncertain subspaces can not be well-defined. An example design is shown in Figure 5.6b. Figure5.6c shows the contour of the prediction error for that example. There are areas of large and small uncertainty, but the borders are flowing. This makes it difficult to find a criterion to split the parameter space in different uncertain areas automatically. Therefore, we evaluate each candidate individually and then choose the most frequent decision.

Simulation studies with mtEGOimp show that the extension is working well. For the large dimensional example in Section 4.4.5 the computation time for one optimiza-

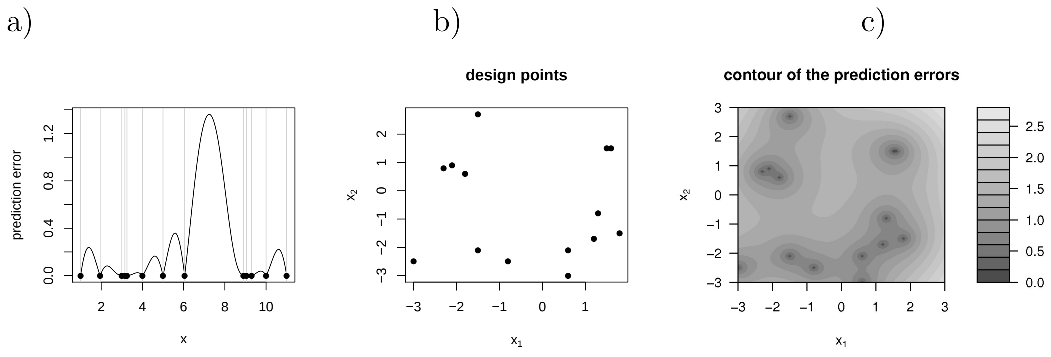


Figure 5.6: Separation of uncertain areas for a) univariate and b)&c) bivariate design space

tion cycle is reduced from approximately two hours to 30 minutes with mtEGOimp. Meanwhile, the global optima are found with about the same number of cycles and experiments. Unfortunately, the time needed to decide which $\alpha$ levels can be excluded, compensates the time savings of the shorter $\alpha$ vector for low dimensional optimization problems. The originally introduced heuristic from Section 3.2 should be used if fewer than five objectives are present. The comments on stopping rules, choices for $\alpha$, and the evaluation of the progress of the optimization hold for the extended heuristics as well.

## 5.2 Optimization in the presence of unknown constraints

Forrester et al. (2006) state that "in an ideal world, a seamless parameterization would ensure that the optimizer could visit wide ranging geometries and move between them without interruption. In reality, however, [...] in all but the most trivial or clearly constrained cases the design evaluation can fail in certain areas of the search space." The mtEGO approach can easily be adapted to this case of unknown constraints. The main problem with unknown constraints is, that eventually experiments are planned that actually lie inside the unknown constrained region. The evaluation of these experiments fails and missing values are present in the design. In sequential optimization the aim is to find a global optimum with as few experiments as possible. The initial designs are very small and the presence of missing values distorts the surrogate model. The

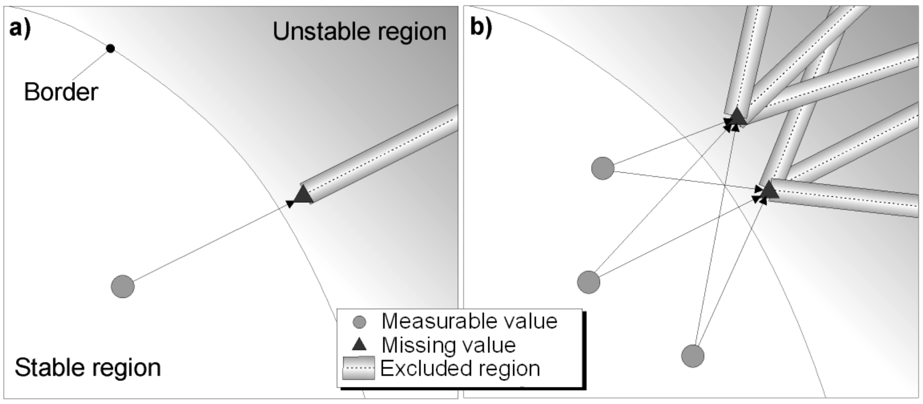


Figure 5.7: Idea of the convex hull strategy (from Henkenjohann et al., 2005)

next subsection provides a procedure to generate an initial design that is space-filling and includes enough so called "non-failures" to enable a good surrogate model. The occurrence of missing values during the updating process is even worse. Why this is such a great problem and how it is solved will be explained in Section 5.2.2.

### 5.2.1 Generation of an appropriate initial design

Henkenjohann et al. (2005) already discuss the problem of generating an appropriate initial design in the presence of unknown constraints. They suggest to extract failure regions adaptively by constructing spatial cylinders from the observations. The method is based on the assumption that the non-failure region is convex and connected which can often be assumed in engineering processes. Assuming convexity is given, a pair of one failure and one non-failure point provides the information that everything from the viewpoint of the failure point lying behind the non-failure point belongs to the failure region and can therefore be extracted. According to Figure 5.7 (left hand side), the boundary of the convex feasible area has to lie somewhere between these two points.
If in the $p$ dimensional parameter space $p$ non-failure points and one failure point are observed, the complete convex hull of the vectors spanned by the failure point with each of the non-failure points has to be part of the failure region (see right hand side of Figure 5.7). With the help of an initial space-filling design the failure region of the parameter space then is excluded constructing a convex hull for each combination of one failure point and $p$ non-failure points. Figure 5.8 gives an example. Henkenjohann et al. then augment the design with new points that are not lying inside one of the

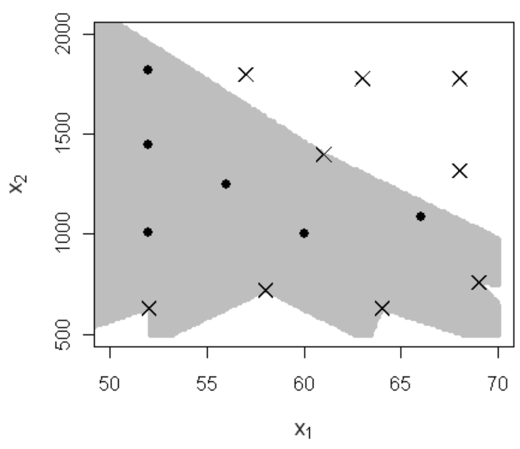


Figure 5.8: Excluded failure region (white) from a parameter space using the convex hull strategy

excluded convex hulls and examine the feasible area adaptively until they reach a desired amount of non-failure points. In Henkenjohann et al. (2005) it is shown that the algorithm still works for slight violations of the convexity assumption. This convex hull exclusion strategy is an excellent strategy to construct a good initial space-filling design for global optimization procedures. It gives a good basis to fit surrogate models. However, the brute force implementation by Henkenjohann is very inefficient and slow for more than three objectives and large space-filling designs.

Therefore, we extend this algorithm using the double description method for polyhedral convex cones (PCC) introduced by Fukuda and Prodon (1996). The core of the double description method are so-called double description pairs (DD pairs). A pair $(A, R)$ of real matrices $A$ and $R$ is called a DD pair if the following relationship holds

$$Ax \geq 0 \text{ iffx} = \text{R}\lambda \text{ for some } \lambda \geq 0. \tag{5.1}$$

The same intersection of a cube for example can hence either be defined by hyperplanes as illustrated on the left hand side of Figure 5.9 or given by vectors as shown on the right hand side of the figure. A DD pair also exists for polyhedral convex cones. A subset $P \in \mathbb{R}^p$ is called a polyhedral cone if there exists any $A \in \mathbb{R}^{n \times p}$ that satisfies the equation

$$P = \{x \in \mathbb{R}^p : Ax \geq 0\} =: P(A). \tag{5.2}$$

In Fukuda it is shown that for a polyhedral cone $P$ and the corresponding matrix $A$

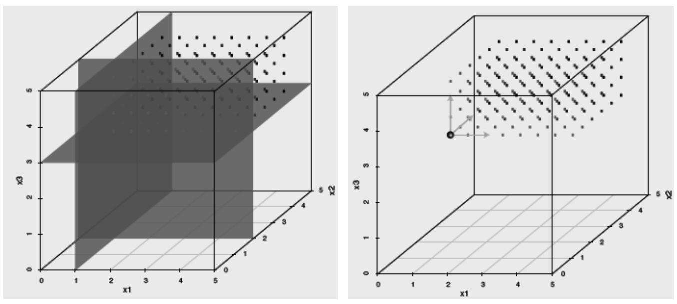


Figure 5.9: Double description pair of a cube intersection: given by the hyperplanes $x_1 \leq 1, x_2 \leq 2$, and $x_3 \leq 3$ (left) and given by the vectors $(1,2,3), (1,0,0), (0,1,0), (0,0,1)$ (right)

that fulfills $P = P(A)$, there exists a matrix $R \in \mathbb{R}^{p \times m}$ such that $(A, R)$ is a DD pair and it is

$$\begin{aligned} P &= \{x \in \mathbb{R}^p : Ax \geq 0\} \\ &= \{x \in \mathbb{R}^p : x = R\lambda \text{ for some } \lambda \geq 0\}. \end{aligned} \tag{5.3}$$

The matrix $A$ then is called the representation matrix and $R$ the generating matrix of the polyhedral cone $P$. With the help of an algorithm presented in Fukuda et al. one can very fast transform between the matrix $A$ and $R$. Thereby, the transformation algorithm considers only extreme rays when determining the representation matrix of a polyhedral cone. In our application, we can produce the generating matrix $R$ immediately using all failure and a non-failure point. Then we determine the corresponding representation matrix $A$ with the transformation algorithm. The equations resulting from $Ax \geq 0$ enable to check very easily whether a new point lies inside the cone.

Thus, the double description method provides a fast solution to generate a polyhedral convex cone (PCC) from a set of non-failure points and one failure point as the tip of the cone. The decision whether a point lies inside this cone or not is very easy.

### 5.2.2 Handling of missing values during the updating process

When performing statistical analysis with missing values, it must be distinguished between data that is missing at random and missing values, where a relationship between the data and its ‘missingness’ is present. When data is missing at random it can be ignored and interpolated. However, when a design point or an updating point fails, the data is usually lying in an infeasible area. If such a failure point is ignored and the sequential optimization process is proceeded without new information from the updating point, the surrogate model will remain unchanged in the region of the missing value. The criterion used to find new updating points will remain unchanged, too. The sequential optimization will then stall in this region. This problem can eventually be solved choosing a random point every time an updating point fails, but it does not prevent the process from recurring to the infeasible region several times and hence may be very inefficient. Forrester et al. (2006) suggest to use the information of the missing value about the infeasible region. They replace a missing value with imputed values before training the surrogate model again. The substitution of the missing value thereby is not important to have a certain value. The imputation is only used to set a value representing the missing observation, such that the prediction error of the refined surrogate model is reduced in the region around the missing value and the optimization is diverted towards the feasible region. Forrester et al. suggest to substitute the missing value with the upper bound of the prediction from the surrogate model at this point $y_{imputed} = \hat{y}(x) + s^2(x)$ for a minimization problem. They state that the asymptotic convergence of the maximum expected improvement criterion is not affected and global convergence is guaranteed in the feasible region. We adapt the imputation method to the target-value optimization case with the help of desirabilities. Suppose $x_{NA}$ is the candidate where the experiment fails and the observation $y(x_{NA})$ is missing. We then substitute $y(x_{NA})$ with

$$y_{imputed}(x_{NA}) = \begin{cases} \hat{y}(x_{NA}) + s^2(x_{NA}) & \text{, if } \hat{y}(x_{NA}) > T \\ \hat{y}(x_{NA}) - s^2(x_{NA}) & \text{, if } \hat{y}(x_{NA}) < T \end{cases}, \qquad (5.4)$$

where $T$ is the target value of objective $y$. The differentiation in the formula affects that adding or subtracting the prediction error diverts the imputed value from the target value. In case of a minimization or maximization problem the imputation of Forrester et al. is to be used. This imputation ensures that the process will not recur to the region of the failure point unless the region is so good, that even the imputed value is better than the already found local optimum.

If the recurrence of the region around the missing value shall be excluded definitively, the missing value can be substituted with the upper or lower specification limit of the corresponding desirability function. The desirability of the imputed $y_{imputed}$ then becomes zero with a prediction error equal to zero. Hence, the region will be avoided in the next optimization cycles.

The optimization procedure can be improved additionally if the imputation of a missing value is combined with the PCC strategy from the previous section. If an updating point fails, we use all present observations to exclude the region behind the failure point from the candidate grid. Then we substitute the missing value by the corresponding imputed value in order to prevent the algorithm from stalling on the boundary of the failure region. Finally, we refine the surrogate model with all new updating points including the imputed one. The new surrogate model now has worse values with a smaller prediction error near the failure point and no predictions are determined for the excluded candidates. In the next optimization cycle the updating points are chosen among the remaining candidate grid, where the known failure region is excluded. We also did some small simulations for this strategy showing that the global optimum is usually still found. Problems can arise when the predictions from the surrogate model are very bad and the global optimum is lying exactly at the boundary of the feasible region. In such cases the strategy might impute the missing value that hard, that the algorithm does not recur to the region between a known feasible observation and the failure point where the global optimum in fact is. To find such an optimum the only strategy that works is to go stepwise backwards from the failure point until a non-failure point is reached. But this is very inefficient and still does not guarantee to find the true optimum, since the question is which direction 'going backwards' is the most reasonable if the dimension of the parameter space is large. However, if an optimum is assumed to lie close to the boundary of the feasible area one definitely should use the imputation defined in Equation 5.4 instead of imputing with one of the specification limits of the desirability.

# 6 Case studies: application to sheet metal spinning and necking-in

In order to check the applicability of mtEGO in industrial practice, we have run two case studies in engineering processes. First, it is used to optimize a pot produced with the sheet metal spinning process. This optimization problem has been solved already in the literature as a maximization/minimization problem, see for example Henkenjohann et al. (2005). But further optimizations with other and more influencing parameters led to problems as explained in Section 1.1. Now, a target-value optimization using mtEGO is performed. As a second case study, mtEGO is used to optimize necking-in of tubes using the spinning process. Both optimizations show encouraging results.

## 6.1 Optimization of a pot produced with the sheet metal spinning process

Sheet metal spinning is one of the oldest chipless forming processes (König and Klocke, 1995, Ch.3.4). The process transforms flat sheet metal blanks to complex rotationally symmetric hollow shapes. Due to low costs of manufacture sheet metal spinning is mostly used for low and medium volumes of production or for prototyping. Examples for industrial applications are centrifuges, funnels, tank ends, wheel rims as well as workpieces in the aerospace industry. Figure 6.1a shows the principles of the process. The circular blank is clumped centrically against the mandrel by means of the tail stock. The mandrel with the mounted workpiece starts rotating. Following a beforehand determined path a roller tool progressively forms the rotating workpiece pressing it against the contour of the mandrel in multiple passes. The roller tool is operated through a CNC (Computer Numerical Control) unit. Figure 6.1b shows a possible path of the roller tool for a process with 10 stages. For certain parameter settings

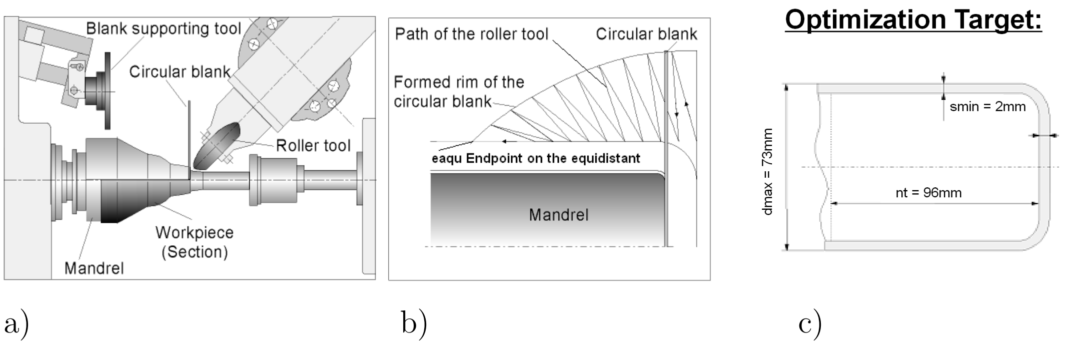


Figure 6.1: Principle of the sheet metal spinning process (from Henkenjohann et al., 2005)

the process fails, i.e. wrinkling or cracking of the workpiece. To stabilize the forming process a blank supporting tool is often used.

In this project, research is concentrated on three quality characteristics. The wall thickness ($smin$), the cup depth ($nt$) and the radius of the cup ($dmax$). Thereby, the overall minimum wall thickness is measured, since the degree of sheet thinning is of interest. The radius is measured in the maximum point to get to know the strength of the springback or a bulge. The depth of the cup is measured in the lowest point of the flange, that often forms tails. Metal spinning is influenced by a large number of interacting parameters. From screening experiments we know that our three quality characteristics are influenced significantly by the following six parameters (other machine parameters that were not significant have been fixed at a certain value):

- The curvature of the passes of the roller tool in forward direction $k_{hin}$ and backward direction $k_{ruck}$ in mm.
- The working radius of the roller tool $rdw$ in mm.
- The end point of the last pass of the roller tool on the equidistant to the mandrel $eaqu$ in % (cf. Figure 6.1b).
- The distribution of the passes on the formed rim of the blank $vrbk$, where 1 indicates equally distributed passes (cf. Figure 6.1b).
- The ratio $f_s$, which is feed rate $f$ divided by the spindel speed $s$.

| ***quality characteristics*** | *target value* |
|---|---|
| $nt$ | 96mm |
| $dmax$ | 73mm |
| $smin$ | 2mm |
| ***influencing parameters*** | *5 considered factor levels* |
| $k_{hin}$ | -2, -1, 0, 1, 2 |
| $k_{ruck}$ | -2, -1, 0, 1, 2 |
| $rdw$ | 2, 5, 10, 15, 20 |
| $eaqu$ | 30, 38, 46, 54, 62 |
| $vrbk$ | 0.92, 0.95, 0.98, 1.01, 1.04 |
| $f_s$ | 1.5, 2.5, 3.5, 4.5, 5.5 |
| ***fixed machine parameters*** | *fixed at value* |
| $vaqu$ | 1.01 |
| number of passes | 14 |
| $f$ | 3000 |
| ***blank*** | |
| blank diameter | 180mm |
| blank thickness | 2mm |
| diameter of mandrel | 69mm |
| material | Al99.5 w7 (soft aluminium) |

Table 6.1: Target values and parameter space of the sheet metal spinning optimization

The ratio $f_s$ was planned to be set up in the machine holding the feed rate $f$ at the constant value of 3000 revolutions per minute and choosing a corresponding spindel speed $s$ to get a given ratio $f_s$. The ratio $f_s$ was used instead of the spindel speed directly in compliance with the requirements of the engineers.

Table 6.1 summarizes the whole experimental setup for the optimization procedure. It gives the target values for the quality characteristics, the factor levels for each influencing parameter, the blank values and the fixed machine parameters. The optimization deals with six influencing parameters and five factor levels for each parameter, resulting in a grid of 15625 points representing all settings in the six-dimensional parameter space. Although we are using soft aluminium for these experiments that is very flexible to be formed, we expect to touch the forming limits with the chosen parameter

| Characteristic | $LSL$ | $target$ | $USL$ | $l$ | $r$ |
|---|---|---|---|---|---|
| nt | 86 | 96 | 111 | 0.15 | 0.1 |
| dmax | 71 | 73 | 80 | 1 | 0.1 |
| smin | 1.3 | 2 | 2.1 | 0.3 | 0.1 |

Table 6.2: Specifications of the Derringer Suich desirability functions in the spinning process optimization

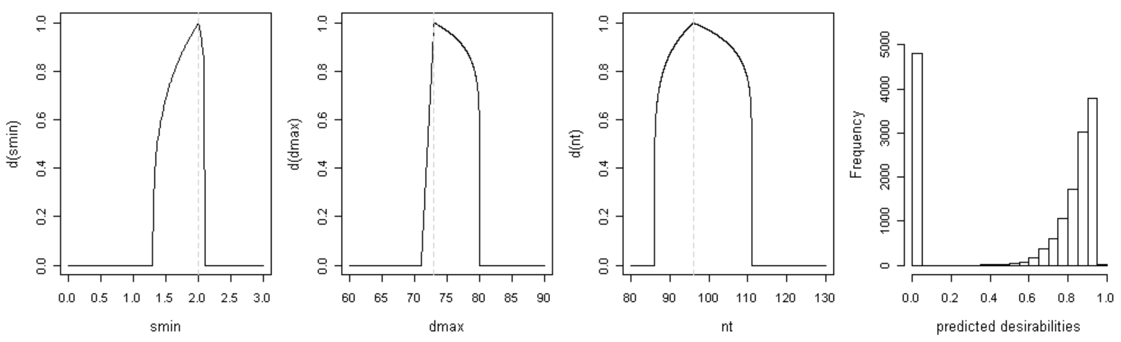


Figure 6.2: Desirability functions for the sheet metal spinning process

space and have failure parts within the optimization procedure.

With the help of expert knowledge from the engineers, desirability functions are specified for the three objectives $dmax, nt$ and $smin$. Particularly, we give attention to the case of over-optimization of the sheet thickness mentioned in the motivating example and, hence, specify asymmetric Derringer-Suich desirability functions. Table 6.2 and Figure 6.2 show the specifications of the desirability functions. The characteristic thickness $smin$ is penalized very hard if it exceeds the target value of 2mm, and less harder if it falls below. The opposite applies for the diameter $dmax$. Here, values below the target are penalized very hard and exceeding the target is less worse. However, the diameter should physically not fall below the target, because of the springback of the cup walls. The target of 73mm is hardly reachable and theoretically not possible to be under-run. We anyway use a two-sided desirability function for this characteristic just to penalize the case of under-running correctly if it happens in spite of the theory. For the third characteristic $nt$, deviations to both sides are penalized almost equally, since the flange of the cup is usually cut after production and hence only cups with much too small diameters are useless, and those with much too large diameters will be too thin or will have bad stiffness.

According to Jones et al. (1998) it is advisable to use 10 experiments per influencing parameter for the initial design. We had a limit of approximately 50 experiments in total for the whole optimization and decided to start with a very small design and keep as many experiments for the updating steps as possible. The optimization of the spinning process is therefore initiated with only 15 design points. The initial design is chosen with the uniform-coverage criterion to cover the parameter space space-fillingly (cf. Section 2.2.3). The first rows of Table 6.3 show the design points of the initial design with their resulting observations and desirability index $DI$. The best observation already has a $DI$ of 0.9295 and also all other values of $DI$ are either 0 or very large ($> 0.7135$). Unfortunately, the majority of the desirability index values lie between 0.8 and 1.0 (confer to the right hand side of Figure 6.2 showing the distribution of $DI$ values for the surrogate model fitted to the initial design). This happens due to too liberal specifications of the desirabilities, particularly for the depth $nt$.

After running the initial design the selection of appropriate surrogate models for the three objectives is done. Kriging models with differently complex deterministic trend components (only intercept or linear) and isotropic as well as anisotropic power exponential covariance functions with smoothness 1 or 2 are fitted. Since the influencing parameters are very differently scaled, we scale all of them to the interval $[0, 1]$ first before fitting the models. This should be remembered during an interpretation of the estimates of the effects. A detailed summary on the fitted models and the model selection can be found in Appendix A.3. The model with the best compromise of lowest BIC, MSCVE and number of estimated parameters is chosen. Tables 6.4 and 6.5 show the estimates for the parameters of the chosen surrogate models during the progress of the optimization. The first column of the table contains the initial surrogate models. For all characteristics $dmax$, $nt$ and $smin$ the linear deterministic trend function $\mu + k_{hin} + rdw + eaqu + vrbk + k_{ruck} + f_s$ was fitted and the covariance function is an isotropic power exponential covariance function with smoothness $s = 2$. The progress of the surrogate models over the optimization cycles shows that the choice of the model was sufficient, since it does not change a lot with increasing number of included observations.

