Single- and Multi-objective Evolutionary Design Optimization Assisted by Gaussian Random Field Metamodels

Dissertation
zur Erlangung des Grades eines
D o k t o r s d e r N a t u r w i s s e n s c h a f t e n
der Universität Dortmund
am Fachbereich Informatik
von

Michael T. M. Emmerich

Dortmund

2005

Tag der mündlichen Prüfung:	21. October 2005
Dekan / Dekanin:	Prof. Dr. Bernhard Steffen
Gutachter:	Prof. Dr. Hans-Paul Schwefel Prof. Dr. Peter Buchholz

Contents

Preface

Mathematics knows no races or geographic boundaries; for mathematics, the cultural world is one country.

David Hilbert

This thesis has been written during the time I worked on research projects at the Computer Science Department of the University of Dortmund and the Center for Applied Systems Analysis at the Informatik Centrum Dortmund. All these years I appreciated the friendly atmosphere and the encouragement of my colleagues. First of all, I would like to thank my supervisor Prof. H.-P. Schwefel, who gave me the opportunity to work in the challenging field of systems analysis and optimization, and to whom I owe a great deal of my knowledge in this field. Also, I am grateful for the time he took for discussing various aspects of the thesis and for his advise on scientific writing. Also, I would like to acknowledge the support of Prof. Dr. P. Buchholz for his constructive criticism and time he took for discussing my ideas. In particular, I want to thank K. Giannakoglou, whose pioneering work in the topic of design optimization and artificial intelligence was a starting point for my work and who supported me by being always a patient and critical discussion partner. B. Naujoks and N. Beume I want to thank for the good collaboration on multi-objective optimization and for helping me to revise the manuscript of this thesis. Then, I want to acknowledge Th. Bäck, J. Jakumeit, F. Hediger, L. Fourment, T. Tho Do, M. Özdemir, I. Schwab, M. Sch(ü)tz, F. Henrich, B. Groß, M. Grötzner, G. Laschet, C. Varcol, L. Willmes and A. Giotis and many more for their good guidance and/or cooperation during the projects at the ICD/CASA, where I started to elaborate on this thesis. Also, I want to acknowledge J. Zhou, L. Schönemann, M. Preuss, G. Jankord, R. Hosenberg, and all other co-workers at the LSXI for supporting me with the project work during my employment at the University of Dortmund. Ch. Richter I would like to thank for his advise on writing and finishing the thesis. Also, I would like to thank U. Hermes for keeping my computer running and for his advise on all kinds of technical problems. My friends and family I would like to thank for their adherence in years where time was a very limited resource and for many other things impossible to mention in this small preface.

1 Introduction

Design optimization deals with the improvement of systems in industry, economy, and society, due to one or several objectives specified by a systems designer. In the field of design optimization, computers have become a valuable tool for exploring large and complex design spaces that can relief the systems designer from tedious computation tasks so that he/she can concentrate on tasks such as modeling and decision making.

This thesis puts forward the development of computing techniques for design optimization. It focuses on the development of robust algorithms for optimization with time-consuming evaluations. The main working principle of these techniques is to combine spatial interpolation techniques with evolutionary algorithms, which are robust population-based search techniques. This thesis will deal also with multi-objective problems, where the focus is on finding compromise solutions and on visualizing trade-offs among various objectives.

1.1 Computer experiments

Design optimization is often carried out as a process of discovery due to repeated experimentation. It is one of the consequences of the so-called 'digital revolution' that *computer experiments* (CE) have widely replaced physical experiments. They often provide a safe and/or cheap alternative to physical experiments.

A computer experiment can be specified as a black box procedure that performs a mapping from a space of input variables $\mathbb{S}$ (*design* or *search space*) to a space of output values $\mathbb{Y}$ (*response space*). Whenever the output values correspond to the objectives of an optimization problem, the response space will also be termed the *decision space*.

In this work deterministic computer experiments for continuous domains with fixed cardinality will be addressed. Accordingly, the input- output function is considered to be a mapping $\mathbb{R}^d \rightarrow \mathbb{R}^{n_y}$ with d input variables and n_y output variables. Of course, it is also possible to have other input spaces, e. g. the space of integer variables or even variable dimensional input spaces. At the end of this work there will be a brief discussion on how the techniques developed here could be generalized for such input spaces.