Having the initial design, the surrogate models and the desirability function we start the optimization procedure. The computing time per step should be less than one hour. For six influencing parameters and three objectives this indicates to use only three confidence levels. Due to the very small initial design and the resulting assumption

| id | cycle | $k_{hin}$ | $rdw$ | $eaqu$ | $vrbk$ | $k_{ruck}$ | $f_s$ | $f_{scorr}$ | failure | $smin$ | $dmax$ | $nt$ | $DI$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Initial | 1 | 2 | 38 | 1.01 | 0 | 3.5 | 3.5 | 0 | 1.29 | 76.2 | 109.14 | 0.0000 |
| 2 | Initial | 1 | 15 | 38 | 1.01 | 0 | 4.5 | 4.5 | 0 | 1.85 | 77.0 | 89.92 | 0.9056 |
| 3 | Initial | 1 | 15 | 46 | 1.04 | 0 | 1.5 | 2.7 | 0 | 1.68 | 76.4 | 98.95 | 0.9134 |
| 4 | Initial | -2 | 5 | 46 | 0.98 | 2 | 3.5 | 3.5 | 0 | 1.38 | 76.7 | 110.46 | 0.7027 |
| 5 | Initial | 2 | 5 | 46 | 0.95 | -1 | 3.5 | 3.5 | 0 | 1.51 | 76.1 | 97.65 | 0.8661 |
| 6 | Initial | -2 | 20 | 46 | 0.98 | 2 | 3.5 | 3.5 | 0 | 1.80 | 77.6 | 93.43 | 0.9193 |
| 7 | Initial | 1 | 20 | 46 | 0.95 | -1 | 3.5 | 3.5 | 0 | 1.94 | 76.8 | 88.68 | 0.9040 |
| 8 | Initial | 1 | 10 | 46 | 0.92 | 2 | 4.5 | 4.5 | 0 | 1.88 | 76.4 | 89.10 | 0.9053 |
| 9 | Initial | 1 | 10 | 46 | 0.98 | 2 | 1.5 | 2.7 | 0 | 1.73 | 76.0 | 94.10 | 0.9250 |
| 10 | Initial | -1 | 10 | 46 | 0.92 | -1 | 4.5 | 4.5 | 0 | 1.61 | 76.9 | 98.50 | 0.8917 |
| 11 | Initial | -1 | 10 | 62 | 0.95 | 0 | 1.5 | 2.7 | 0 | 1.40 | 76.2 | 107.61 | 0.7676 |
| 12 | Initial | -1 | 10 | 46 | 1.04 | -1 | 1.5 | 2.7 | 0 | 1.26 | 78.3 | 118.14 | 0.0000 |
| 13 | Initial | 1 | 10 | 62 | 1.01 | 0 | 4.5 | 4.5 | 0 | 1.36 | 78.3 | 107.08 | 0.7135 |
| 14 | Initial | -2 | 10 | 46 | 1.01 | -1 | 4.5 | 4.5 | 0 | 1.30 | 78.1 | 108.69 | 0.0000 |
| 15 | Initial | -1 | 10 | 30 | 0.95 | 0 | 1.5 | 2.7 | 0 | 1.74 | 76.4 | 97.88 | **0.9295** |
| 16 | Step 1 | 2 | 15 | 46 | 0.92 | 0 | 4.5 | 4.5 | 0 | 1.93 | 76.8 | 86.41 | 0.8217 |
| 17 | Step 1 | 1 | 15 | 30 | 0.92 | 2 | 1.5 | 2.7 | 0 | 2.06 | 75.9 | 86.33 | 0.8034 |
| 18 | Step 1 | -2 | 15 | 38 | 0.92 | 0 | 1.5 | 2.7 | 0 | 1.76 | 76.8 | 96.82 | 0.9324 |
| 19 | Step 1 | 0 | 15 | 54 | 1.01 | -1 | 1.5 | 2.7 | 0 | 1.49 | 77.9 | 104.43 | 0.8203 |
| 20 | Step 1 | 0 | 15 | 62 | 0.95 | 2 | 1.5 | 2.7 | 0 | 1.73 | 76.8 | 97.07 | 0.9256 |
| 21 | Step 1 | 0 | 20 | 30 | 0.98 | 1 | 1.5 | 2.7 | 0 | 1.93 | 76.4 | 88.43 | 0.9017 |
| 22 | Step 1 | -1 | 20 | 38 | 1.01 | 0 | 1.5 | 2.7 | 0 | 1.86 | 76.9 | 94.10 | **0.9418** |
| 23 | Step 1 | -2 | 10 | 30 | 0.92 | 2 | 1.5 | 2.7 | 0 | 1.73 | 76.3 | 98.59 | 0.9265 |
| 24 | Step 1 | 2 | 10 | 30 | 0.92 | 0 | 3.5 | 3.5 | 0 | 1.95 | 76.0 | 86.86 | 0.8618 |
| 25 | Step 1 | 2 | 2 | 46 | 1.04 | 0 | 1.5 | 2.7 | 1 | 1.31 | 77.0 | 115.26 | 0.0000 |
| 26 | Step 2 | -1 | 10 | 30 | 0.92 | 2 | 1.5 | 2.7 | 0 | 1.77 | 76.1 | 95.34 | 0.9392 |
| 27 | Step 2 | 1 | 10 | 38 | 0.98 | 2 | 1.5 | 2.7 | 0 | 1.83 | 76.0 | 93.24 | 0.9393 |
| 28 | Step 2 | 2 | 10 | 38 | 0.95 | 0 | 3.5 | 3.5 | 0 | 1.82 | 76.1 | 89.88 | 0.9080 |
| 29 | Step 2 | 2 | 20 | 30 | 0.92 | -1 | 1.5 | 2.7 | 1 | 2.25 | 76.5 | 77.58 | 0.0000 |
| 30 | Step 2 | -2 | 20 | 30 | 1.04 | 2 | 1.5 | 2.7 | 0 | 1.74 | 77.5 | 94.46 | 0.9147 |
| 31 | Step 2 | 1 | 15 | 38 | 0.92 | 1 | 1.5 | 2.7 | 0 | 1.99 | 76.0 | 88.20 | 0.9086 |
| 32 | Step 2 | 1 | 5 | 38 | 0.92 | 2 | 2.5 | 2.7 | 0 | 1.70 | 76.4 | 92.75 | 0.9068 |
| 33 | Step 3 | 1 | 10 | 30 | 0.98 | 1 | 1.5 | 2.7 | 0 | 1.90 | 76.0 | 92.00 | 0.9421 |
| 34 | Step 3 | 2 | 10 | 54 | 0.95 | 2 | 1.5 | 2.7 | 0 | 1.83 | 75.9 | 91.80 | 0.9297 |
| 35 | Step 3 | 1 | 20 | 54 | 0.92 | 2 | 2.5 | 2.7 | 0 | 2.06 | 76.6 | 88.20 | 0.8778 |
| 36 | Step 3 | -1 | 20 | 62 | 0.92 | 2 | 1.5 | 2.7 | 0 | 1.80 | 77.0 | 94.40 | 0.9318 |
| 37 | Step 3 | 1 | 20 | 54 | 0.92 | -1 | 2.5 | 2.7 | 0 | 1.96 | 76.5 | 91.60 | **0.9437** |
| 38 | Step 3 | 1 | 15 | 38 | 1.01 | 2 | 1.5 | 2.7 | 0 | 1.89 | 76.6 | 90.40 | 0.9211 |
| 39 | Step 3 | 1 | 5 | 30 | 1.04 | 2 | 1.5 | 2.7 | 0 | 1.44 | 76.0 | 99.80 | 0.8275 |
| 40 | Step 4 | -2 | 15 | 30 | 0.92 | 2 | 1.5 |  | 0 | 1.73 | 75.9 | 98.08 | 0.9310 |
| 41 | Step 4 | 0 | 15 | 62 | 0.92 | -1 | 1.5 |  | 0 | 1.67 | 76.2 | 103.82 | 0.8970 |
| 42 | Step 4 | 2 | 15 | 54 | 0.92 | 2 | 1.5 |  | 0 | 1.95 | 75.6 | 90.24 | 0.9363 |
| 43 | Step 4 | 2 | 5 | 30 | 0.92 | 1 | 2.7 |  | 0 | 1.89 | 75.6 | 90.82 | 0.9333 |
| 44 | Step 4 | 2 | 10 | 46 | 0.98 | 2 | 3.5 |  | 0 | 1.86 | 76.4 | 90.50 | 0.9191 |
| 45 | Step 4 | -1 | 15 | 38 | 1.01 | 0 | 1.5 |  | 0 | 1.54 | 77.5 | 104.64 | 0.8437 |
| 46 | Step 4 | 1 | 5 | 30 | 0.92 | -1 | 1.5 |  | 0 | 1.57 | 75.2 | 106.95 | 0.8594 |
| 47 | Verif. | 2 | 20 | 62 | 0.92 | -2 | 1.5 |  | 0 | 1.86 | 75.9 | 94.23 | **0.9513** |
| 48 | Verif. 2 | 0 | 15 | 30 | 0.92 | -1 | 1.5 |  | 0 | 1.96 | 75.7 | 94.67 | 0.9712 |
| 49 | Verif. 2 | 0 | 20 | 54 | 0.92 | 0 | 1.5 |  | 0 | 1.89 | 76.1 | 95.69 | 0.9626 |
| 50 | Verif. 2 | 1 | 20 | 62 | 0.92 | -1 | 1.5 |  | 0 | 1.86 | 76.1 | 96.26 | 0.9585 |
| 51 | Verif. 2 | 2 | 10 | 30 | 0.92 | -1 | 1.5 |  | 0 | 1.95 | 75.4 | 94.69 | **0.9720** |

Table 6.3: Experiments and observations with the resulting desirability index $DI$ for the sheet metal spinning optimization

| model component | parameter | estimator initial design | estimator optimization cycle 1 | estimator optimization cycle 2 | estimator optimization cycle 3 |
|---|---|---|---|---|---|
| *dmax* | | | | | |
| deterministic trend component | $\mu$ | 75.954 | 76.337 | 76.648 | 76.550 |
| | $k_{hin}$ | -1.044 | -0.783 | -0.818 | -0.815 |
| | $rdw$ | 0.622 | 0.514 | 0.675 | 0.746 |
| | $eaqu$ | 0.750 | 0.681 | -0.030 | -0.031 |
| | $vrbk$ | 1.289 | 0.950 | 0.810 | 0.762 |
| | $k_{ruck}$ | -0.601 | -0.544 | -0.523 | -0.452 |
| | $f_s$ | 1.082 | 0.687 | 0.792 | 0.834 |
| power exponential covariance function isotropic (s=2) | $\theta$ | 0 | 1.0E-18 | 0.6153 | 0.5627 |
| | $\sigma_Z^2$ | 0.2959 | 0.2189 | 0.2972 | 0.2309 |
| *nt* | | | | | |
| deterministic trend component | $\mu$ | 110.30 | 122.70 | 115.97 | 114.74 |
| | $k_{hin}$ | -12.746 | -13.322 | -15.008 | -12.450 |
| | $rdw$ | -19.776 | -18.593 | -23.860 | -19.610 |
| | $eaqu$ | 11.080 | 8.711 | 1.488 | 1.258 |
| | $vrbk$ | 12.340 | 11.356 | 12.378 | 9.694 |
| | $k_{ruck}$ | -7.271 | -5.784 | -4.231 | -6.621 |
| | $f_s$ | -3.753 | -5.943 | -2.551 | -1.888 |
| power exponential covariance function isotropic (s=2) | $\theta$ | 0 | 1.8104 | 1.3472 | 1.0080 |
| | $\sigma_Z^2$ | 7.1140 | 165.47 | 64.188 | 30.138 |
| *smin* | | | | | |
| deterministic trend component | $\mu$ | 1.364 | 1.328 | 1.244 | 1.254 |
| | $k_{hin}$ | 0.261 | 0.250 | 0.274 | 0.290 |
| | $rdw$ | 0.611 | 0.564 | 0.601 | 0.592 |
| | $eaqu$ | -0.343 | -0.268 | -0.294 | -0.031 |
| | $vrbk$ | -0.301 | -0.253 | -0.312 | -0.324 |
| | $k_{ruck}$ | 0.212 | 0.185 | 0.168 | 0.115 |
| | $f_s$ | -0.004 | 0.023 | -0.009 | 0.023 |
| power exponential covariance function isotropic (s=2) | $\theta$ | 0 | 3.1E-17 | 0.853 | 0.8642 |
| | $\sigma_Z^2$ | 0.0022 | 0.0029 | 0.0156 | 0.0164 |

Table 6.4: Surrogate models during the optimization of the sheet metal spinning process

| model component | para-meter | estimator corr. model cycle 3 | estimator optimization cycle 4 | estimator verification phase 1 | estimator verification phase 2 |
|---|---|---|---|---|---|
| *dmax* | | | | | |
| deterministic trend component | $\mu$ | 76.321 | 75.893 | 75.878 | 75.841 |
| | $k_{hin}$ | -0.629 | -0.653 | -0.701 | -0.678 |
| | $rdw$ | 0.581 | 0.624 | 0.571 | 0.549 |
| | $eaqu$ | 0.596 | 0.592 | 0.535 | 0.548 |
| | $vrbk$ | 0.862 | 0.964 | 0.967 | 0.980 |
| | $k_{ruck}$ | -0.405 | -0.381 | -0.309 | -0.291 |
| | $f_s$ | 1.040 | 1.144 | 1.227 | 1.265 |
| power exponential covariance function isotropic (s=2) | $\theta$ | 0.4849 | 0.4644 | 0.4602 | 0.4724 |
| | $\sigma_Z^2$ | 0.1886 | 0.1759 | 0.1748 | 0.1659 |
| *nt* | | | | | |
| deterministic trend component | $\mu$ | 110.16 | 118.50 | 118.33 | 118.88 |
| | $k_{hin}$ | -11.532 | -12.563 | -12.758 | -13.054 |
| | $rdw$ | -19.192 | -20.521 | -20.706 | -21.012 |
| | $eaqu$ | 12.538 | 10.852 | 10.428 | 10.517 |
| | $vrbk$ | 10.446 | 8.816 | 8.769 | 8.814 |
| | $k_{ruck}$ | -5.939 | -7.771 | -7.334 | -7.548 |
| | $f_s$ | -6.494 | -11.310 | -10.951 | -10.938 |
| power exponential covariance function isotropic (s=2) | $\theta$ | 0.8443 | 0.7558 | 0.7451 | 0.7766 |
| | $\sigma_Z^2$ | 14.423 | 12.340 | 11.6888 | 12.056 |
| *smin* | | | | | |
| deterministic trend component | $\mu$ | 1.347 | 1.258 | 1.257 | 1.282 |
| | $k_{hin}$ | 0.279 | 0.305 | 0.301 | 0.306 |
| | $rdw$ | 0.563 | 0.571 | 0.568 | 0.570 |
| | $eaqu$ | -0.264 | -0.248 | -0.253 | -0.267 |
| | $vrbk$ | -0.273 | -0.263 | -0.262 | -0.268 |
| | $k_{ruck}$ | 0.141 | 0.157 | 0.162 | 0.150 |
| | $f_s$ | 0.023 | 0.116 | 0.120 | 0.097 |
| power exponential covariance function isotropic (s=2) | $\theta$ | 0.4240 | 0.4276 | 0.4156 | 0.4008 |
| | $\sigma_Z^2$ | 0.0034 | 0.0040 | 0.0039 | 0.0039 |

Table 6.5: Surrogate models during the optimization of the sheet metal spinning process

| | $k_{hin}$ | $rdw$ | $eaqu$ | $vrbk$ | $k_{ruck}$ | $f_s$ ($f_{scorr}$) | $smin$ | $dmax$ | $nt$ | $DI$ |
|---|---|---|---|---|---|---|---|---|---|---|
| optimum after initial design (points 1-15) | | | | | | | | | | |
| observed | -1 | 10 | 30 | 0.95 | 0 | 1.5 (2.7) | 1.74 | 76.4 | 97.88 | 0.9295 |
| predicted | -2 | 10 | 30 | 0.92 | 2 | 1.5 | 1.58 | 75.63 | 94.25 | 0.9516 |
| optimum after optimization cycle 1 (points 1-25) | | | | | | | | | | |
| observed | -1 | 20 | 38 | 1.01 | 0 | 1.5 (2.7) | 1.86 | 76.9 | 94.10 | 0.9418 |
| predicted | -1 | 10 | 30 | 0.92 | 2 | 1.5 | 1.83 | 75.83 | 95.41 | 0.9523 |
| optimum after optimization cycle 2 (points 1-32) | | | | | | | | | | |
| observed | -1 | 20 | 38 | 1.01 | 0 | 1.5 (2.7) | 1.86 | 76.9 | 94.10 | 0.9418 |
| predicted | -1 | 20 | 62 | 0.92 | 2 | 1.5 | 1.94 | 76.85 | 95.76 | 0.9641 |
| optimum after optimization cycle 3 (points 1-39) | | | | | | | | | | |
| observed | 1 | 20 | 54 | 0.92 | -1 | 2.5 (2.7) | 1.96 | 76.5 | 91.60 | 0.9437 |
| predicted | -2 | 15 | 30 | 0.92 | 2 | 1.5 | 1.88 | 76.07 | 95.86 | 0.9619 |
| optimum after optimization cycle 4 (points 1-46) | | | | | | | | | | |
| observed | 1 | 20 | 54 | 0.92 | -1 | 2.5 (2.7) | 1.96 | 76.5 | 91.60 | 0.9437 |
| predicted | 2 | 20 | 62 | 0.92 | -2 | 1.5 | 1.89 | 76.4 | 96.87 | 0.9618 |
| optimum after verification (points 1-47) | | | | | | | | | | |
| observed | 2 | 20 | 62 | 0.92 | -2 | 1.5 | 1.86 | 75.9 | 94.23 | 0.9513 |
| predicted | 2 | 10 | 30 | 0.92 | -1 | 1.5 | 1.84 | 75.33 | 96.02 | 0.9619 |
| optimum after verification phase 2 (points 1-51) | | | | | | | | | | |
| observed | 2 | 10 | 30 | 0.92 | -1 | 1.5 | 1.95 | 75.4 | 94.69 | 0.9720 |
| predicted | 2 | 10 | 30 | 0.92 | -1 | 1.5 | 1.95 | 75.4 | 94.69 | 0.9720 |

Table 6.6: Results of the sequential optimization for the sheet metal spinning process.

that the first model is probably very uncertain, the search should be very global and the $\alpha$ confidence levels are parameterized with small values $\alpha \in \{0.0001, 0.001, 0.01\}$.

Table 6.6 summarizes the progress of the optimization procedure. After each cycle of the procedure the currently observed and currently predicted optimal parameter setting are given with their observations and predictions for $dmax, nt$, and $smin$, and the resulting $DI$. The predicted optimum thereby is that point with the best $DI$ among the predictions for the whole parameter space based on the surrogate model fitted to all currently observed values. After the initial cycle the observed optimum reaches $DI = 0.9295$. The predictions indicate a possible optimum in another setting with $DI = 0.9516$.

In the first optimization cycle, mtEGO determines a set of 98 candidates, that are clustered in nine groups, i.e. nine updating points are chosen. The corresponding cluster dendrograms and scree plots for each optimization step that were used to decide on the number of updating points can be found in Appendix A.2. In addition to the 9 updating points, the predicted optimum is added as updating point. The corresponding observations in Table 6.3 show one failure part. To include this information for the optimization procedure, we substitute the missing observations for $dmax, nt$ and $smin$ of this parameter setting with the predictions of the previous surrogate model and penalize them with their standard prediction error as explained in Section 5.2.2 to avoid that the algorithm searches in this area again. This constructed penalized observation is included when refining the surrogate model. The observations also show that the parameter setting with the previously predicted optimum is actually not better than the previously observed optimum. The fitted surrogate model was wrong there. However, among the ten updating points actually a better parameter setting than the previously observed optimum is found with $DI = 0.9418$ for a completely different parameter setting. During the second optimization cycle seven updating points (including the predicted optimal setting from the first optimization cycle) are chosen from a set of 84 candidates. Again one failure part occurred and was substituted by penalized predictions. The parameter setting with optimal predictions also was again worse than predicted and also no other updating point had a better value for $DI$. The observed optimum stays the same as in the first optimization cycle, but the predictions still indicate a possible improvement of $DI$. During the third optimization cycle again seven design points are chosen from a set of 109 candidates. Although not being the predicted optimum, the $DI$ can be improved in this step. We find a new optimum among the observations with $DI = 0.9437$. Now, the refined surrogate model switches and predicts the optimum in another area of the parameter space. We decide to do a fourth optimization cycle.

Unfortunately, between the third and the fourth cycle the engineers realized that the influencing parameter $f_s$ has been set up wrong from beginning. Instead of setting $f_s$ to the levels 1.5, 2.5, 3.5, 4.5 and 5.5, all levels below 2.7 are set to the fixed value 2.7, all levels above are set correctly. The levels 1.5 and 2.5 are not set up differently contrary to what the design says. Hence, every time when mtEGO suggests to take the smallest possible $f_s$ to improve the results, $f_s$ is indeed larger in the experiments. This mistake leads to a wrong model and is corrected for the next cycle. We determine the correct set up $f_s$ for all conducted experiments (i.e. column $f_{scorr}$ in Table 6.3) and fit a new surrogate model. It is surprising that the model for the wrong $f_s$ does not

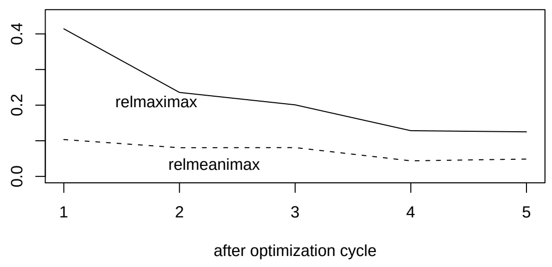


Figure 6.3: Progress of the stopping criterion in the spinning optimization

differ much from the corrected model (see Table 6.4 and 6.5). However, we assume that the $DI$ should improve now, since the models suggest to use a smaller $f_s$ to improve $DI$ from beginning on. Now we can actually realize it. In the fourth optimization cycle seven updating points (now with corrected $f_{scorr}$) from 39 candidates are added. All seven parameter settings yield in observations with good $DI$ values, but none of them improves the already observed optimum. On the one hand, the predictions still indicate a possibility of improvement for a completely other setting. On the other hand, the stopping criteria in Figure 6.3 finally saturates after the fourth optimization cycle, which indicates to stop the optimization now. Since we do not have many blanks left and the optimization is distorted by the engineers mistake, we decide to proceed with verification steps manually. We start with the evaluation of the current predicted optimal setting. Luckily, the predicted optimal setting is good and yields improved observations. $DI$ is now 0.9513 and the refined predictions indicate still the capability for better results. The next points are chosen according to the information of Table 6.7 containing the seven parameter settings with the best predicted $DI$ values in decreasing order based on the model refined with the one verification point. We summarize the information from these seven points as follows: To get a good part that meets the target values one should choose $f_s$ and $vrbk$ as small as possible (i.e. $f_s = 1.5$ and $vrbk = 0.92$ here). The parameter $k_{ruck}$ should be set up small or medium sized (i.e. -1 or 0). The upper values of $k_{hin}$ (i.e. 0,1,2) improve the $DI$. The parameters $eaqu$ and $rdw$ interact. Either $eaqu$ has to be small (30) and $rdw$ medium sized (10 or 15) to get good results. Or one should set both parameters the largest possible ($eaqu = 54$ or 62 and $rdw = 20$). We derive four new verification points from this information.

The results in the design table show that all four settings have better $DI$ values

| id of grid point | $k_{hin}$ | $rdw$ | $eaqu$ | $vrbk$ | $k_{ruck}$ | $f_s$ |
|---|---|---|---|---|---|---|
| 13756 | 2 | 10 | 30 | 0.92 | -1 | 1.5 |
| 12386 | 1 | 20 | 62 | 0.92 | 0 | 1.5 |
| 12381 | 1 | 20 | 62 | 0.92 | -1 | 1.5 |
| 9136 | 0 | 20 | 54 | 0.92 | 0 | 1.5 |
| 8171 | 0 | 15 | 30 | 0.95 | 2 | 1.5 |
| 8131 | 0 | 15 | 30 | 0.92 | -1 | 1.5 |
| 12256 | 1 | 20 | 54 | 0.92 | -1 | 1.5 |

Table 6.7: Parameter settings with seven best predicted $DI$ values for the spinning process in decreasing order

than the observed optimum and the predicted optimum has the best $DI$ of them. Since, the fitted surrogate model does not suggest a better setting now and the total of approximately 50 experiments is reached, we stop the optimization now. The optimal part meets the target values ($smin$=2, $dmax$=73 and $nt$=96) very good. The thickness $smin$ does not exceed the target as it did in the motivating example. Although the $DI$ only improves from 0.9295 to 0.9720 between the first best part and the overall best part, the quality changes a lot. The desirability functions are specified quite poorly, a more appropriate specification, i.e. more conservative specification of $nt$, would show a much larger improvement in quality. Maybe, a weighted desirability index would be appropriate in this case. One could weigh the depth that is very liberal to deviations less than the other two characteristics.

The case study shows the influence of the desirability functions. It is important to specify them very carefully. The study further shows, that a sequential optimization procedure has the advantage to deal with upcoming mistakes in the model. Of course, some additional points and maybe also additional cycles are needed, but the mistake is corrected and the optimum is finally found.

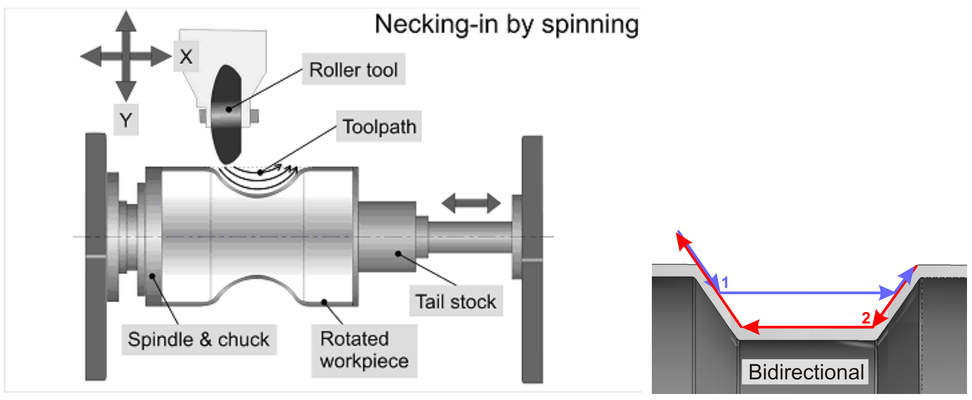


Figure 6.4: Principle of necking-in using the sheet metal spinning process

## 6.2 Optimization of the diameter reduction of a tube produced using necking-in with spinning

Subject of this case study is the optimization of the reduction of the diameter of tubes using the sheet metal spinning technology that has already been subject of the optimization in the previous section. The so-called necking-in with spinning can be used to manufacture tubular products with varying diameter over the part's length like driveshafts or gas containers. In our case we examine a tube with a simple diameter reduction. The principles of the process are shown on the left hand side of Figure 6.4. A tube with constant diameter is spanned between the tail stock and the spindel in the spinning machine. The mandrel starts rotating and a roller tool incrementally reduces the diameter of the desired region of the tube. The roller tool follows a path like the one shown on the right hand side of Figure 6.4. In contrast to the pot produced with the spinning process in the previous section, the diameter reduction of the tube is formed freely without a mandrel that determines the shape of the results. Typical failures of this process are cracks in the corners of the machined region. During our optimization however no failure occured. The quality of the machined region is measured by several characteristics. Figure 6.5 shows a CAD model and the cross section of the formed tube. The outer width $w_0$ comprises the whole machined region, whereas the inner width $w_i$ represents the length of the region where the reduction of the diameter is the achieved radial depth $t$. For a better comparison of experiments for different geometries usually the necking-in ratio $\beta$ is measured instead of the depth $t$. The necking-in ratio is defined as

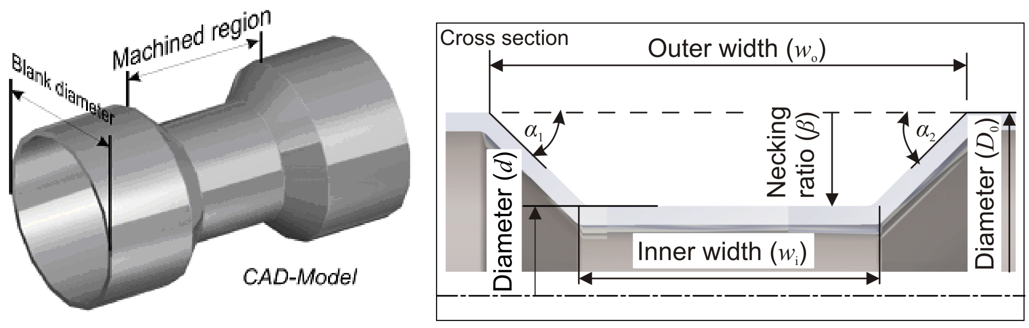


Figure 6.5: Quality characteristics to evaluate the diameter reduction

$$\begin{aligned}\beta &= \text{blank diameter } D_0 \text{ / necked-in diameter } d \\ &= D_0/(D_0 - 2 \cdot t).\end{aligned}$$

The taper regions are described by the angle $\alpha_1$ and the angle $\alpha_2$. Kunert et al. (2007) present an extensive examination of the influencing machine parameters like spindel speed, feed rate, different process strategy, number of passes etc. on those characteristics. Further, the deformation state, discoloration, disruptions, grooves and buckling are examined. They determine active factors on the quality characteristics with screening experiments and develop linear models to describe the basic functional relationship between machine parameters and quality characteristics. The results reveal that the geometry of the produced part does not fit with the targets of the geometry that shall be reached. If the CNC code is programmed to form a part with angles of 45 degrees and an inner width of 35mm, the processed part will have angles and an inner width that are different from the desired targets, depending on the chosen machine parameter settings.

### 6.2.1 Subject of the case study and experimental setup

In this case study the subject is a necked-in tube that shall be used in gear boxes, the chassis or as a drive shaft in automotive industry. The requirement of such a part is that it meets the geometry very precisely because of the restricted space inside the chassis or the gear box. The idea is to vary the input of the CNC programm such that a desired outer geometry is reached. We want to find the setting of angles, inner width and necking-in ratio that has to be coded to the CNC programm to get a part that actually reaches desired target values of angles, inner width and necking-in ratio. We do an optimization of four input geometry parameters $\alpha_1, \alpha_2, w_i$ and $\beta$ towards the

| ***quality characteristics*** | *target value* |
|---|---|
| descent angle $\alpha_{1obs}$ | 45 degrees |
| ascent angle $\alpha_{2obs}$ | 45 degrees |
| inner width $w_{iobs}$ | 35mm |
| necking-in ratio $\beta_{obs}$ | 1.5 |
| ***design parameters*** | *11 factor levels considered* |
| descent angle $\alpha_1$ | 40, 43.5, 47, 50.5, 54, 57.5, 61, 64.5, 68, 71.5, 75 |
| ascent angle $\alpha_2$ | 40, 43.5, 47, 50.5, 54, 57.5, 61, 64.5, 68, 71.5, 75 |
| inner width $w_i$ | 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38 |
| necking-in ratio $\beta$ | 1.4, 1.46, 1.52, 1.58, 1.64, 1.7, 1.76, 1.82, 1.88, 1.94, 2 |
| ***machine parameters*** | *fixed at value* |
| path strategy | see right hand side of Figure 6.4 |
| roller tool | symmetric roller tool with a radius of 8mm |
| number of passes | 7 |
| direction of the roller tool | bidirectional |
| pitch | 3 |
| tailstock | yes |
| ***blank tube*** | |
| blank diameter | 70mm |
| blank length | 250mm |
| blank wall thickness | 1.5mm |
| material | 27MnCrB |

Table 6.8: Target values and parameter space of the necking-in optimization

target values for the corresponding achieved geometry values. To avoid confusion we name the input parameters $\alpha_1, \alpha_2, w_i$ and $\beta$ and name the quality characteristics of the observed geometry $\alpha_{1obs}, \alpha_{2obs}, w_{iobs}$ and $\beta_{obs}$. The target values for the achieved outer geometry and the considered factor levels of the influencing parameters are given in Table 6.8. The machine parameters are fixed for the optimization and set to reliable settings for discoloration, buckling, disruption, deformation, chosen according to the results of the screening experiments of Kunert et al. The fixed settings for the machine parameters can also be found in Table 6.8. Additionally, it gives information about the blank to be formed.

### 6.2.2 Initial design and specified desirabilities

Altogether, this application is a target value optimization problem with four inputs and four objectives. The four dimensional parameter space of the inputs is represented by a point grid spanned over eleven factor levels per parameter, i.e. $11^4 = 14641$ points. For the whole optimization procedure a maximum number of 60 tubes was available. We started with a fourth of the blanks, i.e. 15 experiments as the initial design. The 15 point initial design was determined space-fillingly with the uniform coverage criterion from Section 2.2.3. The chosen initial design points and their experimental results are displayed in the first 15 rows of Table 6.9.

The last column of the table contains the joint desirability index of the four quality characteristics. The currently best desirability index for each step is highlighted with grey color. Figure 6.6 shows the specified desirability functions for each characteristic. All of them are two-sided Derringer-Suich desirability functions as defined in Equation 2.37 parameterized according to Table 6.10. The aim for all characteristics is to meet the geometry as precise as possible such that it fits into the constructive requirements. For the angles, as well as the inner width it makes no difference when the observed values are below or above the target value, only the absolute deviation gets problematic the larger it is. Therefore, the desirability functions are defined symmetrically, deviations in all directions are equally penalized. In contrast, it is worse if the necking-in ratio falls below the aimed target than if it exceeds the target. A necking-in ratio below the target means the diameter in the machined region is too large. This is rarely acceptable if the part shall fit in the provided constructive requirements. On the other hand, a slightly larger diameter yields a better stiffness of the part which is of course beneficial for every part that is exposed strong forces like gear parts or chassis. We defined an asymmetric desirability function, that penalizes necking-in ratios that fall below the target (i.e. necked-in diameter too large) faster and stronger than values exceeding it (i.e. the part tends not to fit anymore but yields more stiffness). All desirability functions were specified with the help of expert knowledge of engineers. The engineers provided us with all specification limits and curtosis of the functions. Their specified desirability functions were aggregated with the unweighted geometric mean to a joint desirability index $DI$. Table 6.9 shows that the best part among the initial experiments already reaches a desirability index of 0.927512. The fact that we found a quite good part at random among the first 15 experiments, simplifies and shortens the

| id | step | $w_i$ | $\alpha_1$ | $\alpha_2$ | $\beta$ | $w_{iobs}$ | $\alpha_{1obs}$ | $\alpha_{2obs}$ | $\beta_{obs}$ | DI |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Initial | 37 | 47.0 | 68.0 | 1.82 | 40.116 | 44.336 | 64.466 | 1.752 | 0.000000 |
| 2 | Initial | 31 | 47.0 | 68.0 | 1.88 | 36.346 | 44.386 | 65.956 | 1.818 | 0.000000 |
| 3 | Initial | 31 | 47.0 | 68.0 | 1.52 | 34.483 | 42.372 | 55.951 | 1.481 | 0.784844 |
| 4 | Initial | 36 | 68.0 | 50.5 | 1.52 | 39.498 | 53.857 | 46.587 | 1.480 | 0.745266 |
| 5 | Initial | 30 | 68.0 | 50.5 | 1.52 | 33.482 | 53.094 | 47.318 | 1.485 | 0.833043 |
| 6 | Initial | 29 | 47.0 | 50.5 | 1.88 | 33.402 | 44.869 | 49.124 | 1.809 | 0.792879 |
| 7 | Initial | 36 | 68.0 | 68.0 | 1.82 | 41.748 | 61.534 | 63.062 | 1.749 | 0.000000 |
| 8 | Initial | 37 | 57.5 | 71.5 | 1.46 | 40.011 | 48.112 | 52.406 | 1.424 | 0.676651 |
| 9 | Initial | 36 | 47.0 | 50.5 | 1.52 | 39.253 | 42.051 | 46.549 | 1.477 | 0.823559 |
| 10 | Initial | 30 | 50.5 | 47.0 | 1.52 | 33.188 | 44.665 | 44.136 | 1.482 | **0.927512** |
| 11 | Initial | 31 | 68.0 | 47.0 | 1.88 | 36.353 | 63.029 | 45.524 | 1.795 | 0.000000 |
| 12 | Initial | 30 | 68.0 | 68.0 | 1.82 | 36.338 | 62.765 | 64.784 | 1.765 | 0.000000 |
| 13 | Initial | 32 | 68.0 | 71.5 | 1.46 | 35.407 | 51.675 | 54.656 | 1.424 | 0.710951 |
| 14 | Initial | 36 | 64.5 | 47.0 | 1.88 | 41.358 | 60.215 | 45.828 | 1.794 | 0.000000 |
| 15 | Initial | 35 | 47.0 | 47.0 | 1.88 | 39.387 | 44.647 | 46.198 | 1.792 | 0.757240 |
| 16 | Step 2 | 32 | 50.5 | 47.0 | 1.58 | 35.338 | 45.749 | 43.996 | 1.529 | **0.967669** |
| 17 | Step 2 | 32 | 50.5 | 43.5 | 1.58 | 35.419 | 46.199 | 40.888 | 1.530 | 0.929059 |
| 18 | Step 2 | 32 | 40.0 | 40.0 | 1.58 | 35.142 | 37.618 | 38.667 | 1.529 | 0.834656 |
| 19 | Step 2 | 31 | 50.5 | 50.5 | 1.52 | 34.540 | 45.752 | 46.651 | 1.481 | 0.944306 |
| 20 | Step 2 | 31 | 47.0 | 47.0 | 1.58 | 34.318 | 43.024 | 44.475 | 1.539 | 0.950238 |
| 21 | Step 3 | 32 | 50.5 | 47.0 | 1.52 | 34.944 | 44.456 | 43.376 | 1.477 | 0.949759 |
| 22 | Step 3 | 31 | 75.0 | 43.5 | 1.52 | 34.222 | 54.747 | 41.151 | 1.476 | 0.792572 |
| 23 | Step 3 | 31 | 61.0 | 47.0 | 1.52 | 34.523 | 51.050 | 43.816 | 1.477 | 0.884666 |
| 24 | Verif. | 32 | 50.5 | 50.5 | 1.52 | 35.575 | 45.102 | 46.829 | 1.482 | 0.947153 |
| 25 | Verif. | 31 | 50.5 | 47.0 | 1.58 | 34.513 | 45.740 | 44.532 | 1.541 | 0.966055 |
| 26 | Verif. | 32 | 50.5 | 47.0 | 1.58 | 35.308 | 45.240 | 44.074 | 1.529 | **0.973921** |
| mean(26,16) | | 32 | 50.5 | 47.0 | 1.58 | 35.323 | 45.495 | 44.035 | 1.529 | 0.970812 |

Table 6.9: Experiments and observations with the resulting desirability index $DI$ for the necking-in optimization

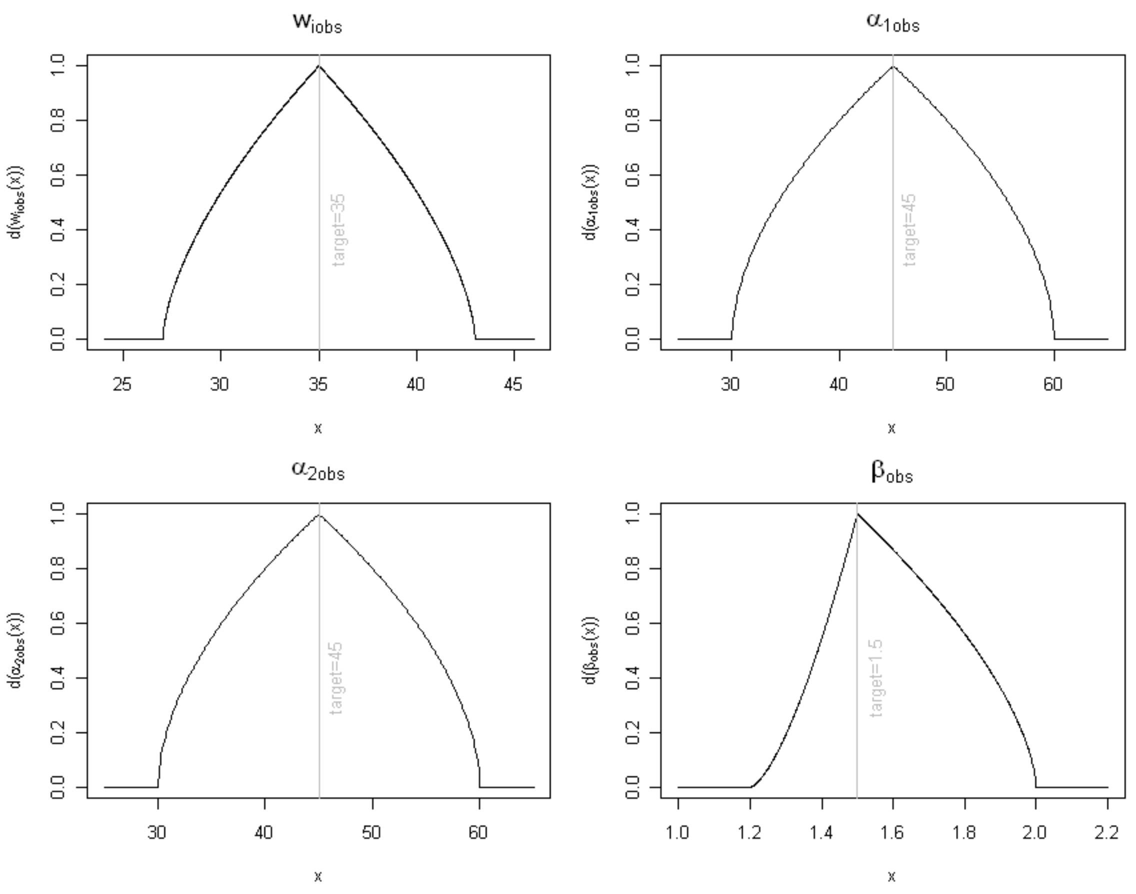


Figure 6.6: Specified desirability functions for the quality characteristics in the necking-in process optimization

| characteristic | $LSL$ | $target$ | $USL$ | $l$ | $r$ |
|---|---|---|---|---|---|
| inner width $w_{iobs}$ | 27 | 35 | 43 | 0.63 | 0.63 |
| angle $\alpha_{1obs}$ | 30 | 45 | 60 | 0.55 | 0.55 |
| angle $\alpha_{2obs}$ | 30 | 45 | 60 | 0.55 | 0.55 |
| necking-in ratio $\beta_{obs}$ | 1.2 | 1.5 | 2 | 1.5 | 0.63 |

Table 6.10: Parameters used to specify Derringer Suich desirability functions for the quality characteristics in the necking-in process optimization

optimization. Altogether, the goodness of the parts varies from very poor (0) to very good ($> 0.92$).

### 6.2.3 Model selection

Like in the first case study, we try various Kriging models with different deterministic trend component and differently parameterized power exponential covariance functions for each objective separately. Isotropic and anisotropic power exponential covariance functions with smoothness s=1 and s=2 are considered as covariance of the stochastic process. For the deterministic trend component we fit the intercept $\mu$ only and the linear trend component consisting of all effects $\mu + w_i + \alpha_1 + \alpha_2 + \beta$. If one or more effects in the linear trend component are not significant, we do a stepwise backwards selection until we get a model where all effects are significant. An overview of all fitted surrogate models and how we select the actual surrogate model is attached in Appendix A.4.

Table 6.11 summarizes the chosen models for every parameter. It gives the estimates for all parameters after every optimization cycle. Note, that for all models the influencing parameters are scaled to the interval $[0, 1]$ before the models are fitted, because the influencing parameters have quite different scales. The parameter estimates of the initial surrogate models are shown in the third column.

### 6.2.4 Sequential optimization

To start the sequential optimization of the process, we now have to choose a parameterization of the $\alpha$ levels. As said before, this is an optimization problem with four objectives and a four dimensional parameter space represented by 14641 grid points. The computation time per optimization cycle shall be restricted to 10 hours, to run at least one cycle per day. Therefore, we decide to choose five different confidence levels. As suggested in Section 4.2 we take the levels $\alpha \in \{0.01, 0.1, 0.5, 0.7, 0.9\}$.

Table 6.12 summarizes the progress of the target value optimization. mtEGO needs four cycles: the initial design, two sets of updating points,and a verification phase. After every cycle of the procedure the optimum observed parameter setting with its observations and desirability index $DI$ and the optimum predicted point with all values is noted. The optimum predicted point is that point that has the best $DI$ value among all predictions for all grid points of the parameter space predicted with the

| model component | parameter | estimator initial design | estimator optimization step 1 | estimator optimization step 2 | estimator verification step |
|---|---|---|---|---|---|
| **inner width** $w_i$ | | | | | |
| deterministic trend component | $\mu$ | 30.2 | 30.4 | 30.4 | 30.4 |
| | $w_i$ | 9.41 | 9.33 | 9.48 | 9.47 |
| | $\alpha_1$ | 1.69 | 1.59 | 1.26 | 1.22 |
| | $\beta$ | 2.83 | 2.83 | 3 | 2.95 |
| power exponential covariance function isotropic (s=2) | $\theta$ | 0.657 | 0 | 0 | 0 |
| | $\sigma_Z^2$ | 0.669 | 0.242 | 0.257 | 0.242 |
| **ascent angle** $\alpha_1$ | | | | | |
| deterministic trend component | $\mu$ | 47.75 | 45.56 | 42.13 | 49.31 |
| power exponential covariance function anisotropic (s=2) | $\theta_{wi}$ | 0.051 | 0.0193 | 0.037 | 0.4403 |
| | $\theta_{\alpha 1}$ | 1.662 | 2.965 | 0.963 | 2.391 |
| | $\theta_{\alpha 2}$ | 0.0025 | 3.865 | 3.487 | 7.373 |
| | $\theta_{\beta}$ | 1.527 | 0.959 | 0.719 | 1.693 |
| | $\sigma_Z^2$ | 70.025 | 105.71 | 164.06 | 64.06 |
| **descent angle** $\alpha_2$ | | | | | |
| deterministic trend component | $\mu$ | 41.87 | 45.54 | 44.13 | 46.37 |
| power exponential covariance function isotropic (s=2) | $\theta$ | 2.763 | 2.119 | 1.365 | 0.847 |
| | $\sigma_Z^2$ | 1240.03 | 676.23 | 191.04 | 80.73 |
| **necking-in ratio** $\beta$ | | | | | |
| deterministic trend component | $\mu$ | 1.37 | 1.37 | 1.37 | 1.37 |
| | $\beta$ | 0.542 | 0.541 | 0.542 | 0.542 |
| power exponential covariance function isotropic (s=2) | $\theta$ | 0.725 | 4.7E-17 | 3.9E-17 | 0 |
| | $\sigma_Z^2$ | 0.0000983 | 0.000051 | 0.000045 | 0.000044 |

Table 6.11: Surrogate models

| | $w_i$ | $\alpha_1$ | $\alpha_2$ | $\beta$ | $w_{iobs}$ | $\alpha_{1obs}$ | $\alpha_{2obs}$ | $\beta_{obs}$ | DI |
|---|---|---|---|---|---|---|---|---|---|
| optimum after initial design (points 1-15) | | | | | | | | | |
| observed | 30 | 50.5 | 47 | 1.52 | 33.188 | 44.665 | 44.136 | 1.582 | 0.9275 |
| predicted | 32 | 50.5 | 47 | 1.58 | 35.528 | 45.505 | 44.486 | 1.532 | 0.9697 |
| optimum after optimization step 1 (points 1-20) | | | | | | | | | |
| observed | 32 | 50.5 | 47 | 1.58 | 35.338 | 45.749 | 43.996 | 1.529 | 0.9677 |
| predicted | 32 | 50.5 | 47 | 1.58 | 35.338 | 45.749 | 43.996 | 1.529 | 0.9677 |
| optimum after optimization step 2 (points 1-23) | | | | | | | | | |
| observed | 32 | 50.5 | 47 | 1.58 | 35.338 | 45.749 | 43.996 | 1.529 | 0.9677 |
| predicted | 32 | 50.5 | 47 | 1.58 | 35.338 | 45.749 | 43.996 | 1.529 | 0.9677 |
| optimum after optimization step 3 & verification (points 1-26) | | | | | | | | | |
| observed | 32 | 50.5 | 47 | 1.58 | 35.308 | 45.240 | 44.074 | 1.529 | 0.9739 |
| predicted | 32 | 50.5 | 47 | 1.58 | 35.308 | 45.240 | 44.074 | 1.529 | 0.9739 |