Optimization with computer experiments often works interactively. The user specifies, executes and analyzes experiments. Based on the analysis the user sets up new experiments and schedules them. He/she repeats the procedure until the result is satisfactory. Such a manual experimentation can be tedious task. It is often far more effective, that the human designer specifies the objective and constraints and then starts an optimization

strategy that automatically searches for feasible solutions that meet the specified objectives. Note, that in practice the specification of a clear problem statement and the right choice of an optimization strategy can be a difficult task on its own that often involves the close co-operation between experts in the application domain and experts in the field of optimization algorithms.

The result of an automatic optimization can be a single solution or, in case of conflicting objectives, a set of compromise solutions. From the obtained set the systems designer can then choose a solution that fits his/her expectations, or else restart the optimization process with modified objectives or a different choice and parametrization for the optimization algorithm.

Many design optimization problems can be transformed to the following standard form:

$$f_1(\mathbf{x}) \to \min, \ldots, f_{n_f}(\mathbf{x}) \to \min \tag{1.1.1}$$

$$g_1(\mathbf{x}) \geq 0, \ldots, g_{n_g}(\mathbf{x}) \geq 0 \tag{1.1.2}$$

$$\mathbf{x} \in \mathbb{S} \subseteq \mathbb{R}^d \tag{1.1.3}$$

Here $f_i : \mathbb{R}^d \to \mathbb{R}^{n_f}, i = 1, \ldots, n_f$ denotes objective functions that are to be minimized, while $g_i : \mathbb{R}^d \to \mathbb{R}^{n_g}, i = 1, \ldots, n_g$ denotes constraint functions that have to be kept higher than zero. It is a common practice to restrict the search space by means of variable bounds $\mathbb{S} := [\mathbf{x}_{\min}, \mathbf{x}_{\max}]$, where $\mathbf{x}_{\min} \in \mathbb{R}^d$ and $\mathbf{x}_{\max} \in \mathbb{R}^d$ are user defined lower and upper bounds for the design variable $\mathbf{x}$. Note, that many optimization algorithms do not demand for such bounds and can also deal with unconstrained search spaces like for example $\mathbb{R}^d$.

The objective and constraint functions are typically computed from the output values $\mathbf{y}$ of the black box analysis tool by means of simple transformations. For the sake of transparency, we make the default assumption that the output values $\mathbf{y}$ directly correspond to the values of the objective and constraint functions, i. e.:

$$f_1(\mathbf{x}) = y_1(\mathbf{x}), \ldots, f_{n_f}(\mathbf{x}) = y_{n_f}(\mathbf{x}), g_1(\mathbf{x}) = y_{n_f+1}(\mathbf{x}), \ldots, g_{n_g}(\mathbf{x}) = y_{n_f+n_g}(\mathbf{x}) \tag{1.1.4}$$

However, it has to be noted that it sometimes can be important to take a closer look at the mapping between the actual output values of a computer experiment and the constraint and objective function values, e. g. in order to exploit simple dependencies among different objectives and constraints.

The context in which optimization methods are used can be very important, since the desirable characteristics of a method strongly depend on it. This work was motivated by problems that actually were encountered in industrial design optimization.

Looking at many examples from industrial optimization and also studying the applications that motivated this work [EGN05, EGÖ+02, EJ03, ESB02, ESGG00], an important class of industrial design optimization problems can be obtained, where the number of variables ranges from 1 to 20 and a small number of objectives and constraints $\ll 10$ is given. Furthermore, computer experiments are often based on simulation tools, utilizing solvers for nonlinear (differential) equation system. This means that the time of one evaluation ranges from several minutes up to hours and thus only a few hundred objective function evaluations (100 – 1000) can be spend for the purpose of optimization. However, for medium-sized research enterprizes it is often possible to run a small number (≈ 10) of experiments in parallel. With regard to the black box function topology,

another assumption that will be made is that there is some continuity in the output of the computer experiment, meaning that similar input values are likely to result in similar output function values.

During the course of an optimization study, optimization methods are often further refined in order to better tackle the problem at hand. Therefore, the systems designer wants to understand the basic search principles of the search method. Many optimization techniques are very complex since they are designed to have some kind of optimal behavior on certain types of function classes, like for example quadratic functions. However, in a black box scenario it is often questionable whether the true structure of a black box function corresponds to the assumed class of functions and thus the complexity of the optimization method might be unjustified. In such cases *transparent* and *flexible* methods should be preferred, which are open to the integration of further domain specific knowledge. Furthermore, methods should be *robust*, meaning that they work also in cases when the behavior of the black box functions deviates from the model assumptions.