Table 6.12: Results of the sequential optimization for the necking-in process.

currently fitted model. After the initial step the observed optimal setting is $w_i = 30, \alpha_1 = 50.5, \alpha_2 = 47$ and $\beta = 1.52$. The observed results reach a desirability index of 0.9275, which is a really good result already. Though the predictions forecast that the setting $w_i = 32, \alpha_1 = 50.5, \alpha_2 = 47$ and $\beta = 1.58$ even reaches $DI = 0.9697$. The first cycle of mtEGO suggests 13 candidates for the global optimum. With the help of the dendrogram and scree plot (see first row of Figure 6.7) the candidates are clustered in four groups and representatives are chosen for each cluster. Additionally, the predicted optimum point is added as updating point (see Table 6.9 for the exact points). The observed optimum among the 15 initial and 5 new updating points now is the previously predicted optimum. The observations for this setting are only slightly worse than predicted and have $DI = 0.9677$. The DI has improved approximately 4.3%, where the DI can improve approximately 7.8% at maximum (assuming the theoretical maximum value of $DI = 1$ can be reached with one of the parameter space grid points). The predicted joint optimum now is the same setting as the already observed optimum. Leaving the uncertainty aside, this means that according to the fitted model the found optimum is the best point in the parameter space. The second cycle of mtEGO suggests 28 candidates, which are clustered in three groups after consulting the dendrogram and

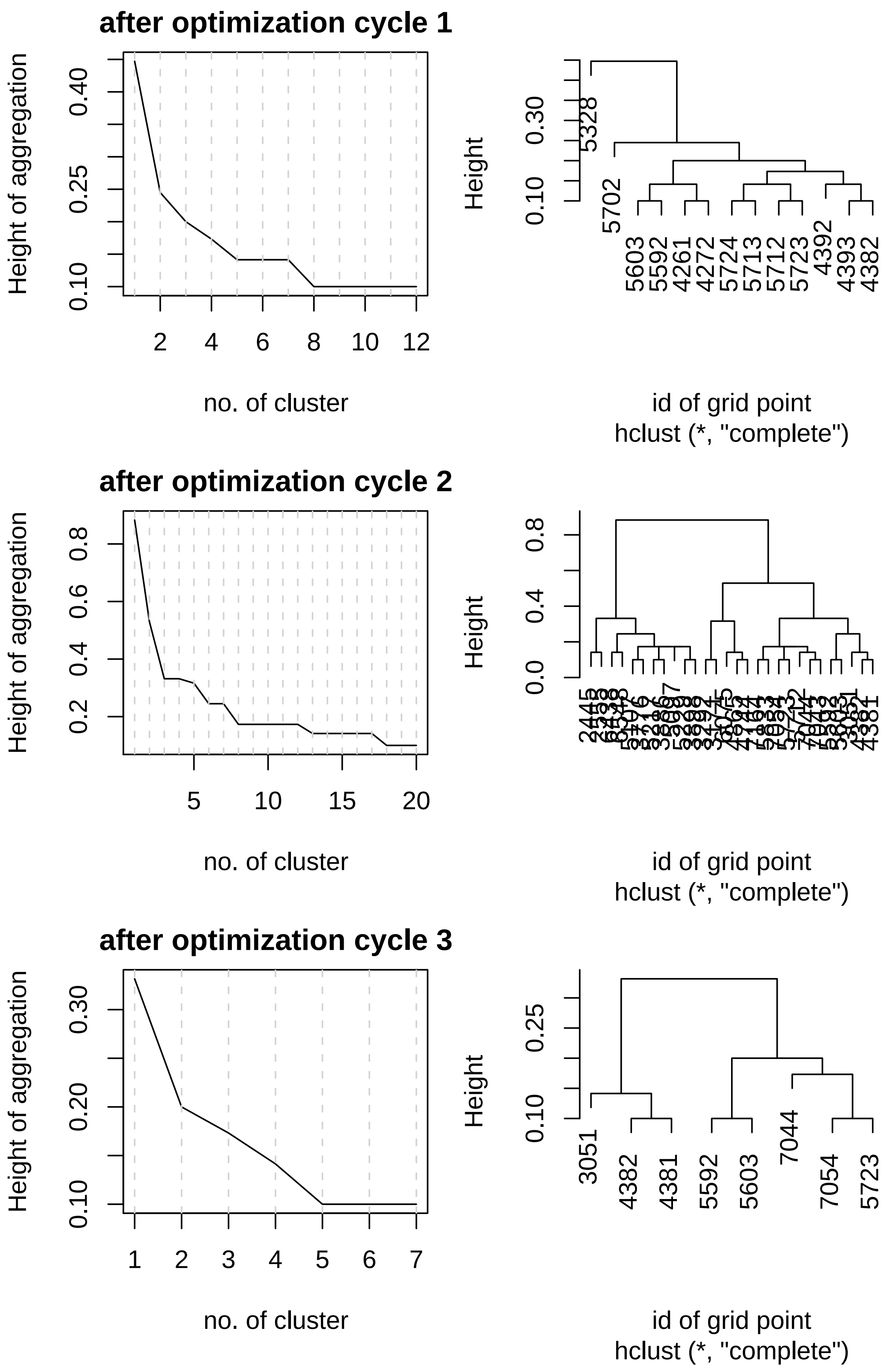


Figure 6.7: Cluster dendrograms and scree plots of the candidate sets for every optimization cycle for the necking-in optimization

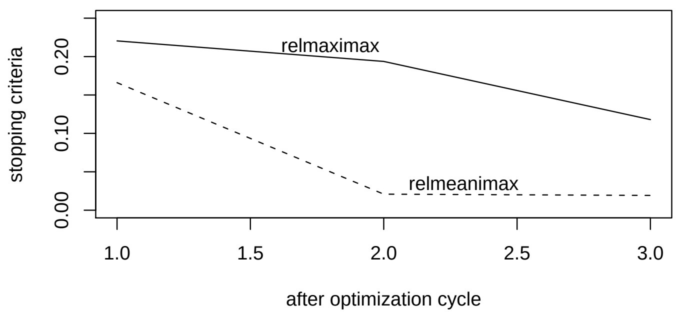


Figure 6.8: Progress of the stopping criterion for the necking-in optimization

the scree plot again (cf. second row of Figure 6.7). Three representatives are chosen and added to the design. The predicted optimum is already available and therefore not added again. The observed and the predicted optimum after this second cycle are still the same as after cycle 1. The recurring optimal points indicate to stop the algorithm now. However, mtEGO is run a third time to determine the stopping criterion again. Figure 6.8 shows the progress of the criterion. The maximax criterion gets clearly smaller and the meanimax criterion already saturates slightly. We could stop here, but to be more certain about our results we decide to add more updating points. The third cycle of mtEGO suggests only 8 candidates (also an indicator that we are close to the end of the procedure). The dendrogram and scree plot show two cluster (third row of Figure 6.7). Two representatives are added as updating points, which are the points with second and third best $DI$ and are the grid points in the parameter space next to the found optimum. Actually, the twenty best points are all lying next to the found optimum, which indicates that there is only one global optimum and no local optimum of about the same size. The best $DI$ is still our found optimum. Since our CNC machine has been modified to be used for other experiments between optimization cycle 2 and 3 we add the found optimum once again to the design to assure the repeatability of the experiments. The third optimization cycle becomes a verification step. The results in Table 6.9 verify what is already known. The found optimum stays the best point and the other two settings also yield very good results. The replication of the optimal parameter setting could now yield even better results and has an index of $DI = 0.9739$ now. For the finally fitted models we took the average of the two replications of the parameter setting that is noted below all other experiments in the design table. The predicted optimum in the final models is again the observed optimum. The algorithm stops after finding the same optimal point for three steps and the stopping criterion is saturating. The tube formed with the necking-in process could be optimized with the help of mtEGO successfully within only 26 experiments.

# 7 Summary and outlook

In engineering processes the specification of optimization targets is usually reduced to minimization or maximization problems. The specification of challenging multivariate target structures is excluded, due to the lack of algorithms that are able to handle them. Often however the optimum is a precise target instead of a minimum or maximum and it would be helpful if the deviation from the target could be penalized asymmetrically.

In this thesis a new heuristic named mtEGO for multi-objective target value sequential optimization has been developed. A small initial space-filling design is used to fit a surrogate model for each objective of the optimization problem. Based on the predictions and prediction errors of the surrogate model for the whole parameter space virtual observations with different $(1 - \alpha)$ confidence levels are constructed. These virtual observations are used to roughly simulate the effect of the model uncertainty on the capability of each setting in the parameter space to be the global optimum. A transformation with desirability functions and the aggregation to a joint desirability index turns the multi-objective target value problem in a simple single-objective maximization problem. Improvements are determined for this single-objective maximization problem then, which are maximized to find the global optimum. mtEGO therefore works in a hybrid way, which means for each combination of $(1 - \alpha)$ confidence levels an own candidate for the global optimum is determined simultaneously. The candidates are reduced to a small number of updating points using hierarchical clustering. Finally, the model is refined with the observations from the updating points and the algorithm proceeds to generate and add new updating points until the stopping criterion is fulfilled.

In contrast to the existing approaches, mtEGO is not based on any distribution assumptions and can be used with any kind of desirability function and desirability index. For example one- and two-sided Harrington or Derringer-Suich desirabilities can be used. In addition, the user can vary e.g. between the weighted and unweighted geometric mean as the desirability index. This enables a precise specification of the optimization target according to the users requirements.

The mtEGO algorithm was validated successfully by means of extensive simulation studies and two case studies from mechanical engineering. Beside the fact that the two case studies demonstrate the applicability of mtEGO to real applications, they show that mtEGO even works successfully if basic conditions change in an ongoing optimization process. Another benefit of the mtEGO approach for real applications is the fact that it permits to update the model with several points at the same time in each step. If real experiments are performed, it is economically not reasonable to start the whole machine and e.g. code a CNC programme for only one experiment.

Further, an improved variant of mtEGO,named mtEGOimp, was developed. It does a pre-selection of reasonable confidence levels before cross-combining them. As a consequence, the computation time of the mtEGO approach is strongly reduced, which relaxes time limitations. The incorporation of a convex hull restriction method for failure points and an imputation of missing values into the mtEGO approach, finally extends it to a powerful tool for optimization problems even in the presence of unknown constraints.

Nevertheless, the mtEGO approach suffers from the following weak points.

- If the optimum to be found is lying on a kind of flat hill, the algorithm has problems to locate the optimum precisely. Likewise, the algorithm has problems to locate optima that lie on the boundary of the parameter space, particularly if they lie on the boundary of an unknown constraint. These problems are a consequence of constructional principles and it is necessary to insert some kind of special rules to solve them.

- In some cases the satisfaction of the stopping criterion is hard to recognize for an unexperienced user. One should aim for a clearer stopping criterion that is a measure that has to exceed or underrun a ceratin threshold.

- The automatized choice of the number of clusters sometimes fails. To recognize the failure of the criterion is another challenge for an unexperienced user. Other criteria that are more precise should be developed here, eventually including some cooling scheme that permits to take less updating points the closer the target is.

- The pre-selection of the $\alpha$ levels used for the confidence intervals in mtEGOimp is currently done identically for the whole parameter space. Actually, the different regions of uncertainty in the parameter space usually could permit different selections of levels and the reduction could hence be more efficient. However, a separation of the regions of uncertainty is currently not available.

The weak points listed before are a good starting point for future work. To better evaluate future criteria for the automatic choice of clusters or future stopping criteria it would be reasonable to develop a representative suite of test problems especially for target-value optimization problems. On the basis of this new test suite further research on different initial designs, different parameterizations, improved stopping criteria, and different cluster methods could be done. Particularly, the exemplary advanced test problems, could be evaluated with a representative test suite to examine the properties of the mtEGO approach in such situations more precisely.

# Bibliography

Akaike, H. (1974) Information theory and extension of the maximum likelihood principle. In Petrov, B. and F.Czaki, B., Academic Kiado (editors), *Proceedings of the 2nd International Symposium on Information.*

Aldenderfer, M. and Blashfield, R. (1984) Cluster analysis. *Sage University Paper series on Quantitative Applications in the Social Sciences* 07-044.

Cormack, R. (1971) A review of classification. *Journal of the Royal Statistical Society A* 134, pp. 321–367.

Cox, D. and John, S. (1997) Sdo: A statistical method for global optimization. In Alexandrow, N. and Hussaini, M. (editors), *Multidisciplinary Design Optimization: State of the Art.* SIAM, Philadelphia, pp. 315–329.

Cressie, N. (1993) *Statistics for Spatial Data.* J.Wiley, New York.

Deb, K.; Thiele, L.; Laumanns, M. and Zitzler, E. (2001) Scalable test problems for evolutionary multi-objective optimization. *TIK-Technical Report* 112.

Derringer, G. and Suich, D. (1980) Simultaneous optimization of several response variables. *Journal of Quality Technology* 12(4), pp. 214–219.

Everitt, B. (1980) *Cluster Analysis.* Social Science Research Council, Halsted Press.

Fang, K.; Lin, D.; Winker, P. and Zhang, Y. (2000) Uniform design: Theory and application. *Technometrics* 42, pp. 237–248.

Forrester, A.; Sóbester, A. and Keane, A. (2006) Optimization with missing data. *Proceedings of the Royal Society A* 462, pp. 935–945.

Fukuda, K. and Prodon, A. (1996) Double description method revisited. In *Combinatorics and Computer Science*, *Lecture Notes in Computer Science*, volume 1120. Springer, pp. 91–111.

Govaerts, B. and Le Bailly de Tilleghem, C. (2005) Distribution of desirability index in multicriteria optimization using desirability functions based on the cumulative distribution function of the standard normal. Technical report, Discussion Paper 0531, Institut de Statistique.

Gutmann, H. (2001) A radial basis function method for global optimization. *Journal of Global Optimization* 19(3), pp. 201–227.

Handcock, M. (1991) On cascading latin hypercube designs and additive models for experiments. *Communications Statistics-Theory Methods* 20, pp. 417–439.

Harrington, J. (1965) The desirability function. *Industrial Quality Control* 21(10), pp. 494–498.

Hawe, G. and Sykulsky, J. (2004) An enhanced probability of improvement utility function for locating pareto optimal solutions. *Transactions on Magnetics* , pp. 965–966.

Hawe, G. and Sykulsky, J. (2007) A hybrid one-then-two stage algorithm for comutationally expensive electromagnetic design optimization. *Journal for Computation and Mathematics in Electrical and Electronic Enigneering* 26(2), pp. 236–246.

Henkenjohann, N. (2006) *Eine adaptive sequentielle Prozedur zur effizienten Optimierung des CNC-gesteuerten Drückprozesses.* Ph.D. thesis, University of Dortmund, Germany.

Henkenjohann, N.; Göbel, R.; Kleiner, M. and Kunert, J. (2005) An adaptive sequential procedure for efficient optimization of the sheet metal spinning process. *Quality and Reliability Engineering International* 21, pp. 439 – 455.

Huang, D.; Allen, T.; Notz, W. and Zheng, N. (2006) Global optimization of stochastic black-box systems via kriging meta-models. *Journal of Global Optimization* 34(3), pp. 441–466.

Jain, A. and Dubes, R. (1988) *Algorithms for Clustering Data.* Prentice Hall Advanced Reference Series, New Jersey.

Jeong, S. and Obayashi, S. (2005) Efficient global optimization (ego) for mulit-objective problem and data mining. In *The IEEE Congress on Evolutioary Computation, 2-5 September.*

Jones, D. (2001) A taxonomy on global optimization methods based on response surfaces. *Jorunal of Global Optimization* 21, pp. 345–383.

Jones, D.; Schonlau, M. and Welch, W. (1998) Efficient global optimization of expensive black-box functions. *Journal of Global Optimization* 13, pp. 455–492.

Kim, K. and Lin, D. (2000) Simultaneous optimization of mechancial properties of steel by maximizing desirability functions. *Applied Statistics* 49(3), pp. 311–326.

Knowles, J. (2005) Parego: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation* 10(1), pp. 50–66.

König, W. and Klocke, F. (1995) *Fertigungsverfahren, Band 5, Blechbearbeitung*, volume 3. VDI Verlag, Düsseldorf.

Kunert, J.; Tekkaya, A. E.; Kwiatkowski, L.; Melsheimer, O. and Straatmann, S. (2007) Use of experimental design to analyse a necking-in process. In *ENBIS 7, Dortmund, 24-26 September, Dortmund, Germany.*

Kushner, H. (1964) A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise. *Joural of Basic Engineering* 86, pp. 97–106.

Morris, M. and Mitchell, T. (1995) Exploratory designs for computational experiments. *Journal of Statistical Planning and Inference* 43, pp. 381–402.

Müller, W. (2001) *Optimum design 2000.* Kluwer Academics Publsihers.

O'Conell, M. and Wolfinger, R. (1997) Spatial regression models, response surfaces, and process optimization. *Jorunal of Computational and Graphical Statistics* 6, pp. 224–241.

Okabe, T.; Jin, Y.; Olhofer, M. and Sendhoff, B. (2004) On test functions for evolutionary multi-objective optimization. *Lecture Notes in computer Science* 3242, pp. 792–802.

Osio, I. and Amon, C. (1996) An engineering design methodology with multistage bayesian surrogates and optimal sampling. *Research in Engineeering Design* 8, pp. 189–206.

Owen, A. (1992) Orthogonal arrays for computer experiments, integration and visualization. *Statistica Sinica* 2, pp. 439–452.

Pareto, V. (1896) *Cours D'Economie Politique, Volume I and II.* F.Rouge, Lausanne.

Patterson, H. and Thompson, R. (1974) Maximum likelihood estimation of variance components. In *Proceedings of the 8th International Biometric Conference.* pp. 197–207.

Queipo, N.; Pintos, S.; Rincón, N.; Contreras, N. and Colmenares, J. (2002) Surrogate modeling-based optimization for the integration of static dynamic data into a reservoir description. *Journal of Petroleum Science and Engineering* 35, pp. 167–181.

R Development Core Team (2005) *R: A language and environment for statistical computing.* R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0.

Sacks, J.; Welch, W.; Mitchell, T. and Wynn, H. (1989) Design and analysis of computer experiemnts. *Statistical Science* 4, pp. 409–423.

Santner, T. J.; Williams, B. and Notz, W. (2003) *The Design and Analysis of Computer Experiments.* Springer-Verlag.

Sasena, M. (2002) *Flexibility and Efficiency Enhancements for Constrained Global Design Optimization with Kriging Approximations.* Ph.D. thesis, University of Michigan, USA.

Sasena, M.; Papalambros, P. and Goovaerts, P. (2000) Metamodeling sampling criteria in a global optimization framework. *American Instiute of Aeronautics and Astronautics* 4921, pp. 1–11.

Schonlau, M. (1997) *Computer experiments and global optimization.* Ph.D. thesis, Department of Statistics and Actuarial Science, University of Waterloo, Waterloo, CA.

Schwarz, G. (1975) Estimating the dimension of a model. *Annals of Statistics* 16, pp. 461–464.

Sóbester, A.; Leary, S. and Keane, A. (2005) On the design of optimization strategies based on global response surface approximation models. *Journal of Global Optimization* 33, pp. 31–59.

Steuer, D. (2005) *Statistische Eigenschaften der multikriteriellen Optimierung mittels Wünschbarkeiten*. Dissertation, Universität Dortmund.

Törn, A. and Žilinskas, A. (1987) *Global optimization*. Springer, Berlin.

Trautmann, H. and Weihs, C. (2004) Pareto-optimality and desirability indices. *Technical Report SFB 475* 63.

Trautmann, H. and Weihs, C. (2006) On the distribution of the desriability index using harrington's desirability function. *Metrika* 63, pp. 207–213.

Turner, C.; Crawford, R. and Campbell, M. (2007) Multidimensional sequential sampling for nurbs-based metamodel development. *Engineering with Computers* 23, pp. 155–174.

Van Veldhuizen, D. and Lamont, G. (1999) Multiobjective evolutionary algorithm test suites. In *Proceedings of the ACM Symposium on Applied Computations*. San Antonia, Texas.

Villemonteix, J.; Vazquez, E. and Walter, E. (2009) An informational approach to the global optimization of expensive-to-evaluate functions. *Journal of Global Optimization* 44, pp. 509–534.

Wang, G. (2003) Adaptive response surface method using inherited latin hypercube design points. *ASME Transaction, Journal of Mechanical design* 125, pp. 210–220.

Wang, G. and Simpson, T. (2004) Fuzzy clustering based hierarchical metamodeling for space reduction and design optimization. *Journal of Engineering Optimization* 36, pp. 323–335.

Weber, H. and Weihs, C. (2003) On the distribution of the desirability index using harrington's desirability function. *Forschungsbericht* 3, Department of Statistics, University of Dortmund, Germany.

Williams, B.; Santner, T. and Notz, W. (2000) Sequential design of computer experiments to minimize integrated response functions. *Statistica Sinica* 10, pp. 1133–1152.

Wolfinger, R.; Tobias, R. and Sall, J. (1994) Computing gaussian likelihood and their derivatives for generalized linear mixed models. *SIAM Journal on Scientific Computing* 15, pp. 1294–1310.

# A Appendix

## A.1 Proofs for Step 3.C.1 of mtEGO

In Step 3.C.1 of mtEGO it is stated that if the point with the currently best prediction has not already been observed, it is the candidate for the desirability index of the pure predictions and hence included in the set of candidates. If it equals the current observed optimum, no candidate is suggested for this single desirability index and the point is not included in the final candidate set.

Suppose $\chi$ is the parameter space and $x^* \in \chi$ is the point with the current best prediction. Let further $x_{curopt}$ be the currently observed optimal point, which means $DI(x_{curopt}) = DI_{max}$.

**Part 1:**
The point with the best prediction maximizes the improvement values of the desirability index of these pure predictions.
If $x^*$ is the point with the best predictions

$$\begin{array}{lcr} & DI(x^*) \geq DI(x) & \forall x \in \chi \\ \Leftrightarrow & DI(x^*) - DI_{max} \geq DI(x) - DI_{max} & \forall x \in \chi \\ \Leftrightarrow & I(x^*) \geq I(x) & \forall x \in \chi \end{array}$$

$\Leftrightarrow$ $x^*$is the point with the best improvement value.

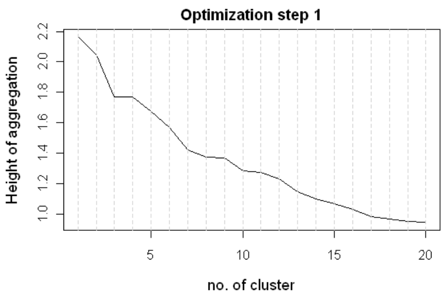
**Part 2:**
If the best prediction is exactly the already observed current optimum, i.e. $x^* = x_{curopt}$, no point is suggested as a candidate for the desirability index of the pure predictions.

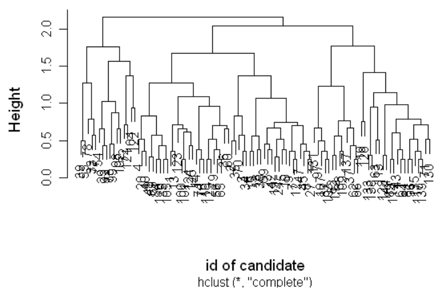
$$\begin{array}{lc} & DI(x^*) = DI(x_{curopt}) \\ \Leftrightarrow & DI(x^*) - DI_{max} = 0 = DI(x_{curopt}) - DI_{max} \\ \Leftrightarrow & I(x^*) = 0 = I(x_{curopt}) \end{array}$$

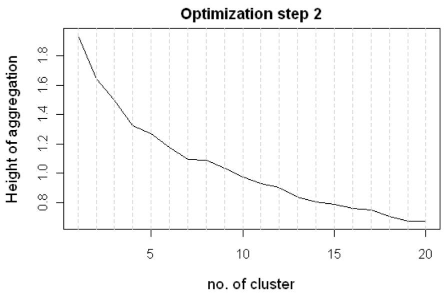
And since it was shown in part 1 of the proof, that $I(x^*) \geq I(x)$ $\forall x \in \chi$, here

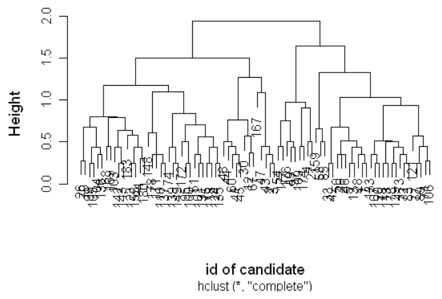
$I(x*) = I(x) = 0 \ \forall x \in \chi$. That means no point could improve the current results and hence no candidate is suggested.

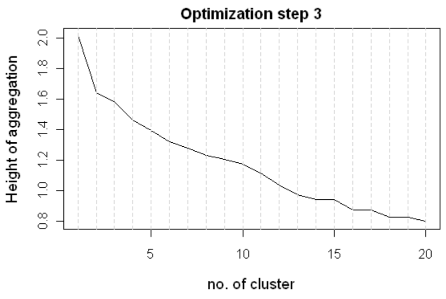
## A.2 Cluster dendrograms and scree plots of the candidate sets for each optimization step during the sheet metal spinning optimization

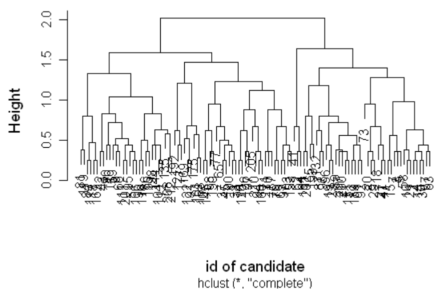


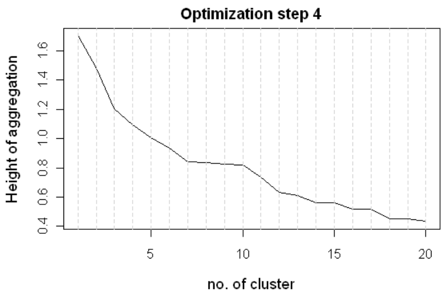


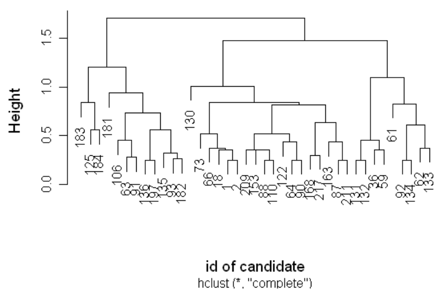












## A.3 Model selection for the initial surrogate model of the sheet metal spinning optimization

Various Kriging models with different deterministic trend components and differently parameterized power exponential covariance functions are fitted for each objective separately. Isotropic and anisotropic power exponential covariance functions with smoothness s=1 and s=2 are considered as covariance of the stochastic process. For the deterministic trend component we fit the intercept $\mu$ only and the linear trend component consisting of all effects $\mu + k_{hin} + rdw + eaqu + vrbk + k_{ruck} + f_s$. Depending on the significance of the effects in the linear choice for the trend component, we do a stepwise backwards selection until we get a model where all effects are significant. The different models with their corresponding number of estimated parameters, BIC and MSCVE values can be found in the tables below. In all three tables one can find NA's for the MSCVE values. When calculating the MSCVE one design point after another is taken away and the same model specification is fitted to the subset of the design. The difference of the prediction of the design point taken aside beforehand and its actually observed value is summarized to the MSCVE (for more details see Section 2.3.4). For the model specifications where NA is written it was not possible to fit the given model specification for one or several of the design subsets, because of problems during the estimation of the covariance parameters. The certain model specification seems maybe appropriate for the whole design, but not for a subset of the design. It is not appropriate to use these models for the optimization, since it is very uncertain if the model is good after refining it with updating points in the next steps and it is advisable to avoid too many model changes between the optimization steps. These models are only listed in the tables for the sake of completeness. The models with the best (smallest) BIC values and smallest MSCVE are bold in the table. The model highlighted with the grey background is the one finally chosen as the surrogate model for the optimization procedure.

**Surrogate models considered for the maximum cup diameter** $dmax$

| deterministic trend function | Power exponential covariance function | # parameters | MSCVE | BIC |
|---|---|---|---|---|
| $\mu$ | isotropic (s=1) | 3 | **7.52** | 39.1 |
| | isotropic (s=2) | 3 | 12.3325 | 40.1 |
| | anisotropic (s=1) | 8 | 14.8 | 41.6 |
| | anisotropic (s=2) | 8 | 18.7418 | 47.1 |
| $\mu + k_{hin} + rdw + eaqu + +vrbk + k_{ruck} + f_s$ | isotropic (s=1) | 9 | 9.57323 | **19.2** |
| | isotropic (s=2) | 9 | **8.51797** | **19.2** |
| | anisotropic (s=1) | 14 | NA | 31.6 |
| | anisotropic (s=2) | 14 | 9.13527 | 21.4 |

The cup diameter $damx$ is modeled best with the linear deterministic trend component $\mu + k_{hin} + rdw + eaqu + vrbk + k_{ruck} + f_s$ and an isotropic power exponential covariance function with smoothness $s = 2$. The same model specification with smoothness $s = 1$ yields the same minimum BIC values, but worse MSCVE values. Hence, we decide for the smoother version with better MSCVE values. The best MSCVE value has the simple ordinary Kriging model with only the intercept as trend component. However, it has clearly worse BIC values, that do not compensate the fewer number of unknown parameters.

**Surrogate models considered for the cup depth** $nt$

| deterministic trend function | Power exponential covariance function | # parameters | MSCVE | BIC |
|---|---|---|---|---|
| $\mu$ | isotropic (s=1) | 3 | 404.392 | 102.2 |
| | isotropic (s=2) | 3 | 251.827 | 97.6 |
| | anisotropic (s=1) | 8 | 425.367 | 103.6 |
| | anisotropic (s=2) | 8 | 565.689 | 99.2 |
| $\mu + k_{hin} + rdw + eaqu + +vrbk + k_{ruck} + f_s$ | isotropic (s=1) | 9 | 184.295 | **44.6** |
| | isotropic (s=2) | 9 | 184.295 | **44.6** |
| | anisotropic (s=1) | 14 | **172.705** | 57.1 |
| | anisotropic (s=2) | 14 | NA | 53.1 |

For the cup depth $nt$ the Kriging model with the linear deterministic trend component $\mu + k_{hin} + rdw + eaqu + vrbk + k_{ruck} + f_s$ yields far better results then the ordinary Kriging model using only the intercept as trend component. The BIC value is best for an isotropic power exponential covariance function whatever smoothness is used. The MSCVE value is minimum fitting an anisotropic power exponential covariance function with smoothness $s = 1$. Due to the idea to use as few unknown parameters as possible but as smooth functions as possible, we decide to model $nt$ by means of an isotropic power exponential covariance function with smoothness $s = 2$.

**Surrogate models considered for the sheet thickness** $smin$

| deterministic trend function | Power exponential covariance function | # parameters | MSCVE | BIC |
|---|---|---|---|---|
| $\mu$ | isotropic (s=1) | 3 | 0.247 | 0.2 |
| | isotropic (s=2) | 3 | 0.0688 | -12 |
| | anisotropic (s=1) | 8 | NA | -0.9 |
| | anisotropic (s=2) | 8 | NA | -16.4 |
| $\mu + k_{hin} + rdw + eaqu +$ $+ vrbk + k_{ruck} + f_s$ | isotropic (s=1) | 9 | **0.06468** | -18 |
| | isotropic (s=2) | 9 | 0.09146 | **-20.1** |
| | anisotropic (s=1) | 14 | 0.24446 | -7.6 |
| | anisotropic (s=2) | 14 | NA | -7.6 |

The Kriging model with the linear deterministic trend component $\mu + k_{hin} + rdw + eaqu + vrbk + k_{ruck} + f_s$ and an isotropic power exponential covariance function yields the best model fit. The MSCVE value is minimum using a smoothness value of $s = 1$. The BIC value is minimum using a smoothness value of $s = 2$. Since, there are no more differences between the two models but the smoothness, we decide for the smoother model with $s = 2$.

## A.4 Model selection for the initial surrogate model of the necking-in optimization

This subsection gives an insight in the model selection procedure for surrogate model for the necking-in optimization. For the necking-in optimization the same models are fitted as for the optimization of the sheet metal spinning process. Further, the selection of the appropriate model and the comments on the problems with the MSCVE values for the necking-in case study are the same as for the sheet metal spinning case study (cf. Appendix A.3). We again fit various Kriging models with different deterministic trend component and differently parameterized power exponential covariance functions for each objective separately. For the deterministic trend component we fit the intercept $\mu$ only and the linear trend component consisting of all effects $\mu + k_{hin} + rdw + eaqu + vrbk + k_{ruck} + f_s$. Depending on the significance of the effects in the linear choice for the trend component, we do a stepwise backwards selection until we get a model where all effects are significant. The different models with their corresponding number of estimated parameters, BIC and MSCVE values can be found in the tables below. In all three tables one can find NA's for the MSCVE values. When calculating the MSCVE one design point after another is taken away and the same model specification is fitted to the subset of the design. The difference of the prediction of the design point taken aside beforehand and its actually observed value is summarized to the MSCVE (for more details see Section 2.3.4). For the model specifications where NA is written it was not possible to fit the given model specification for one or several of the design subsets, because of problems during the estimation of the covariance parameters. The certain model specification seems maybe appropriate for the whole design, but not for a subset of the design. It is not appropriate to use these models for the optimization, since it is very uncertain if the model is good after refining it with updating points in the next steps and it is advisable to avoid too many model changes between the optimization steps. These models are only listed in the tables for the sake of completeness. The models with the best (smallest) BIC values and smallest MSCVE are bold in the table. The model highlighted with the grey background is the one finally chosen as the surrogate model for the optimization procedure.

**Surrogate models considered for the inner width $w_i$**

| deterministic trend function | Power exponential covariance function | # parameters | MSCVE | BIC |
|---|---|---|---|---|
| $\mu$ | isotropic (s=1) | 3 | 15.61 | 62.8 |
| $\mu$ | isotropic (s=2) | 3 | 9.27 | 51.3 |
| $\mu$ | anisotropic (s=1) | 6 | 17.19 | 47.6 |
| $\mu$ | anisotropic (s=2) | 6 | 13.38 | 50.1 |
| $\mu + w_i + \alpha_1 + \alpha_2 + \beta$ | isotropic (s=1) | 7 | 6.84 | 24.2 |
| $\mu + w_i + \alpha_1 + \beta$ | isotropic (s=1) | 6 | 7.84 | 26.2 |
| $\mu + w_i + \alpha_1 + \alpha_2 + \beta$ | isotropic (s=2) | 7 | **6.57** | **23.2** |
| $\mu + w_i + \alpha_1 + \beta$ | isotropic (s=2) | 6 | 7.48 | 25.4 |
| $\mu + w_i + \alpha_1 + \alpha_2 + \beta$ | anisotropic (s=1) | 10 | NA | 26.6 |
| $\mu + w_i + \alpha_1 + \alpha_2 + \beta$ | anisotropic (s=2) | 10 | NA | 29.1 |
| $\mu + w_i + \alpha_1 + \beta$ | anisotropic (s=2) | 9 | NA | 31.3 |

For the inner width $w_i$ a Kriging model with the linear deterministic trend component $\mu + w_i + \alpha_1 + \alpha_2 + \beta$ and an isotropic power exponential covariance function with smoothness $s = 2$ yields the best MSCVE and BIC values. However, in this model the factor $\alpha_2$ is not significant. Using a deterministic trend component without $\alpha_2$, i.e. the trend component is $\mu + w_i + \alpha_1 + \beta$ yields almost as good MSCVE and BIC values. Since this reduced model has one parameter less, we choose it instead of the full model. Ordinary Kriging models with only the intercept as trend component or anisotropic covariance functions have clearly worse MSCVE and BIC values.

**Surrogate models considered for the angle $\alpha_1$**

| deterministic trend function | Power exponential covariance function | # parameters | MSCVE | BIC |
|---|---|---|---|---|
| $\mu$ | isotropic (s=1) | 3 | 139.56 | 91.6 |
| $\mu$ | isotropic (s=2) | 3 | 27.73 | 77.6 |
| $\mu$ | anisotropic (s=1) | 6 | 24.01 | 70.6 |
| $\mu$ | anisotropic (s=2) | 6 | **9.73** | 59.4 |
| $\mu + w_i + \alpha_1 + \alpha_2 + \beta$ | isotropic (s=1) | 7 | 83.26 | 50.5 |
| $\mu + w_i + \alpha_1 + \beta$ | isotropic (s=1) | 6 | 89.05 | 54.7 |
| $\mu + \alpha_1 + \beta$ | isotropic (s=1) | 5 | 76.47 | 55.9 |
| $\mu + w_i + \alpha_1 + \alpha_2 + \beta$ | isotropic (s=2) | 7 | 39.59 | 45.8 |
| $\mu + w_i + \alpha_1 + \beta$ | isotropic (s=2) | 6 | 50.11 | 51.9 |
| $\mu + \alpha_1 + \beta$ | isotropic (s=2) | 5 | 288.3 | 57.1 |
| $\mu + w_i + \alpha_1 + \alpha_2 + \beta$ | anisotropic (s=1) | 10 | 19.51 | **40.8** |
| $\mu + w_i + \alpha_1 + \beta$ | anisotropic (s=1) | 9 | 17.18 | 42.1 |
| $\mu + \alpha_1 + \beta$ | anisotropic (s=1) | 8 | **14.12** | 43.6 |
| $\mu + w_i + \alpha_1 + \alpha_2 + \beta$ | anisotropic (s=2) | 10 | NA | **37.5** |
| $\mu + w_i + \alpha_1 + \beta$ | anisotropic (s=2) | 9 | NA | **35.5** |
| $\mu + \alpha_1 + \beta$ | anisotropic (s=2) | 9 | NA | 48.2 |

With respect to the BIC values, the angle $\alpha_1$ could be modeled best with a linear trend component and anisotropic covariance function with $s = 2$, but here the determination of the MSCVE values is problematic and the models should rather be avoided. The best BIC value without having troubles when determining the MSCVE values is reached for the model with the linear trend component $\mu + w_i + \alpha_1 + \alpha_2 + \beta$ and anisotropic covariance function with smoothness $s = 1$. In contrast, the ordinary Kriging model with trend component $\mu$ and anisotropic covariance function with $s = 2$ has a much better MSCVE value and medium BIC value. Having only about half of the parameters than the previous model, we use this model for the optimization.

**Surrogate models considered for the angle $\alpha_2$**

| deterministic trend function | Power exponential covariance function | # parameters | MSCVE | BIC |
|---|---|---|---|---|
| $\mu$ | isotropic (s=1) | 3 | 198.18 | 93.8 |
| $\mu$ | isotropic (s=2) | 3 | **7.41** | **78.6** |
| $\mu$ | anisotropic (s=1) | 6 | NA | 71.7 |
| $\mu$ | anisotropic (s=2) | 6 | 136.70 | 67.9 |
| $\mu + w_i + \alpha_1 + \alpha_2 + \beta$ | isotropic (s=1) | 7 | 149.1 | 57.2 |
| $\mu + w_i + \alpha_2 + \beta$ | isotropic (s=1) | 6 | 141.9 | 62.3 |
| $\mu + \alpha_2 + \beta$ | isotropic (s=1) | 5 | 152.5 | 67.1 |
| $\mu + w_i + \alpha_1 + \alpha_2 + \beta$ | isotropic (s=2) | 7 | NA | **43.8** |
| $\mu + w_i + \alpha_2 + \beta$ | isotropic (s=2) | 6 | **6.06** | **53.3** |
| $\mu + \alpha_2 + \beta$ | isotropic (s=2) | 5 | 20.78 | 61.1 |
| $\mu + \alpha_2$ | isotropic (s=2) | 4 | 16.60 | 69 |
| $\mu + w_i + \alpha_1 + \alpha_2 + \beta$ | anisotropic (s=1) | 10 | NA | 46.2 |
| $\mu + w_i + \alpha_2 + \beta$ | anisotropic (s=2) | 9 | NA | 69.8 |
| $\mu + \alpha_2 + \beta$ | anisotropic (s=1) | 8 | NA | 49.3 |
| $\mu + w_i + \alpha_1 + \alpha_2 + \beta$ | anisotropic (s=2) | 10 | NA | **41.5** |
| $\mu + w_i + \alpha_2 + \beta$ | anisotropic (s=2) | 9 | NA | 44.1 |
| $\mu + \alpha_2 + \beta$ | anisotropic (s=2) | 8 | NA | 48.1 |
| $\mu + \alpha_2$ | anisotropic (s=2) | 7 | NA | 55.9 |

For angle $\alpha_2$ the determination of the MSCVE values is not possible for half of the different fitted models. We concentrate on those models where it was possible. Among these models the linear trend component $\mu + w_i + \alpha_2 + \beta$ and isotropic covariance function with $s = 2$ reaches the best BIC and MSCVE values. However, in this model the factors $w_i$ and $\beta$ are not significant. The models that use a reduced trend component have much worse goodness of fit and the larger model is even excluded because its MSCVE value is not available. Therefore, we choose the ordinary Kriging model with isotropic covariance function with $s = 2$ instead. This model has only a little bit worse MSCVE value, a medium sized BIC value and uses the fewest parameters within all models.

**Surrogate models considered for the necking-in ratio $\beta$**

| deterministic trend function | Power exponential covariance function | # parameters | MSCVE | BIC |
|---|---|---|---|---|
| $\mu$ | isotropic (s=1) | 3 | 0.0303 | -20 |
| $\mu$ | isotropic (s=2) | 3 | **0.000124** | -67.9 |
| $\mu$ | anisotropic (s=1) | 6 | 0.001477 | -72.3 |
| $\mu$ | anisotropic (s=2) | 6 | 0.003659 | -66.4 |
| $\mu + w_i + \alpha_1 + \alpha_2 + \beta$ | isotropic (s=1) | 7 | 0.000682 | -66.6 |
| $\mu + w_i + \alpha_2 + \beta$ | isotropic (s=1) | 6 | 0.000712 | -74 |
| $\mu + w_i + \beta$ | isotropic (s=1) | 5 | 0.000682 | -79.5 |
| $\mu + \beta$ | isotropic (s=1) | 4 | 0.000674 | -85.3 |
| $\mu + w_i + \alpha_1 + \alpha_2 + \beta$ | isotropic (s=2) | 7 | NA | -67.8 |
| $\mu + w_i + \alpha_2 + \beta$ | isotropic (s=2) | 6 | 0.000718 | -75 |
| $\mu + w_i + \beta$ | isotropic (s=2) | 5 | 0.000661 | -80.5 |
| $\mu + \beta$ | isotropic (s=2) | 4 | **0.000632** | **-86.5** |
| $\mu + w_i + \alpha_1 + \alpha_2 + \beta$ | anisotropic (s=1) | 10 | NA | -66.1 |
| $\mu + w_i + \alpha_2 + \beta$ | anisotropic (s=1) | 9 | 0.000908 | -75.8 |
| $\mu + w_i + \beta$ | anisotropic (s=1) | 8 | 0.10032 | -81.6 |
| $\mu + w_i + \alpha_1 + \alpha_2 + \beta$ | anisotropic (s=2) | 10 | NA | -72.8 |

The model with the smallest MSCVE value for $\beta$ is the ordinary Kriging model with isotropic covariance function with $s = 2$. This is actually also the simplest model fitted and would be the most preferred one. The Kriging model using the trend component $\mu + \beta$ and same covariance function reaches only a slightly worse MSCV error, but much better BIC value. Since it has only one additional parameter and the estimators of the parameters indicate that the observed necking-in ratio $\beta_{obs}$ is mainly influenced by the input $\beta$ and the effect from the stochastic component is very small, we prefer this model.

## A.5 Progress of candidates and updating points for the different parameterizations of $\alpha$ in Section 4.2

### MOSI

global optimum: 6.47

| $\alpha$'s | cycle | # of design points | currently observed optimum | currently predicted optimum | model fit | $relmax-imax$ | global optimum found? | # of updatings | location of updatings |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 5 | 6 | 8.02 | bad | 0.118 | not found | 2 | ok |
| | 2 | 7 | 6 | 6.64 | better | 0.204 | approaching opt | 3 | ok |
| | 3 | 10 | 6.64 | 6.47 | ok | 0.04 | opt close | 4 | ok |
| | 4 | 14 | 6.47 | 6.47 | ok | 0.025 | opt found | stop | ok |
| 2 | 1 | 5 | 6 | 8.02 | bad | 0.137 | not found | 4 | ok |
| | 2 | 9 | 6 | 6.54 | better | 0.171 | approaching opt | 4 | ok |
| | 3 | 13 | 6.47 | 6.47 | ok | 0.028 | opt found | 3 | ok |
| | 4 | 16 | 6.47 | 6.47 | ok | 0.025 | opt found | stop | ok |
| 3 | 1 | 5 | 6 | 8.02 | bad | 0.207 | not found | 3 | ok |
| | 2 | 8 | 2.75 | 6.5 | better | 0.156 | local opt close | 4 | ok |
| | 3 | 12 | 6.5 | 6.47 | ok | 0.031 | global opt close | 3 | ok |
| | 4 | 15 | 6.47 | 6.47 | ok | 0.025 | opt found | stop | ok |
| 4 | 1 | 5 | 6 | 8.02 | bad | 0.207 | not found | 6 | ok |
| | 2 | 11 | 6.32 | 6.53 | better | 0.072 | approaching opt | 3 | ok |
| | 3 | 14 | 6.53 | 6.47 | ok | 0.03 | opt close | 3 | ok |
| | 4 | 17 | 6.47 | 6.47 | ok | 0.023 | opt found | stop | ok |
| 5 | 1 | 5 | 6 | 8.02 | bad | 0.16 | not found | 7 | ok |
| | 2 | 12 | 6.32 | 6.5 | better | 0.069 | approaching opt | 4 | ok |
| | 3 | 16 | 6.5 | 6.47 | ok | 0.027 | opt close | 3 | ok |
| | 4 | 19 | 6.47 | 6.47 | ok | 0.023 | opt found | stop | ok |
| 6 | 1 | 5 | 6 | 8.02 | bad | 0.207 | not found | 3 | ok |
| | 2 | 8 | 2.75 | 6.63 | better | 0.153 | local opt close | 3 | ok |
| | 3 | 11 | 6.63 | 6.47 | ok | 0.039 | global opt close | 3 | ok |
| | 4 | 14 | 6.47 | 6.47 | ok | 0.028 | opt found | stop | ok |
| 7 | 1 | 5 | 6 | 8.02 | bad | 0.159 | not found | 5 | ok |
| | 2 | 10 | 6.32 | 6.51 | better | 0.076 | approaching opt | 3 | ok |
| | 3 | 13 | 6.51 | 6.47 | ok | 0.029 | opt close | 3 | ok |
| | 4 | 16 | 6.47 | 6.47 | ok | 0.024 | opt found | stop | ok |
| 8 | 1 | 5 | 6 | 8.02 | bad | 0.159 | not found | 4 | ok |
| | 2 | 9 | 2.74 | 6.52 | better | 0.141 | local opt close | 3 | ok |
| | 3 | 12 | 6.52 | 6.47 | ok | 0.032 | global opt close | 4 | ok |
| | 4 | 16 | 6.47 | 6.47 | ok | 0.022 | opt found | stop | ok |