1.2 Objectives of this work

Metamodel-assisted evolutionary algorithms that are studied in this work are well-suited techniques for tackling the aforementioned kind of problems. They combine robust search heuristics, namely evolutionary algorithms (EA), with versatile tools for spatial interpolation, namely gaussian random field metamodels (GRFM). Encouraged by some earlier results [EGÖ+02], the goal of this thesis is to put forward the study of these algorithms and to extend make them applicable for solving constrained and multi-objective problems.

The following list summarizes the main research problems tackled within this thesis:

- How does different methods for modeling experimental data compare? In particular we are interested in a comparison between radial basis function networks, regression models and gaussian random field metamodels, all of them being frequently proposed for assisting optimization algorithms.
- How can metamodels be integrated into evolutionary algorithms? A focus will be on techniques, so called *filters*, that identify promising solutions from a large set of generated variants by means of the metamodel. Unlike in most of the previous work in the field, the confidence information of predictions should be considered, e. g. in order to facilitate the search in less explored regions of the search space.
- Which theoretical properties can be deduced from the algorithmic design of the metamodel-assisted evolutionary algorithms? Before applying them in practise, we want to discuss theoretical properties of metamodel-assisted evolutionary algorithms. An important question will be, how different filters relate to each other, e. g. if certain filters are equivalent under certain parameterizations.
- How to analyze the behavior metamodel-assisted algorithms? First of all, we want to mark the limits of a analytical study of the metamodel-assisted evolutionary algorithms and thus motivate the necessity of empirical research. Furthermore, we look for well-suited experimental analysis methods for these algorithms. Besides

performance measures, indicators that allow for a deeper understanding of their working principles are to be found.

- How do metamodel-assisted algorithms perform on different classes of problems? The aforementioned analysis techniques are to be used to identify the assets and shortcomings of different MAEA variants. Furthermore, we aim at a deeper understanding of the practical working principles of the MAEA.

- How can metamodel-assisted evolutionary algorithms be generalized for constrained optimization? Here we intend the straightforward generalization of filters for constrained optimization, if the evaluation of constraint function is part of the time consuming computer experiment.

- How can metamodel-assisted evolutionary algorithms be generalized to multi-objective optimization? The aim is to generalize the filters for Pareto optimization, where the aim is to find a set of efficient solutions. In order to achieve this aim a redesign of existing evolutionary multi-objective optimization techniques is required. Furthermore, new test problems are to be developed that are better suited than the ones proposed in literature for comparing and understanding the behavior of multi-objective optimization algorithms in the presence of time consuming evaluations.

- How do metamodel-assisted evolutionary algorithms perform on real-world problems? Artificial test problems can hardly emulate all characteristics of practical design optimization problems. Thus, finally, the promising variants of metamodel-assisted evolutionary algorithms should be tested on optimization problems in industrial design. We envisage a broad spectrum of application domains, and a comparison to state-of-the-art optimization techniques in the particular field.

This thesis is structured as follows: In chapter 2 gaussian random field models are introduced and compared to other interpolation methods. Chapter 3 provides a brief survey on the state of the art in global optimization methods for single-objective optimization. In chapter 4 metamodel-assisted evolution strategies for single criterion optimization are developed and studied. A focus will be on the comparison of different filters used in the pre-selection. In chapter 5 the proposed algorithms are generalized for the constrained case. In chapter 6 the problem of multi-objective optimization are discussed and a new class of evolutionary algorithms for multi-objective optimization that uses the hypervolume metric as a selection criterion will be developed and studied. In chapter 7 the generalization of the EMOA for problems with multiple objectives based on this new algorithm will be discussed. Finally, case studies on selected applications are presented in chapter 8. A summary of the work is given in chapter 9. Here, also directions for future research are envisioned.

2 Spatial modeling with gaussian random field models (GRFM)

Models should be used, not believed.