## VLMOP2

global optimum: (0,0)

| α's | cycle | # of design points | currently observed optimum | currently predicted optimum | model fit | $relmax-imax$ | global optimum found? | # of updatings | location of updatings |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 5 | (0.8,1.8) | (0.85,-2) | bad | 0.592 | not found | 3 | clumping |
| 1 | 2 | 8 | (0.9,1.7) | (-0.05,-0.2) | better | 0.419 | approaching opt | 6 | few clumping |
| 1 | 3 | 14 | (-0.05,-0.2) | (0.15,0) | moderate | 0.176 | approaching opt | 6 | few clumping |
| 1 | 4 | 20 | (0.15,0) | (0.1,-0.1) | moderate | 0.133 | opt close | 5 | ok |
| 1 | 5 | 25 | (0.1,-0.1) | (0.05,-0.05) | ok | 0.105 | opt close | stop | |
| 2 | 1 | 5 | (0.8,1.8) | (0.85,-2) | bad | 0.594 | not found | 4 | ok |
| 2 | 2 | 9 | (0,-1.05) | (0.15,-0.15) | better | 0.378 | approaching opt | 6 | clumping |
| 2 | 3 | 15 | (0.15,-0.15) | (0.15,-0.15) | moderate | 0.124 | stagnation | 8 | clumping |
| 2 | 4 | 23 | (0.1,-0.1) | (0.1,-0.1) | ok | 0.108 | opt close | 4 | few clumping |
| 2 | 5 | 27 | (0.1,-0.1) | (0.1,-0.1) | ok | 0.107 | opt close | stop | |
| 3 | 1 | 5 | (0.8,1.8) | (0.85,-2) | bad | 0.595 | not found | 7 | few clumping |
| 3 | 2 | 12 | (-0.05,-0.6) | (0.25,-0.05) | better | 0.317 | approaching opt | 5 | ok |
| 3 | 3 | 17 | (0.25,-0.1) | (0.05,-0.05) | ok | 0.141 | opt close | 4 | good |
| 3 | 4 | 21 | (0.05,-0.05) | (0,0) | ok | 0.091 | opt very close | 6 | good |
| 3 | 5 | 27 | (0,0) | (0,0) | good | 0.074 | opt found | stop | |
| 4 | 1 | 5 | (0.8,1.8) | (0.85,-2) | bad | 0.595 | not found | 5 | few clumping |
| 4 | 2 | 10 | (0.8,1.8) | (-0.05,-0.25) | better | 0.392 | approaching opt | 6 | clumping |
| 4 | 3 | 16 | (0.15,-0.05) | (0.1,-0.1) | ok | 0.134 | opt close | 6 | few clumping |
| 4 | 4 | 22 | (0.1,-0.1) | (0,0) | ok | 0.113 | opt close | 5 | ok |
| 4 | 5 | 27 | (0,0) | (0,0) | good | 0.091 | opt found | stop | |
| 5 | 1 | 5 | (0.8,1.8) | (0.85,-2) | bad | 0.595 | not found | 5 | ok |
| 5 | 2 | 10 | (0,-1.05) | (0.15,-0.2) | better | 0.384 | approaching opt | 6 | few clumping |
| 5 | 3 | 16 | (0.2,-0.1) | (0.15,-0.15) | ok | 0.139 | opt close | 10 | ok |
| 5 | 4 | 26 | (0.05,0.05) | (0,0) | ok | 0.106 | opt very close | 5 | good |
| 5 | 5 | 31 | (0,0) | (0,0) | good | 0.086 | opt found | stop | |
| 6 | 1 | 5 | (0.8,1.8) | (0.85,-2) | bad | 0.595 | not found | 6 | clumping |
| 6 | 2 | 11 | (0,-1.05) | (0.15,-0.15) | better | 0.331 | approaching opt | 6 | few clumping |
| 6 | 3 | 17 | (0.15,-0.15) | (0.05,-0.05) | ok | 0.13 | opt close | 6 | good |
| 6 | 4 | 23 | (0,0) | (0,0) | good | 0.096 | opt found | 9 | ok |
| 6 | 5 | 32 | (0,0) | (0,0) | very good | 0.071 | opt found | stop | |
| 7 | 1 | 5 | (0.8,1.8) | (0.85,-2) | bad | 0.595 | not found | 5 | ok |
| 7 | 2 | 10 | (0.8,1.8) | (0.1,-0.15) | better | 0.364 | approaching opt | 5 | good |
| 7 | 3 | 15 | (0.1,-0.15) | (0.2,-0.15) | ok | 0.132 | opt close | 5 | good |
| 7 | 4 | 20 | (0.1,-0.15) | (0.05,0) | good | 0.114 | opt close | 6 | good |
| 7 | 5 | 26 | (0,0.05) & (0.05,0) | (0,0) | very good | 0.071 | opt very close | 4 | good |
| 7 | 6 | 30 | (0,0) | (0,0) | very good | 0.065 | opt found | stop | |
| 8 | 1 | 5 | (0.8,1.8) | (0.85,-2) | bad | 0.595 | not found | 6 | ok |
| 8 | 2 | 11 | (-0.2,-1.15) | (0.1,-0.2) | ok | 0.361 | approaching opt | 5 | good |
| 8 | 3 | 16 | (0.1,-0.2) | (0.05,0) | good | 0.138 | opt close | 4 | good |
| 8 | 4 | 20 | (0,0) | (-0.05,0.05) | very good | 0.096 | opt very close | 5 | good |
| 8 | 5 | 25 | (0,0) | (0,0) | very good | 0.084 | opt found | stop | |

## VLMOP3

global optima: (0.6,-2.5) & (2.5,0.4)

| α's | cycle | # of design points | currently observed optimum | currently predicted optimum | model fit | $relmax-imax$ | global optima found? | # of updatings | location of updatings |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 5 | (1.8,-1.5) | (0.6,-3) | bad | 5.151 | not found | 2 | clumping |
| 1 | 2 | 7 | (0.6,-3) | (0.6,-1.9) | better | 3.973 | approaching one opt | 5 | clumping |
| 1 | 3 | 12 | (0.6,-3) | (0.5,-2.5) | moderate | 2.776 | one opt close | 7 | few clumping |
| 1 | 4 | 19 | (0.5,-2.5) | (0.5,-2.5) | ok | 0.337 | one opt quasi found | 6 | ok |
| 1 | 5 | 25 | (0.5,-2.5) | (0.6,-2.5) | good | 0.291 | one opt quasi found, second close | stop | |
| 2 | 1 | 5 | (1.8,-1.5) | (0.6,-3) | bad | 5.83 | not found | 5 | few clumping |
| 2 | 2 | 10 | (0.3,-3) | (0.5,-2.4) | better | 2.452 | approaching one opt | 5 | clumping |
| 2 | 3 | 15 | (0.5,-2.4) | (0.5,-2.6) | moderate | 0.856 | one opt close | 6 | few clumping |
| 2 | 4 | 21 | (0.5,-2.6) | (0.5,-2.5) | ok | 0.424 | one opt close | 6 | ok |
| 2 | 5 | 27 | (0.5,-2.5) | (0.6,-2.5) | good | 0.265 | one opt quasi found | 4 | ok |
| 2 | 6 | 31 | (0.6,-2.5) | (0.6,-2.5) | good | 0.219 | one opt found, second close | stop | |
| 3 | 1 | 5 | (1.8,-1.5) | (0.6,-3) | bad | 5.873 | not found | 5 | few clumping |
| 3 | 2 | 10 | (0.6,-3) | (0.5,-2.5) | better | 3.216 | approaching one opt | 8 | ok |
| 3 | 3 | 18 | (0.5,-2.5) | (3,0.5) | moderate | 0.346 | one opt close | 4 | good |
| 3 | 4 | 22 | (0.5,-2.5) | (0.5,-2.5) | ok | 0.318 | one opt quasi found, second close | stop | |
| 4 | 1 | 5 | (1.8,-1.5) | (0.6,-3) | bad | 5.873 | not found | 5 | ok |
| 4 | 2 | 10 | (0.6,-3) | (0.5,-2.2) | better | 3.198 | approaching one opt | 7 | few clumping |
| 4 | 3 | 17 | (0.5,-2.2) | (0.4,-2.6) | moderate | 2.29 | one opt close | 9 | ok |
| 4 | 4 | 26 | (0.4,-2.6) | (0.5,-2.5) | ok | 0.362 | one opt close | 6 | good |
| 4 | 5 | 32 | (0.5,-2.5) | (0.6,-2.5) | good | 0.265 | one opt quasi found | 4 | good |
| 4 | 6 | 36 | (0.6,-2.5) | (0.6,-2.5) | good | 0.222 | one opt found, second close | stop | |
| 5 | 1 | 5 | (1.8,-1.5) | (0.6,-3) | bad | 5.872 | not found | 10 | ok |
| 5 | 2 | 15 | (0.6,-3) | (0.5,-2.5) | better | 2.791 | approaching one opt | 6 | good |
| 5 | 3 | 21 | (0.5,-2.5) | (0.5,-2.5) | ok | 0.308 | one opt close | 6 | good |
| 5 | 4 | 27 | (0.6,-2.5) | (0.6,-2.5) | good | 0.324 | one opt quasi found, second close | stop | |
| 6 | 1 | 5 | (1.8,-1.5) | (0.6,-3) | bad | 5.873 | not found | 8 | ok |
| 6 | 2 | 13 | (0.6,-2.3) | (0.5,-2.6) | better | 2.231 | approaching one opt | 7 | ok |
| 6 | 3 | 20 | (0.5,-2.6) | (0.5,-2.5) | ok | 0.595 | one opt close | 9 | good |
| 6 | 4 | 29 | (0.5,-2.5) | (0.6,-2.5) | ok | 0.336 | one opt quasi found | 5 | good |
| 6 | 5 | 34 | (0.6,-2.5) | (3,0.5) | good | 0.275 | one opt found, second close | stop | |
| 7 | 1 | 5 | (1.8,-1.5) | (0.6,-3) | bad | 5.874 | not found | 7 | ok |
| 7 | 2 | 12 | (0.8,-2.7) | (0.7,-2.3) | better | 2.364 | approaching one opt | 7 | good |
| 7 | 3 | 19 | (0.8,-2.3) | (3,1) | ok | 1.016 | approaching opt | 5 | good |
| 7 | 4 | 24 | (0.7,-2.4) | (0.6,-2.5) | good | 0.446 | one opt close | 4 | very good |
| 7 | 5 | 28 | (0.6,-2.5) | (0.6,-2.5) | very good | 0.278 | one opt found, second close | stop | |
| 8 | 1 | 5 | (1.8,-1.5) | (0.6,-3) | bad | 5.874 | not found | 5 | ok |
| 8 | 2 | 10 | (0.6,-3) | (0.6,-2.4) | better | 3.292 | approaching one opt | 4 | good |
| 8 | 3 | 14 | (0.6,-2.4) | (0.8,-2.4) | ok | 0.671 | one opt close | 5 | good |
| 8 | 4 | 19 | (0.8,-2.4) | (0.6,-2.5) | good | 0.418 | one opt close | 3 | very good |
| 8 | 5 | 22 | (0.6,-2.5) | (2.5,0.3) | very good | 0.257 | one opt found, second close | stop | |

## VLMOP3Bound

global optimum: (2.5,0.4)

| $\alpha$'s | cycle | # of design points | currently observed optimum | currently predicted optimum | model fit | $relmax-imax$ | global optimum found? | # of upda-tings | location of updatings |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 5 | (1.8,-1.5) | (0.7,-2) | bad | 5.17 | not found | 2 | clumping |
| 1 | 2 | 7 | (1.8,-1.5) | (1.4,-1.1) | better | 4.179 | not found | 6 | clumping |
| 1 | 3 | 13 | (1.8,-1.5) | (1.7,-1.3) | moderate | 2.693 | not found | 5 | clumping |
| 1 | 4 | 18 | (1.8,-1.5) | (1.3,-2) | moderate | 2.201 | not found | 5 | clumping |
| 1 | 5 | 23 | (1.3,-2) | (1.2,-2) | ok | 1.228 | not found | 4 | clumping |
| 1 | 6 | 27 | (1.3,-2) | (2.5,0.1) | ok | 1.172 | approaching opt | 4 | ok |
| 1 | 7 | 31 | (2.5,0.1) | (2.5,0.4) | good | 0.474 | opt close | 4 | ok |
| 1 | 8 | 35 | (2.5,0.4) | (2.5,0.4) | good | 0.289 | opt found | stop | |
| 2 | 1 | 5 | (1.8,-1.5) | (0.7,-2) | bad | 5.851 | not found | 5 | few clumping |
| 2 | 2 | 10 | (1.8,-1.5) | (1.8,-0.7) | better | 3.332 | not found | 8 | clumping |
| 2 | 3 | 18 | (1.8,-1.5) | (1.3,-2) | moderate | 2.178 | not found | 5 | clumping |
| 2 | 4 | 23 | (1.3,-2) | (1.2,-2) | moderate | 1.281 | not found | 5 | few clumping |
| 2 | 5 | 28 | (1.3,-2) | (2.5,0.2) | ok | 1.226 | approaching opt | 4 | ok |
| 2 | 6 | 32 | (2.5,0.2) | (2.5,0.4) | ok | 0.329 | opt close | 4 | ok |
| 2 | 7 | 36 | (2.5,0.4) | (2.5,0.4) | good | 0.266 | opt found | stop | |
| 3 | 1 | 5 | (1.8,-1.5) | (0.7,-2) | bad | 5.898 | not found | 5 | ok |
| 3 | 2 | 10 | (1.8,-1.5) | (1.7,-1.2) | better | 5.187 | not found | 8 | ok |
| 3 | 3 | 18 | (1.8,-1.5) | (1.5,-1.6) | moderate | 2.65 | not found | 5 | good |
| 3 | 4 | 23 | (1.8,-1.5) | (2.5,0.2) | moderate | 2.127 | approaching opt | 5 | good |
| 3 | 5 | 28 | (2.5,0.2) | (2.5,0.2) | ok | 0.325 | opt close | 4 | good |
| 3 | 6 | 32 | (2.5,0.2) | (2.5,0.2) | ok | 0.308 | opt close | stop | |
| 4 | 1 | 5 | (1.8,-1.5) | (0.7,-2) | bad | 5.898 | not found | 5 | ok |
| 4 | 2 | 10 | (1.8,-1.5) | (1.8,-1.1) | better | 3.376 | not found | 6 | few clumping |
| 4 | 3 | 16 | (1.8,-1.5) | (2.5,-0.3) | moderate | 2.183 | approaching opt | 8 | ok |
| 4 | 4 | 24 | (2.5,-0.3) | (2.5,0) | ok | 0.943 | approaching opt | 4 | good |
| 4 | 5 | 28 | (2.5,0) | (2.5,0.2) | good | 0.427 | opt close | 4 | good |
| 4 | 6 | 32 | (2.5,0.2) | (2.5,0.2) | good | 0.269 | opt close | 4 | very good |
| 4 | 7 | 36 | (2.5,0.2) | (2.5,0.2) | good | 0.258 | opt close | stop | |
| 5 | 1 | 5 | (1.8,-1.5) | (0.7,-2) | bad | 5.898 | not found | 5 | ok |
| 5 | 2 | 10 | (1.8,-1.5) | (1.6,-1.1) | better | 3.512 | not found | 10 | few clumping |
| 5 | 3 | 20 | (2.5,0.6) | (2.5,0.4) | moderate | 0.379 | opt close | 5 | good |
| 5 | 4 | 25 | (2.5,0.4) | (2.5,0.4) | ok | 0.284 | opt found | 3 | good |
| 5 | 5 | 28 | (2.5,0.4) | (2.5,0.4) | good | 0.267 | opt found | stop | |
| 6 | 1 | 5 | (1.8,-1.5) | (0.7,-2) | bad | 5.898 | not found | 5 | ok |
| 6 | 2 | 10 | (1.8,-1.5) | (1.7,-1.2) | better | 5.243 | not found | 6 | ok |
| 6 | 3 | 16 | (1.8,-1.5) | (1.6,-1.6) | moderate | 3.295 | not found | 5 | good |
| 6 | 4 | 21 | (1.8,-1.5) | (1.3,-1.8) | moderate | 2.715 | not found | 8 | good |
| 6 | 5 | 29 | (1.3,-1.8) | (2.5,0.3) | ok | 1.597 | approaching opt | 8 | very good |
| 6 | 6 | 37 | (2.5,0.3) | (2.5,0.3) | good | 0.243 | opt quasi found | 3 | very good |
| 6 | 7 | 40 | (2.5,0.3) | (2.5,0.3) | good | 0.233 | opt quasi found | stop | |
| 7 | 1 | 5 | (1.8,-1.5) | (0.7,-2) | bad | 5.898 | not found | 5 | ok |
| 7 | 2 | 10 | (1.8,-1.5) | (1.6,-1.2) | better | 3.456 | not found | 5 | ok |
| 7 | 3 | 15 | (1.8,-1.5) | (1.4,-1.8) | moderate | 2.291 | not found | 4 | good |
| 7 | 4 | 19 | (1.4,-1.8) | (2.5,-0.3) | ok | 1.805 | approaching opt | 4 | good |
| 7 | 5 | 23 | (2.5,-0.3) | (2.5,0.3) | ok | 0.862 | approaching opt | 4 | very good |
| 7 | 6 | 27 | (2.5,0.3) | (2.5,0.3) | good | 0.269 | opt quasi found | 4 | very good |
| 7 | 7 | 31 | (2.5,0.3) | (2.5,0.3) | good | 0.254 | opt quasi found | stop | |
| 8 | 1 | 5 | (1.8,-1.5) | (0.7,-2) | bad | 5.898 | not found | 5 | few clumping |
| 8 | 2 | 10 | (1.8,-1.5) | (1.5,-1.2) | better | 3.381 | not found | 8 | very good |
| 8 | 3 | 18 | (1.8,-1.5) | (2.5,0.3) | ok | 2.062 | approaching opt | 4 | very good |
| 8 | 4 | 22 | (2.5,0.3) | (2.5,0.3) | ok | 0.326 | opt quasi found | 3 | very good |
| 8 | 5 | 25 | (2.5,0.3) | (2.5,0.3) | ok | 0.316 | opt quasi found | stop | |

## DTLZ1

global optima: (0.8,0.1) & (0.8,0.9)

| $\alpha$'s | cycle | # of design points | currently observed optimum | currently predicted optimum | model fit | $relmax-imax$ | global optima found? | # of updatings | location of updatings |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 4 | (1,0.4) | (0.8,0.6) | bad | 2.993 | not found | 2 | ok |
| 1 | 2 | 6 | (1,0.4) | (0.9,0) | bad | 2.836 | approaching 1.opt | 7 | ok |
| 1 | 3 | 13 | (1,0.4) | (0.7,0.2) | moderate | 3.139 | 1.opt close | 5 | ok |
| 1 | 4 | 18 | (0.8,0) | (0.8,1) | moderate | 3.085 | 1.opt quasi found | 4 | ok |
| 1 | 5 | 22 | (0.9,0.1) | (0.8,0.1) | ok | 2.385 | 1.opt quasi found | 6 | few clumping |
| 1 | 6 | 28 | (0.8,0.1) | (0.8,0.1) | good | 0.131 | 1.opt found | 2 | few clumping |
| 1 | 7 | 30 | (0.8,0.1) | (0.8,0.1) | good | 0.112 | 1.opt found, 2.opt close | stop | |
| 2 | 1 | 4 | (1,0.4) | (0.8,0.6) | bad | 2.996 | not found | 6 | good |
| 2 | 2 | 10 | (0.9,0.2) | (0.8,0.8) | moderate | 0.615 | 1.opt close | 5 | good |
| 2 | 3 | 15 | (0.9, 0.2)& (0.9, 0.8) | 0.8,0.2 | ok | 0.539 | 1.and 2. opt close | 9 | ok |
| 2 | 4 | 24 | (0.8,0.1) | (0.8,0.1) | good | 0.136 | 1. opt found | 4 | good |
| 2 | 5 | 28 | (0.8,0.1) | (0.8,0.1) | good | 0.16 | 1.opt found, 2.opt quasi | stop | |
| 3 | 1 | 4 | (1,0.4) | (0.8,0.6) | bad | 2.993 | not found | 6 | few clumping |
| 3 | 2 | 10 | (0.9,0.4) | (0.9,0.2) | moderate | 1.02 | approaching 1.opt | 7 | good |
| 3 | 3 | 17 | (0.8,0.1) | (0.8,0.8) | ok | 0.18 | 1.opt close | 5 | good |
| 3 | 4 | 22 | (0.8,0.1) | (0.8,0.1) | good | 0.175 | 1.opt found, 2.opt quasi | stop | |
| 4 | 1 | 4 | (1,0.4) | (0.8,0.6) | bad | 2.993 | not found | 6 | few clumping |
| 4 | 2 | 10 | (0.9,0.4) | (0.7,0.9) | moderate | 1.464 | approaching 1.opt | 9 | clumping |
| 4 | 3 | 19 | (0.9, 0.4)& (0.9, 0.6) | 0.7,0.1 | ok | 1.245 | approaching 1.opt | 6 | few clumping |
| 4 | 4 | 25 | (0.9,0.7) | (0.8,0.1) | good | 0.552 | 1.opt close | 4 | ok |
| 4 | 5 | 29 | (0.8,0.1) | (0.8,0.1) | very good | 0.213 | 1.opt found | 7 | ok |
| 4 | 6 | 36 | (0.8,0.1) | (0.8,0.1) | very good | 0.158 | 1.opt found | 5 | ok |
| 4 | 7 | 41 | (0.8, 0.1)& (0.8, 0.9) | (0.8, 0.8) (0.8,0.9) | very good | 0.18 | 1.and 2.opt found | stop | |
| 5 | 1 | 4 | (1,0.4) | (0.8,0.6) | bad | 2.996 | not found | 5 | ok |
| 5 | 2 | 9 | (0.9,0.4) | (0.8,0.8) | moderate | 1.251 | approaching 1.opt | 8 | good |
| 5 | 3 | 17 | (0.9,0.2) | (0.8,0.2) | ok | 0.59 | 1.opt close | 6 | ok |
| 5 | 4 | 23 | (0.9,0.3) | (0.8,0.9) | good | 0.405 | 1.opt close | 5 | ok |
| 5 | 5 | 28 | (0.8, 0.1)& (0.8, 0.9) | (0.8, 0.1)& (0.8,0.9) | good | 0.147 | 1.and 2.opt found | 3 | good |
| 5 | 6 | 31 | (0.8, 0.1)& (0.8, 0.9) | (0.8, 0.1)& (0.8,0.9) | very good | 0.142 | 1.and 2.opt found | stop | |
| 6 | 1 | 4 | (1,0.4) | (0.8,0.6) | bad | 2.996 | not found | 6 | ok |
| 6 | 2 | 10 | (1,0.4) | (0.8,0.8) | moderate | 3.117 | approaching 1.opt | 7 | few clumping |
| 6 | 3 | 17 | (0.9, 0.8)& (0.9, 0.2) | 0.8,0.1 | ok | 0.582 | both opts close | 8 | ok |
| 6 | 4 | 25 | (0.8,0.1) | (0.8,0.1) | good | 0.166 | 1.opt found | 4 | ok |
| 6 | 5 | 29 | (0.8,0.1) | (0.8,0.1) | very good | 0.138 | 1.opt found, 2.opt quasi | stop | |
| 7 | 1 | 4 | (1,0.4) | (0.8,0.6) | bad | 3.001 | not found | 7 | ok |
| 7 | 2 | 11 | (1,0.5) | (0.8,0.8) | moderate | 2.743 | approaching 1.opt | 7 | ok |
| 7 | 3 | 18 | (0.8,0.9) | (0.8,0.1) | ok | 0.18 | 1.opt found | 3 | good |
| 7 | 4 | 21 | (0.8, 0.1)& (0.8, 0.9) | (0.8, 0.1)& (0.8,0.9) | good | 0.169 | 1.and 2.opt found | stop | |
| 8 | 1 | 4 | (1,0.4) | (0.8,0.6) | bad | 3.001 | not found | 5 | ok |
| 8 | 2 | 9 | (0.9,0.6) | (0.8,0.2) | bad | 0.999 | approaching 1.opt | 12 | good |
| 8 | 3 | 21 | (0.8,0.1) | (0.8,0.8) | ok | 0.192 | 1.opt found | 3 | good |
| 8 | 4 | 24 | (0.8,0.1) | (0.8,0.1) | ok | 0.189 | 1.opt found, 2.opt quasi | stop | |

## DTLZ3

global optimum: (0.1,0.9)

| $\alpha$'s | cycle | # of design points | currently observed optimum | currently predicted optimum | model fit | $relmax-imax$ | global optimum found? | # of upda-tings | location of updatings |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 4 | (0.5,0.9) | (0.3,0.9) | moderate | 6.018 | not found | 4 | ok |
| 1 | 2 | 8 | (0.3,0.9) | (0.2,0.8) | moderate | 6.776 | approaching opt | 5 | clumping |
| 1 | 3 | 13 | (0.2,0.8) | (0.2,0.8) | ok | 0.403 | local opt found | 3 | clumping |
| 1 | 4 | 16 | (0.2,0.8) | (0.2,0.8) | ok | 0.32 | local opt found, shape ok | stop | |
| 2 | 1 | 4 | (0.5,0.9) | (0.3,0.9) | moderate | 7.422 | not found | 7 | few clumping |
| 2 | 2 | 11 | (0.3,0.9) | (0.2,0.8) | moderate | 5.958 | approaching opt | 5 | few clumping |
| 2 | 3 | 16 | (0.2,0.8) | (0.3,0.6) | ok | 0.367 | local opt found | 6 | good |
| 2 | 4 | 22 | (0.2,0.8) | (0.5,0.4) | good | 0.415 | local opt found | 4 | ok |
| 2 | 5 | 26 | (0.2,0.8) | (0.5,0.3) | good | 0.375 | local opt found shape ok | stop | |
| 3 | 1 | 4 | (0.5,0.9) | (0.3,0.9) | moderate | 9.087 | not found | 5 | ok |
| 3 | 2 | 9 | (0.1,0.8) | (0.3,0.7) | moderate | 3.354 | opt close | 6 | ok |
| 3 | 3 | 15 | (0.2,0.8) | (0.5,0.3) | ok | 0.382 | local opt found | 6 | good |
| 3 | 4 | 21 | (0.2,0.8) | (0.2,0.8) | good | 0.395 | local opt found shape good | stop | |
| 4 | 1 | 4 | (0.5,0.9) | (0.3,0.9) | moderate | 9.087 | not found | 7 | good |
| 4 | 2 | 11 | (0.5,0.3) | (0.3,0.7) | ok | 3.026 | approaching opt | 7 | ok |
| 4 | 3 | 18 | (0.3,0.7) | (0.2,0.8) | ok | 1.225 | local opt found shape ok | stop | |
| 5 | 1 | 4 | (0.5,0.9) | (0.3,0.9) | moderate | 9.07 | not found | 10 | good |
| 5 | 2 | 14 | (0.1,0.7) | (0.4,0.4) | ok | 8.032 | approaching opt | 6 | good |
| 5 | 3 | 20 | (0.3,0.6) | (0.4,0.3) | good | 3.714 | opt close | 10 | ok |
| 5 | 4 | 30 | (0.2,0.7) | (0,1) | very good | 0.614 | opt close | 5 | ok |
| 5 | 5 | 35 | (0.2,0.7) | (0.1,0.9) | very good | 0.625 | opt found shape good | stop | |
| 6 | 1 | 4 | (0.5,0.9) | (0.3,0.9) | moderate | 9.087 | not found | 5 | few clumping |
| 6 | 2 | 9 | (0.2,0.9) | (0.3,0.7) | moderate | 3.359 | global opt quasi found | 5 | few clumping |
| 6 | 3 | 14 | (0.3,0.7) | (0.4,0.5) | moderate | 0.918 | approaching local opt | 6 | ok |
| 6 | 4 | 20 | (0.2,0.8) | (0.2,0.8) | ok | 0.397 | local opt found | 5 | ok |
| 6 | 5 | 25 | (0.1,0.9) | (0.1,0.9) | ok | 0.012 | global opt found | 3 | ok |
| 6 | 6 | 28 | (0.1,0.9) | (0.1,0.9) | good | 0.009 | global opt found shape very good | stop | |
| 7 | 1 | 4 | (0.5,0.9) | (0.3,0.9) | moderate | 9.07 | not found | 6 | few clumping |
| 7 | 2 | 10 | (0.1,0.7) | (0.2,0.8) | moderate | 6.849 | global opt quasi found | 6 | good |
| 7 | 3 | 16 | (0.2,0.8) | (0.2,0.8) | ok | 0.508 | local opt found | 5 | good |
| 7 | 4 | 21 | (0.2,0.8) | (0.5,0.3) | ok | 0.44 | local opt found | 4 | ok |
| 7 | 5 | 25 | (0.2,0.8) | (0.2,0.8) | good | 0.338 | local opt found shape good | stop | |
| 8 | 1 | 4 | (0.5,0.9) | (0.3,0.9) | moderate | 9.075 | not found | 10 | ok |
| 8 | 2 | 14 | (0.2,0.9) | (0.4,0.4) | ok | 4.808 | global opt quasi found | 5 | good |
| 8 | 3 | 19 | (0.2,0.8) | (0.4,0.3) | ok | 0.551 | local opt found | 4 | good |
| 8 | 4 | 23 | (0.1,0.9) | (0.1,0.9) | good | 0.012 | global opt found | stop | |

## KNO

global optima: (0.4,1.4) & (1.4,2.4)

| $\alpha$'s | cycle | # of design points | currently observed optimum | currently predicted optimum | model fit | $relmax-imax$ | global optima found? | # of upda-tings | location of updatings |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 4 | (0.7,2.2) | (1.1,2.1) | bad | 1.721 | not found | 1 | ok |
| 1 | 2 | 5 | (0.7,2.2) | (1.1,2.7) | better | 1.492 | approaching 1.opt | 1 | ok |
| 1 | 3 | 6 | (1.1,2.7) | (0.1,1.2) | moderate | 0.135 | approaching 2.opt | 2 | ok |
| 1 | 4 | 8 | (1.1,2.7) | (0.6,1.7) | ok | 0.11 | both opts not found, shape ok | stop | |
| 2 | 1 | 4 | (0.7,2.2) | (1.1,2.1) | bad | 1.723 | not found | 5 | ok |
| 2 | 2 | 9 | (0.5,2.1) | (1.3,2.8) | better | 0.481 | approaching 1.opt | 6 | ok |
| 2 | 3 | 15 | (0.5,2.1) | (0.3,1.4) | moderate | 0.404 | approaching 1.opt | 5 | few clumping |
| 2 | 4 | 20 | (2.3,1.5) | (1.4,2.4) | ok | 0.077 | approaching 2.opt | 5 | few clumping |
| 2 | 5 | 25 | (1.4,2.4) | (0.5,1.3) | ok | 0.004 | 2.opt found | 5 | ok |
| 2 | 6 | 30 | (1.4,2.4) | (1.4,2.4) | ok | 0.004 | 2.opt found, 1.opt quasi shape ok | stop | |
| 3 | 1 | 4 | (0.7,2.2) | (1.1,2.1) | bad | 1.724 | not found | 5 | clumping |
| 3 | 2 | 9 | (1,0.7) | (0.1, 0.9) (0.3,1.1) | better | 0.208 | approaching 1.opt | 6 | clumping |
| 3 | 3 | 15 | (1.1,0.7) | (1.3,2.6) | moderate | 0.076 | approaching 2.opt | 9 | few clumping |
| 3 | 4 | 24 | (1.1,0.7) | (1.4,2.4) | ok | 0.071 | 2.opt found, 1.opt close, shape ok | stop | |
| 4 | 1 | 4 | (0.7,2.2) | (1.1,2.1) | bad | 1.721 | not found | 5 | few clumping |
| 4 | 2 | 9 | (1.3,0.6) | (0.1,1.1) | better | 0.496 | approaching 1.opt | 5 | few clumping |
| 4 | 3 | 14 | (1.1,0.6) | (0.7,1.6) | moderate | 0.192 | 1.opt close | 7 | few clumping |
| 4 | 4 | 21 | (1.1,0.7) | (1.5, 2.8) (1.5, 2.9) | ok | 0.067 | approaching 2.opt | 7 | few clumping |
| 4 | 5 | 28 | (1.1,0.7) | (1.6,2.1) | good | 0.071 | 2 local opts close to global opts, shape ok | stop | |
| 5 | 1 | 4 | (0.7,2.2) | (1.1,2.1) | bad | 1.724 | not found | 5 | few clumping |
| 5 | 2 | 9 | (1.1,0.7) | (1.3,2.7) | better | 0.092 | not found | 5 | ok |
| 5 | 3 | 14 | (1.1,0.7) | (1.4,2.3) | moderate | 0.088 | approaching 1.opt | 6 | ok |
| 5 | 4 | 20 | (1.5,2.3) | (1.5,2.3) | moderate | 0.004 | 1.opt close | 6 | ok |
| 5 | 5 | 26 | (1.5,2.3) | (0.6,1.4) | ok | 0.004 | both opts close, shape ok | stop | |
| 6 | 1 | 4 | (0.7,2.2) | (1.1,2.1) | bad | 1.724 | not found | 5 | clumping |
| 6 | 2 | 9 | (0.7,2.2) | (0.1,1.2) | better | 1.036 | approaching 2.opt | 8 | few clumping |
| 6 | 3 | 17 | (2.5,2.6) | (0.6,1.6) | moderate | 0.086 | approaching 1.opt | 10 | ok |
| 6 | 4 | 27 | (2.5,2.6) | (2.5,3) | ok | 0.078 | one local opt close to global opt, shape ok | stop | |
| 7 | 1 | 4 | (0.7,2.2) | (1.1,2.1) | bad | 1.724 | not found | 9 | ok |
| 7 | 2 | 13 | (0.8,1.9) | (0.5,1.4) | better | 0.367 | approaching 1.opt | 6 | good |
| 7 | 3 | 19 | (0.4,1.3) | (1.6,2.7) | ok | 0.146 | 1.opt close | 9 | good |
| 7 | 4 | 28 | (1.2,0.6) | (0.4,1.4) | ok | 0.084 | approaching 2.opt | 5 | good |
| 7 | 5 | 33 | (0.4,1.4) | (0.1,0.9) | good | 0.018 | 1.opt found | 6 | good |
| 7 | 6 | 39 | (0.4,1.4) | (1.5,2.3) | good | 0.018 | 1.opt found,2.opt close, shape good | stop | |
| 8 | 1 | 4 | (0.7,2.2) | (1.1,2.1) | bad | 1.724 | not found | 6 | few clumping |
| 8 | 2 | 10 | (0.7,2.1) | (0.3,1.4) | better | 0.578 | approaching 1.opt | 5 | ok |
| 8 | 3 | 15 | (0.3,1.4) | (0.7,1.3) | moderate | 0.181 | 1.opt close | 8 | good |
| 8 | 4 | 23 | (1.5,2.3) | (1.4,2.4) | ok | 0.005 | approaching 2.opt | 6 | good |
| 8 | 5 | 29 | (1.4,2.4) | (1.4,2.4) | good | 0.004 | 2.opt found, 1.opt quasi found, shape good | stop | |

**MEINE**

global optima: (0.6,1.1) & (2.2,2.65)

| $\alpha$'s | cy-cle | # of design points | currently observed optimum | currently predicted optimum | model fit | $relmax-imax$ | global optima found? | # of upda-tings | location of updatings |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 5 | (1.5,2.25) | (0.75,1.25) | moderate | 1.154 | not found | 5 | ok |
| 1 | 2 | 10 | (0.75,1.25) | (0.65,1.15) | moderate | 0.043 | approaching 1.opt | 5 | good |
| 1 | 3 | 15 | (0.65,1.15) | (0.65,1.15) | ok | 0.008 | 1.opt quasi found | 5 | ok |
| 1 | 4 | 20 | (0.65,1.15) | (0.65,1.15) | ok | 0.008 | 1.opt quasi found,2.opt close, shape ok | stop | |
| 2 | 1 | 5 | (1.5,2.25) | (0.75,1.25) | moderate | 1.157 | not found | 6 | good |
| 2 | 2 | 11 | (0.75,1.25) | (0.65,1.15) | moderate | 0.033 | approaching 1.opt | 10 | ok |
| 2 | 3 | 21 | (0.65,1.15) | (0.65,1.15) | ok | 0.011 | 1.opt quasi found | 5 | good |
| 2 | 4 | 26 | (0.6,1.1) | (0.6,1.1) | good | 0 | 1.opt found, 2.opt quasi found, shape good | stop | |
| 3 | 1 | 5 | (1.5,2.25) | (0.75,1.25) | moderate | 1.157 | not found | 7 | good |
| 3 | 2 | 12 | (0.75,1.25) | (0.15,0.1) | ok | 0.044 | approaching 1.opt | 7 | ok |
| 3 | 3 | 19 | (0.55,1.05) | (0.65,1.15) | ok | 0.02 | 1.opt close | 6 | few clumping |
| 3 | 4 | 25 | (0.65,1.15) | (0.6,1.1) | good | 0.007 | 1.opt quasi found | 3 | good |
| 3 | 5 | 28 | (0.6,1.1) | (0.6,1.1) | very good | 0.002 | 1.opt found, 2.opt quasi found, shape very good | stop | |
| 4 | 1 | 5 | (1.5,2.25) | (0.75,1.25) | moderate | 1.157 | not found | 7 | few clumping |
| 4 | 2 | 12 | (0.75,1.25) | (0.7,1.2) | moderate | 0.065 | approaching 1.opt | 7 | few clumping |
| 4 | 3 | 19 | (0.7,1.2) | (0.65,1.15) | ok | 0.028 | 1.opt close | 4 | good |
| 4 | 4 | 23 | (0.65,1.15) | (0.65,1.15) | good | 0.008 | 1.opt quasi found | 7 | ok |
| 4 | 5 | 30 | (0.65,1.15) | (0.6,1.1) | very good | 0.012 | both opts quasi found, shape very good | stop | |
| 5 | 1 | 5 | (1.5,2.25) | (0.75,1.25) | moderate | 1.157 | not found | 6 | good |
| 5 | 2 | 11 | (0.75,1.25) | (2.05,2.55) | ok | 0.076 | approaching 1.opt | 3 | ok |
| 5 | 3 | 14 | (0.6,1.1) | (0.65,1.15) | ok | 0.011 | 1.opt found | 4 | very good |
| 5 | 4 | 18 | (0.6,1.1) | (0.6,1.1) | good | 0.01 | 1.opt found, 2.opt quasi found, shape good | stop | |
| 6 | 1 | 5 | (1.5,2.25) | (0.75,1.25) | moderate | 1.158 | not found | 8 | ok |
| 6 | 2 | 13 | (0.65,1.2) | (0.65,1.15) | ok | 0.047 | 1.opt close | 5 | good |
| 6 | 3 | 18 | (0.65,1.15) | (0.6,1.1) | good | 0.013 | 1.opt quasi found | 5 | ok |
| 6 | 4 | 23 | (0.6,1.1) | (0.6,1.1) | very good | 0.01 | 1.opt found, 2.opt quasi found, shape very good | stop | |
| 7 | 1 | 5 | (1.5,2.25) | (0.75,1.25) | moderate | 1.158 | not found | 7 | ok |
| 7 | 2 | 12 | (0.75,1.25) | (0.2,0.1) | moderate | 0.031 | 1.opt close | 8 | good |
| 7 | 3 | 20 | (0.55,1.05) | (0.6,1.1) | ok | 0.016 | 1.opt close | 7 | few clumping |
| 7 | 4 | 27 | (0.6,1.1) | (0.6,1.1) | good | 0 | 1.opt found, 2.opt quasi found, shape very good | stop | |
| 8 | 1 | 5 | (1.5,2.25) | (0.75,1.25) | moderate | 1.158 | not found | 6 | good |
| 8 | 2 | 11 | (0.75,1.25) | (0.15,0.1) | ok | 0.037 | 1.opt close | 6 | good |
| 8 | 3 | 17 | (0.75,1.25) | (0.65,1.15) | ok | 0.046 | 1.opt close | 3 | very good |
| 8 | 4 | 20 | (0.6,1.1) | (0.6,1.1) | ok | 0.007 | 1.opt found | 3 | good |
| 8 | 5 | 23 | (0.6,1.1) | (0.6,1.1) | good | 0.006 | 1.opt found, 2.opt close, shape ok | stop | |

## A.6 Implementation of mtEGO in R

### Exemplary R code for Step 1 of mtEGO

(including the missing value extension)

```
## example how to build a 3-dimensional grid representing the parameter space
x1 <- x2 <- x3 <- c(0,2,4,6,8)
library(gtools)
kombis <- permutations(5,3,repeats.allowed=TRUE)
Grid <- data.frame(x1=x1[kombis[,1]], x2=x2[kombis[,2]], x3=x3[kombis[,3]])
## generate a space-filling design
library(fields)
design <- cover.design(Grid, 15, nruns=10)[1:15,]
## restrict parameter space if missing values occur
library(rPorta)
restrictedGrid <- failureRegions(as.poi(design),as.poi(Grid),failures)
restricted <- as.matrix(getFeasiblePoints(restrictedGrid))
## add more points to the existing space-filling design
old <- as.matrix(design[which(design == failure),])
kombi <- rbind(old, restrictedGrid)
fixrows <- which(duplicated(kombi)==TRUE)-length(old[,1])
rownames(restrictedGrid) <- 1:length(restrictedGrid[,1])
newDesignpoints <- cover.design(restrictedGrid, 5 ,nn=FALSE, fixed=fixrows)[1:5,]
```

### Exemplary SAS code for Step 2 of mtEGO

('C:/mtEGO-Vlmop2Two-auto.sas')

```
DATA daten;
INFILE 'C:\auto-vp.txt' FIRSTOBS=2;
INPUT id $ x1 x2 y1 y2 w1 w2 wges;
RUN;

DATA punkte;
INFILE 'C:\testproblem-paramspace.txt' FIRSTOBS=2;
INPUT id $ x1 x2;
RUN;

DATA all;
SET daten punkte;
RUN;

PROC MIXED DATA=all method=reml scoring=50 convh=1e-8 alpha=0.05 asycov noinfo noitprint;
model y1=x1 x2/ solution outpred=predicted;
repeated /type=SP(exp)(x1 x2) subject=intercept;
RUN;QUIT;

DATA t1;
SET predicted;
IF y1 NE . THEN DELETE;
```

```
py1=pred;
s1=stderrpred;
d1 = DF;
RUN;

PROC MIXED DATA=all method=reml  scoring=50 convh=1e-8 alpha=0.05 asycov noinfo noitprint;
model y2=x1 x2/ solution outpred=predicted;
repeated /type=SP(exp)(x1 x2) subject=intercept;
RUN;QUIT;

DATA t2;
SET predicted;
IF y2 NE . THEN DELETE;
py2=pred;
s2=stderrpred;
d2=DF;
RUN;

DATA t_final;
MERGE t1 t2;
KEEP x1 x2 py1 py2 s1 s2 d1 d2;
RUN;

DATA t_final2;
SET t_final;
predy1=py1;
predy2=py2;
sy1=s1;
sy2=s2;
df1=d1;
df2=d2;
RUN;

DATA t_final3;
SET t_final2;
KEEP x1 x2 predy1 predy2 sy1 sy2 df1 df2;
RUN;

PROC EXPORT DATA= WORK.t_final3
            OUTFILE= "C:\auto-model.csv"
            DBMS=CSV REPLACE;
RUN;

# Code to run the sas-model-fitting procedure from R
# vpw -- the design which shall be modeled
# sasprog -- the path of the sas programme with the model fittign code that shall be run
# sascmd -- the path of the sas.exe file, to tell R where to search the sas-programme
# vpwfile -- file-path of the file where the current design is to be stored temporarily
# modelfile -- file-path of the file where the fitted model is to be stored temporarily
sasmodelfit <- function(vpw, sasprog="C:/SAScode.sas", sascmd="C:/Programme/SAS/SAS 9.1/sas.exe",
                        vpwfile="C:/design.txt", modelfile="C:/outputfile.csv")
```

```
{ write.table(vpw, vpwfile)
  sascmds <- paste(shQuote(sascmd), "-sysin")
  sasrun <- try(sysret <- system(paste(sascmds, sasprog)))
  if (inherits(sasrun, "try-error") | sysret < 0)
  { cat("SAS Sytem error - view log files")
  }else{
  sasmodel <- read.table(modelfile, sep=",", header=TRUE)
 }
 for(j in 1:odim)
  {  a <- which(is.na(sasmodel[,idim+odim+j]))
     if(length(a) != 0) sasmodel[a,idim+odim+j] <- 0
  }
  return(sasmodel)
}
```

## Code for Step 3.A and 3.B and 4 of mtEGO

```
# desifun(), wimaxcorrfun(), allowrepeatfun(), desvirtobsfun() are subroutines for the
# functions mtEGO() and mtEGOeff()

### function to transform any objective y to desirabilities
# y -- the objective to be transformed to desirabilities
# desiparams -- a list with one entry per objective specifying the desirability function
#               Each entry starts with the number
#               1 for "one-sided desirability",
#               2 for "Harrington two-sided desirability"
#               or 3 for "Derringer-suich two-sided desirabilities"
#               the specification parameters of the desirabilities are given in the following order:
#               for one-sided: 1, b0, b1
#               for two-sided Harrington: 2, LSL, target, USL, nu
#               for two-sided Derringer: 3, LSL, target, USL, l, r
desifun <- function(y, desiparams)
{desis <- NULL
 if(desiparams[1]==1)
 desis <- round(exp(-exp(-(desiparams[2]+desiparams[3]*y))), digits=4)
 if(desiparams[1]==2)
 desis <- round(exp(-1*abs((y - (desiparams[4] + desiparams[2])/2 )
              /((desiparams[4] - desiparams[2])/2))^desiparams[5]), digits=4)
 if(desiparams[1]==3)
 {desis <- numeric(length(y))
  h1 <- which(y < desiparams[2])
  h2 <- which(y >= desiparams[2] & y <= desiparams[3])
  h3 <- which(y > desiparams[3] & y <= desiparams[4])
  h4 <- which(y > desiparams[4])
  if(length(h1) != 0)
  desis[h1] <- 0
  if(length(h2) != 0)
  desis[h2] <- ((y[h2] - desiparams[2])/(desiparams[3] - desiparams[2]))^desiparams[5]
  if(length(h3) != 0)
  desis[h3] <- ((y[h3] - desiparams[4])/(desiparams[3] - desiparams[4]))^desiparams[6]
  if(length(h4) != 0)
  desis[h4] <- 0
 }
 return(desis)
}

#For the improvement the current maximal desirability (wimax) is needed. If this value is smaller in
#the prediction compared to the measured one, since the model does not interpolate exactly, a correction
#of the wimax value is needed; wimax then is the currently maximum desirability among the predictions
wimaxcorrfun <- function(vp1w, models, desiparams)
{wimax <- max(vp1w$wges)
 maxicand <- vp1w[which(vp1w$wges == wimax)[1],1:idim]
 indges <- NULL
 trues <- rep(NA,dim(models)[1])
 for(k in 1:idim)
 { for(z in 1:dim(models)[1])
   {trues[z] <- all.equal(models[z,k],as.numeric(maxicand[k])) }
   indges <- c(indges, which(trues==TRUE))
 }
 ind <- as.numeric(names(which(table(indges)==idim)))
 ws <-rep(NA, odim)
 for(f in 1:odim){ ws[f] <- desifun(models[ind,idim+f], desiparams[[f]]) }
 wimaxcorr <- prod(ws^weightedDI)
 if(wimaxcorr > wimax) wimaxcorr = wimax
 return(wimaxcorr)
}

#set prediction error s=0 for observations to force the improvement to be 0
allowrepeatfun <- function(vp1w, sda, model1)
{samples <- data.frame(dummy=1:dim(vp1w)[1],vp1w[,1:idim])
 for(j in 1:dim(vp1w)[1])
 { for(k in 1:dim(model1)[1])
   { if(all(model1[k,1:idim] == samples[j,-1]) == TRUE) sda[k,] <- rep(0,(odim+1))
   }
 }
 return(sda)
}

#function to calculate the virtual observations and their corresponding desirabilities for a given
#\alpha vector and a set of predicted values
desvirtobsfun <- function(alphafinepart, ydpart, sdapart, dfspart, k)
{desvirtobs <- list((odim))
 desvirtob <- matrix(rep(NA,length(ydpart)*length(alphafinepart)),nrow=length(ydpart))
 for(h in 1:length(alphafinepart))
 {if(alphafinepart[h] != 1)
  {virtob <- ydpart+sdapart*sign(alphafinepart[h])*
             qt((1-abs(alphafinepart[h])/2),as.numeric(dfspart))
  }else{virtob <- ydpart}
  desvirtob[,h] <- desifun(virtob, desiparams[[k]])
 }
 return(desvirtob)
}
```

```
#####function that generates the candidates for the updating points (the core of mtEGO)
#vp1w -- data.frame of the observed design points with their desirabilities
#model1 -- data.frame of the predictions and predictions errors for the currently fitted model
#           the data frame has to be ordered as follows:
#           x1, ... xn, y1, .., yb, sy1, ..., syb, df1, ..., dfb
#desiparams -- list of specification parameters for the desirability functions
#              (specified as in desifun() )
#weightedDI -- defualt is an unweighted geometric mean as desirability index
#              (optional can weights be given as a vector)
#alphafine -- the vector of alpha-levels used for the optimization given in the form
#              as shown by this example c(-0.0001, -0.001, -0.01, 1 , 0.01,0.001, 0.0001)
#idim -- dimension of the parameter space
#odim -- number of objectives
#allow.repeatpoint -- the algorithm may suggest update points at already observed points
#                     if the predciton error is not 0 there, this can be surpressed
mtEGO <- function(vp1w, model1, desiparams, weightedDI=NULL, alphafine,
                                      idim, odim, allow.repeatpoint=FALSE)
{library(gtools)
 if(is.null(weightedDI)) weightedDI <- rep(1/odim, odim)
 # yd, sda, and x include a dummy variable to enable odim=1
 yd <- data.frame(model1[,(idim+1):(idim+odim)], rep(NA,dim(model1)[1]))
 sda <- data.frame(model1[,(idim+odim+1):(idim+2*odim)], rep(NA,dim(model1)[1]))
 x <- data.frame(model1[, 1:idim], rep(NA,dim(model1)[1]))
 dfs <- c(as.vector(model1[1,(idim+2*odim+1):(idim+3*odim)]), NA)

 wimaxcorr <- wimaxcorrfun(vp1w, model1, desiparams)
   if(allow.repeatpoint==FALSE) sda <- allowrepeatfun(vp1w, sda, model1)

 #determine virtual observations and their desirabilities for all confidence boundaries
 desvirtobs <- list()
 for(k in 1:odim){desvirtobs[[k]]<-desvirtobsfun(alphafine,yd[,k],sda[,k],dfs[k],k=k)}

 #determine improvements and candidates that maximize the improvements
 virtkombis <- permutations(length(alphafine), odim, repeats.allowed=TRUE)
 imax <- imean <- rep(NA, dim(virtkombis)[1])
 optipoints <- list()
 sdadummy <- data.frame(rep(NA,dim(yd)[1]), sda)
 for(g in 1:dim(virtkombis)[1])
 { desvirtobskombi <- matrix(rep(NA,odim*dim(yd)[1]),ncol=odim)
   for(z in 1:odim) desvirtobskombi[,z] <- desvirtobs[[z]][,virtkombis[g,z]]
   simwges <- apply(desvirtobskombi, 1, function(y) prod(y^weightedDI))
   imps <- simwges - wimaxcorr
   imps[which(simwges <= wimaxcorr)] = 0
   imps[which(sdadummy[,2] == 0)] = 0
   ims <- imps
   #measures needed for the stopping criterion
   imax[g] <- max(ims)
   imean[g] <- mean(ims)
   if(all(alphafine[virtkombis[g,]] == rep(1, odim))) wcurr <- simwges
   #determine the canidates
   if(length(which(ims != 0)) != 0)
   { optipoints[[g]] <- which(ims == max(ims))
   }else{ optipoints[[g]] <- NA
   }
 }
 remove(sdadummy)
 opti <- NULL
 for(n in 1: dim(virtkombis)[1])
 {if(!is.na(optipoints[[n]])[1])opti<-c(opti,t(optipoints[[n]]))}

 #if all improvements are 0, the global optimum is assumed to be found and printed out
 if( is.null(opti))
 {  gloptimum <- model1[which(wcurr==max(wcurr)),1:idim]
    cat("All Improvements 0, global Optimum found at", as.character(gloptimum))
    optihelper <- "all expected improvements 0 - no optimum/updating step"
 }else{
    optiind <- opti
    opti <- x[optiind,-(idim+1)]
    if(idim==1){
      opti <- data.frame(opti, rep(NA, length(opti)))
      optihelper <- unique(opti[which(!is.na(opti)[,1]),])
    }else{
      opti <- data.frame(opti, rep(NA, dim(opti)[1]))
      optihelper <- unique(opti[which(!is.na(opti[,1])),])
   }
 }
 #stopping criteria
 errmax <- apply(model1[,(idim+odim+1):(idim+2*odim)], 2, max, na.rm=TRUE)
 optvor <- data.frame(curopt=model1[wcurr == max(wcurr),1:idim], wmax=max(wcurr))
 maximax <- max(imax)
 meanimax <- mean(imax)
 maxerrmax <- max(errmax)
 stops <- data.frame(relmaximax=maximax*maxerrmax, relmeanimax=meanimax*maxerrmax)
 stopps <- list(optvor, stops)
 return(list(optihelper=optihelper, optiind=unique(optiind), stoppings=stopps))
}
```

## R code for Step 3.C of mtEGO

```
###subroutine to determine the currently predicted optimum, i.e.  the parameter setting with the
#currently best predictions (model, desiparams, weightedDI, odim as described in mtEGO())
currentOpt <- function(model, desiparams, weightedDI, odim)
{ if(is.null(weightedDI)) weightedDI <- rep(1/odim, odim)
  wfinal <-matrix(rep(NA,odim*dim(model)[1]), ncol=odim)
  for(f in 1:odim){ wfinal[,f] <- desifun(model[,idim+f], desiparams[[f]]) }
  wifinalges <- apply(wfinal, 1, function(y) prod(y^weightedDI))
  gloptimum <- model[which(wifinalges==max(wifinalges)),1:idim]
  return(list(wiges=wifinalges,gloptimum=gloptimum))
}

###function to eliminate the current optimum (step 3.C.1)
```

```
#curopt -- currently predicted optimum (value gloptimum from function currentOpt() )
#candidates -- and object from function mtEGO() containing the values optihelper and optiind
elimcuropt <- function(curopt, candidates)
{if(idim==1){ ind <- which(candidates$optihelper[,1] == curopt)
 }else{
  for(m in 1:dim(curopt)[1])
  {indges <- NULL
   indpart <- list()
   equals <- rep(NA, dim(candidates$optihelper)[1])
   for(k in 1:idim)
   {for(z in 1:dim(candidates$optihelper)[1])
    {equals[z] <- all.equal(candidates$optihelper[z,k],as.numeric(curopt[m,k]))
    }
    indpart[[k]] <- which(equals==TRUE)
    indges <- c(indges, indpart[[k]])
   }
  ind <- as.numeric(names(which(table(indges)==idim)))
  }
 }
 if(length(ind) != 0)
 { elimoptihelper <- candidates$optihelper[-ind,]
   elimoptiind <- candidates$optiind[-ind]
 }else{elimoptihelper <- candidates$optihelper; elimoptiind <- candidates$optiind}
 return(list(elimoptihelper=elimoptihelper, elimoptiind=elimoptiind))
}

### function to (automatically) choose the appropriate number of groups (step 3.C.2) and to produce
#the dendrogram and scree plot for the clustering
#elimoptihelper -- data.frame of candidates suggested by mtEGO after the elimination of the
#                  current optimum, i.e. set C* (value elimoptihelper fom function elimcuropt())
#grmax -- maximum of group fusions that is displayed in the Screegraph
#         (grmax must be larger than maximum)
#minimum -- minimum number of groups to be fromed
#maximum -- maximum number of groups to be formed
#model, desiparams, weightedDI, odim, idim as described in mtEGO()
grouping <- function(elimoptihelper, grmax=20, model, desiparams, weightedDI=NULL, odim,
                     idim, minimum=1, maximum=10)
{if(dim(elimoptihelper)[1] > 5)
 {  par(mfrow=c(2,1))
    klass <- hclust(dist(elimoptihelper[,-dim(elimoptihelper)[2]]))
    plot(klass)
    vals <- data.frame(gr=seq(dim(elimoptihelper)[1]-1, 1,-1), he=klass$height)
    valshe <- sort(vals$he, decreasing=TRUE)[1:grmax]
    steig <- (c(NA,valshe) - c(valshe, NA)) / -1
    diffm <- (c(steig[-1],NA) - c(steig))
    halb <- which(valshe < (max(valshe, na.rm=T)*0.5))
    if(length(halb)==0) halb <- c(1:(length(valshe)))
    if(any(halb < maximum)){halb9 <- halb[which(halb < maximum)]}else{halb9 <- halb}
    if(any(halb9 < minimum)) halb9 <- halb9[-which(halb9 < minimum)]
    num <- which(diffm==max(diffm[halb9], na.rm=T))
    plot(valshe, type="l", ylim=c(-2,max(valshe, na.rm=T)), xlim=c(0,grmax))
    lines(seq(0.5,(grmax+0.5),1), steig, type="l", col="green")
    points(diffm, col="red")
    for(i in 1:grmax) abline(v=i, col="lightgrey", lty=2)
    text(15,-2, paste("automatic group-choice: ",num))
  }else{num <- dim(elimoptihelper)[1] }
  return(list(num=num, values=vals))
}

###function to determine groups and their representatives (step 3.C.3)
#elimcuropt -- object of function elimcuropt(), containing elimoptihelper and elimoptiind
#clusterm -- number of groups to be formed (usually determined with function grouping())
#vp1w -- current design before updating including the measurements and desirabilities
#        (with the columns ordered as follows x1, ..., xn, y1, .., yb, dy1, .., dyb, DI)
#model, desiparams, odim, idim, weightedDI as described in mtEGO()
updates <- function(elimcuropt, clusterm, vp1w, odim, idim, model, desiparams, weightedDI=NULL)
{elimoptihelper <- elimcuropt$elimoptihelper
 elimoptiind <- elimcuropt$elimoptiind
 curmod <- currentOpt(model, desiparams, weightedDI, odim)
 #subset in clusterm groups (result from step 3.C.2)
 if(dim(elimoptihelper)[1] == 1)
 { updatepoint <- elimoptihelper[,-dim(elimoptihelper)[2]]
   kombi <- "Only one updatepoint suggested, hence no clustering"
 }else{
   klass <- hclust(dist(elimoptihelper[,-dim(elimoptihelper)[2]]))
   kombi <- data.frame(optis=elimoptihelper[,-dim(elimoptihelper)[2]], kl=cutree(klass,
            k=min(c(dim(elimoptihelper)[1],clusterm)) ), curwges=curmod$wiges[elimoptiind])
 }
 ### determine representatives
 curwges <- curmod$wiges
 knr <- max(kombi$kl)
 updatepoint <- matrix(rep(NA, knr*(idim+1)), ncol=(idim+1))
 for(j in 1:knr)
 {klj <- subset(kombi, kl==j)[,-(idim+2)]
  if(dim(klj)[1] == 1){
   for(m in 1:(idim+1))
   { updatepoint[j,m] <- klj[,m]
   }
  }else{
   h1 <- as.matrix(dist(klj))
   means <- as.vector(apply(h1, 1, mean))
   reprind <- which(means == min(means))
   h2 <- which(kombi$curwges[reprind] == max(kombi$curwges[reprind]))
   if(length(h2) > 1)
   {  reprind2 <- reprind[sample(length(reprind),1)]
   }else{ reprind2 <- reprind[h2] }
   updatepoint[j,] <- as.numeric(klj[reprind2,])
  }
 }
 z  <- NULL
 for(hj in 1:idim)
 { z <- c(z,which(vp1w[,hj] == curmod$gloptimum[,hj]))
 }
 repres <- as.matrix(updatepoint[,1:idim])
 if(!any((table(z) == idim)))
 updatingpoints<-rbind(as.matrix(updatepoint[,1:idim]),as.matrix(curmod$gloptimum[,1:idim]))[,-(idim+1)]
 if(idim==2)
 {plot(model[,1],model[,2], col="white")
  points(vp1w)
  points(updatingpoints, col="red")}
 return(list(kombi=kombi, updatingpoints=updatingpoints, curopt=as.matrix(curmod$gloptimum)))
}
```

## Exemplary application of the functions

```
x1 <- x2 <- seq(-2,2,0.05)
paramspace <- data.frame(x1=rep(x1, length(x2)), x2=rep(x2,rep(length(x2),length(x2))))
idim=2; odim=2; weightedDI=NULL; thopt=data.frame(x1=0,x2=0)
desi1 <-desi2 <- c(2,0.3,0.5,0.7,2)
desiparams <- list(desi1, desi2)
measure <- function(x)
{   if(is.null(dim(x))){ x1a <- x[1]; x2a <- x[2] }else{ x1a <- x[,1]; x2a <- x[,2]}
    y1 <- 1 - exp(-1*((x1a - 1/sqrt(2))^2 + (x2a - 1/sqrt(2))^2))
    y2 <- 1 - exp(-1*((x1a + 1/sqrt(2))^2 + (x2a + 1/sqrt(2))^2))
    w1 <- round(desifun(y1, desiparams[[1]]), digits=4)
    w2 <- round(desifun(y2, desiparams[[2]]), digits=4)
    return(point=data.frame(x1=x1a,x2=x2a, y1=y1, y2=y2, w1=w1,w2=w2, wges=w1^0.5*w2^0.5))
}
vlmop2 <- measure(paramspace)
####step 1
vp1 <- measure(data.frame(x1=c(1,0.8,1,-1,-1.4), x2=c(-1.2,1.8,0.4,-1.2,1)))
####step 2
model <- sasmodelfit(vp1, sasprog=paste("C:/mtEGO-Vlmop2Two-auto.sas",sep=""),
        vpwfile="C:/Xphd-project/1a-thesis/simulationstudy/auto-vp.txt",
        modelfile="C:/Xphd-project/1a-thesis/simulationstudy/auto-model.csv")
####step 3
#3.A.1
alphafine <- c(-0.01, -0.1, -0.5,-0.9, 1,0.9, 0.5, 0.1, 0.01)
#3.A.2 - 3.B.2
candidates <- mtEGO(vp1w=vp1,model1=model,desiparams=desiparams,weightedDI=weightedDI,
                  alphafine=alphafine, idim=idim, odim=odim, allow.repeatpoint=FALSE)
#3.C.1
curopt <- currentOpt(model, desiparams, weightedDI, odim)
casterix <- elimcuropt(curopt$gloptimum, candidates)
#3.C.2
grnum <- grouping(casterix$elimoptihelper, grmax=25, model, desiparams,weightedDI=NULL, odim, idim,
                  minimum=1, maximum=10)
#3.C.3
updatingpoints <- updates(casterix, clusterm=grnum$num, vp1w=vp1, odim, idim, model, desiparams,
                          weightedDI=NULL)
```

## Improved version of mtEGO, implemented only for two-sided optimization problems

```
#vp1w -- data.frame of the observed design points with their desirabilities
#model1 -- data.frame of the predictions and predictions errors for the currently fitted model
#          the data frame has to be ordered as follows:
#          x1, ... xn, y1, .., yb, sy1, ..., syb, df1, ..., dfb
#desiparams -- list of specification parameters for the desirability functions
#             (specified as in desifun() )
#weightedDI -- defualt is an unweighted geometric mean as desirability index
#             (optional can weights be given as a vector)
#alphadec -- alpha-level to decide whether smaller, similar or larger alphas are needed
#alphasmall -- vector of alpha-levels used if the decision is to use smaller alpha levels,
#             it must be given in the form c(0.01, 0.1)
#alphalarge -- vector of alpha-levels used if the decision is to use larger alpha levels,
#             it must be given in the form alphalarge=c(0.5,0.8)
#alphasimilar --vector of alpha-levels used if the decision is to use smaller alpha levels,
#             it must be given in the form alphasimilar=c(0.1,0.5)
#idim -- dimension of the parameter space
#odim -- number of objectives
#allow.repeatpoint -- the algorithm may suggest update points at already observed points
#                     if the predciton error is not 0 there, this can be surpressed
mtEGOimp <- function(vp1w, model1, desiparams, weightedDI=NULL, alphadec, alphasmall, alphalarge,
                     alphasimilar, idim, odim, allow.repeatpoint=FALSE)
{library(gtools)
 if(is.null(weightedDI)) weightedDI <- rep(1/odim, odim)
 # yd, sda, and x include a dummy variable to enable odim=1
 yd <- data.frame(model1[,(idim+1):(idim+odim)], rep(NA,dim(model1)[1]))
 sda <- data.frame(model1[,(idim+odim+1):(idim+2*odim)], rep(NA,dim(model1)[1]))
 x <- data.frame(model1[, 1:idim], rep(NA,dim(model1)[1]))
 dfs <- c(as.vector(model1[1,(idim+2*odim+1):(idim+3*odim)]), NA)
 wimaxcorr <- wimaxcorrfun(vp1w, model1, desiparams)
 if(allow.repeatpoint==FALSE) sda <- allowrepeatfun(vp1w, sda, model1)
 # determine which alphas are to be taken, smaller, larger or similar?
 # (decisions only valid for two-sided problems)
 orgob <- decob1 <- decob2 <- decis <- matrix(rep(NA,length(yd[,1])*odim),ncol=odim)
 maxoccur <- targetp <- targetm <-  NULL
 desvirtobs <- list()
 alphafine <- list()
 le <- length(yd[,1])
 target <- NULL
 for(k in 1:odim)
 {target[k] <- desiparams[[k]][3]
  decob1[,k] <- yd[,k] + sda[,k] * qt((1-alphadec/2), as.numeric(dfs[k]))
  decob2[,k] <- yd[,k] - sda[,k] * qt((1-alphadec/2), as.numeric(dfs[k]))
  orgob[,k] <- yd[,k]
  targetp[k] <- target[k] + 0.25*(as.numeric(dist(range(orgob[,k]))))
  targetm[k] <- target[k] - 0.25*(as.numeric(dist(range(orgob[,k]))))
  lower <- upper <- 0
  for(m in 1:le)
  {if(orgob[m,k] > target[k])
  {lower <- lower + 1
   if(decob2[m,k] >= targetp[k]) decis[m,k] <- "smaller"
   if((decob2[m,k] > targetm[k]) & (decob2[m,k] < targetp[k])) decis[m,k] <-"similar"
   if(decob2[m,k] <= targetm[k]) decis[m,k] <- "larger"
  }
  if(orgob[m,k] < target[k])
  {upper <- upper + 1
   if(decob1[m,k] > targetp[k]) decis[m,k] <- "larger"
   if((decob1[m,k] > targetm[k]) & (decob1[m,k] < targetp[k])) decis[m,k] <-"similar"
   if(decob1[m,k] < targetm[k]) decis[m,k] <- "smaller"
  }
  if(orgob[m,k] == target[k])
  {   decis[m,k] <- "similar"
  }
 }

occur <- c(length(which(decis[,k] == "larger")),length(which(decis[,k] == "similar")),
           length(which(decis[,k] == "smaller")))
maxoccur[k] <- c("larger","similar","smaller")[which(occur == max(occur))]
if(lower == le) #if all predictions larger than target, consider only lower boundaries
{if(maxoccur[k] == "similar") alphafine[[k]] <- c(-1*alphasimilar, -1*alphadec, 1)
 if(maxoccur[k] == "larger")  alphafine[[k]] <- c(-1*alphadec, -1*alphalarge, 1)
 if(maxoccur[k] == "smaller") alphafine[[k]] <- c(-1*alphasmall, -1*alphadec, 1)
}
if(upper == le) #if all predictions smaller than target, consider only upper boundaries
{if(maxoccur[k] == "similar") alphafine[[k]] <- c(1, alphadec, alphasimilar)
 if(maxoccur[k] == "larger")  alphafine[[k]] <- c(1, alphalarge, alphadec)
 if(maxoccur[k] == "smaller") alphafine[[k]] <- c(1, alphadec, alphasmall)
}
if(lower != le && upper != le)
{if(maxoccur[k] == "similar")
 alphafine[[k]] <- c(-1*alphasimilar, -1*alphadec, 1, alphadec, alphasimilar)
 if(maxoccur[k] == "larger")
 alphafine[[k]] <- c(-1*alphadec, -1*alphalarge, 1, alphalarge, alphadec)
 if(maxoccur[k] == "smaller")
 alphafine[[k]] <- c(-1*alphasmall, -1*alphadec, 1, alphadec, alphasmall)
}
#determine virtual observations and their desirabilities for all confidence boundaries
desvirtobs[[k]] <- desvirtobsfun(alphafine[[k]], yd[,k], sda[,k], dfs[k], k=k)
}
#determine all cross-combinations of alpha levels acc. to step 3.A.3
lens <- NULL
for(i in 1:odim)
{  lens[i] <- length(alphafine[[i]]) }
eins <- function(a){ rep(1,a) }
virtkombis <- matrix(rep(NA, prod(lens)*odim), ncol=odim)
virtkombis[,1] <- rep(c(1:lens[1]), prod(lens[-1]))
if(odim > 2)
{for(p in 2:(odim-1))
 {virtkombis[,p] <- eins(prod(lens[(p+1):odim]))%x%(c(1:lens[p])
                   %x%eins(prod(lens[1:(p-1)])))
 }
}
virtkombis[,odim] <- c(1:lens[odim]) %x% eins(prod(lens[-odim]))

#determine improvements and candidates that maximize the improvements
imax <- imean <- rep(NA, dim(virtkombis)[1])
optipoints <- list()
sdadummy <- data.frame(rep(NA,dim(yd)[1]), sda)
for(g in 1:dim(virtkombis)[1])
{ desvirtobskombi <- matrix(rep(NA,odim*dim(yd)[1]),ncol=odim)
  for(z in 1:odim) desvirtobskombi[,z] <- desvirtobs[[z]][,virtkombis[g,z]]
  simwges <- apply(desvirtobskombi, 1, function(y) prod(y^weightedDI))
  imps <- simwges - wimaxcorr
  imps[which(simwges <= wimaxcorr)] = 0
  imps[which(sdadummy[,2] == 0)] = 0
  ims <- imps
  #measures needed for the stopping criterion
  imax[g] <- max(ims)
  imean[g] <- mean(ims)
  #determine the candidates for the updatings
  if(length(which(ims != 0)) != 0)
 { optipoints[[g]] <- which(ims == max(ims))
  }else{ optipoints[[g]] <- NA
  }
}
remove(sdadummy)
opti <- NULL
for(n in 1: dim(virtkombis)[1])
{if(!is.na(optipoints[[n]])[1]) opti <- c(opti, t(optipoints[[n]])) }

#if all expected improvements are 0, the global optimum is assumed to be found
wfinal <-matrix(rep(NA,odim*dim(model1)[1]), ncol=odim)
for(f in 1:odim)
{  wfinal[,f] <- desifun(model1[,idim+f], desiparams[[f]])
}
wcurr <- apply(wfinal, 1, function(y) prod(y^weightedDI))
if( is.null(opti))
{  gloptimum <- model1[which(wcurr==max(wcurr)),1:idim]
   cat("All Improvements 0, global Optimum found at", as.character(gloptimum))
   optihelper <- "all expected improvements 0 - no optimum/updating step"
}else{
optiind <- opti
opti <- x[optiind,-(idim+1)]
 opti <- matrix(opti, ncol=idim, byrow=TRUE)
if(idim==1){
  opti <- data.frame(opti, rep(NA, length(opti)))
  optihelper <- unique(opti[which(!is.na(opti)[,1]),])
}else{
  opti <- data.frame(opti, rep(NA, dim(opti)[1]))
  optihelper <- unique(opti[which(!is.na(opti[,1])),])
}
```

```
 }
 #the stopping criteria
 errmax <- apply(model1[,(idim+odim+1):(idim+2*odim)], 2, max, na.rm=TRUE)
 optvor <- data.frame(curopt=model1[wcurr == max(wcurr),1:idim], wmax=max(wcurr))
 maximax <- max(imax)
 meanimax <- mean(imax)
 maxerrmax <- max(errmax)
 stops <- data.frame(relmaximax=maximax*maxerrmax, relmeanimax=meanimax*maxerrmax)
 stopps <- list(optvor, stops)
 return(list(optihelper=optihelper, optiind=unique(optiind), stoppings=stopps,decis=decis, lens=lens))
}
```