Henri Theil

In this chapter spatial modeling techniques are introduced with a focus on gaussian random field models (GRFM). The chapter starts with some practical definition of GRFM viewed as a black box in section 2.1. In section 2.2 the statistical assumptions of GRFM are discussed. In section 2.3 we proceed with running time complexity of alternative prediction procedures based on GRFM. In section 2.4 computational efficient algorithms for the implementation of GRFM are discussed. Then GRFM are related to regression models in section 2.5. In section 2.6, we relate alternative interpolation models to GRFM. In particular, we establish a mapping between artificial neural networks with radial basis functions (RBFN) and GRFM models. The section continuous includes a comparison of the computational effort for predicting multiple reponses both for the GRFM and the RBFN. Next, the some general results on the relationship between the number of evaluated points and the prediction error will be summarized (section 2.7), and practical methods for determining the prediction error online will be introduced (section 2.8).

2.1 Black box view of gaussian random field models

Let us now assume that we have some evaluated search points

$$\mathbf{X} := [\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(m)}] \in \mathbb{R}^{d \times m} \tag{2.1.1}$$

and corresponding scalar responses

$$\mathbf{y} = [y^{(1)}, \ldots, y^{(m)}]^T \in \mathbb{R}^m \tag{2.1.2}$$

with

$$y^{(1)} := y(\mathbf{x}^{(1)}), \ldots, y^{(m)} := y(\mathbf{x}^{(m)}) \tag{2.1.3}$$

that have already been calculated by means of (expensive) computer experiments. No assumption on the regularity of the distribution of $\mathbf{x}^{(1)}$, ..., $\mathbf{x}^{(m)}$ in $\mathbb{S}$ is made.

Our aim is to build a fast prediction tool, capable of approximating the output corresponding to a new point $\mathbf{x}' \in \mathbb{S}$, in conformity with the unknown $\mathbb{R}^d \to \mathbb{R}$ mapping, which

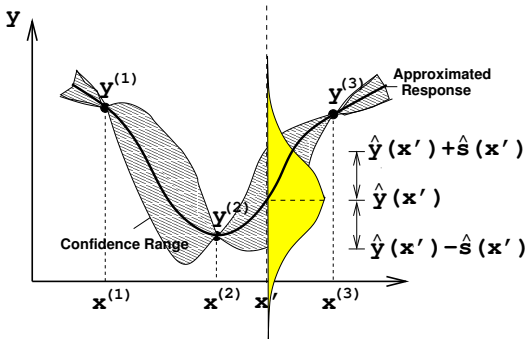


Figure 2.1.1: Understanding the output of GRFM for a problem with a single input ($d = 1$) and a single output ($n_y = 1$). With three training patterns $\mathbf{x}^{(i)}$, $i = 1, 2, 3$ the bold line corresponds to the predicted response $\hat{y} = \hat{f}(\mathbf{x}')$. The two thin lines confine the confidence interval of the response, that can be expressed by adding/substracting an estimated local standard deviation $\hat{s}(\mathbf{x}')$. The former is equal to the expected value of the conditional random variable $\mathcal{F}_{\mathbf{x}'|\mathbf{X},\mathbf{y}}$ at a new point $\mathbf{x}'$, while the latter is equal to a multitude of its standard deviation.

is assumed to be continuous. Moreover, if $\mathbf{x}' \in \mathbf{X}$, then the known output value shall be reproduced. Above we addressed the mathematical problem of exact interpolation for which a variety of methods are available, ranging from radial basis function networks [Mye92] to splines [FSATV92] and Shepard polynomials [Zup04].

Apart from a predicted approximate response $\hat{y}(\mathbf{x}')$, an additional information that we may ask for is a measure of confidence for each prediction. It is straightforward to assume that the confidence for a prediction might be better, if the density of evaluated points is very high in the neighborhood of $\mathbf{x}'$. Another relevant piece of information for its estimation is the standard deviation of the known output values and the average correlation between responses at neighboring points.

A GRFM can fulfill these requirements by interpolating data values and estimating their prediction accuracy. Putting things into more concrete terms, the GRFM predicts a gaussian random field $\mathcal{F}$. A gaussian random field is a function that assigns a one dimensional gaussian random variable $\mathcal{F}_{\mathbf{x}}$ to every position $\mathbf{x}$ in the search space $\mathbb{R}^d$. Each random variable is characterized by its expected (or: mean) value $\hat{y}(\mathbf{x})$ and its standard deviation $\hat{s}(\mathbf{x})$, and it quantifies the *probability of presence* $\Pr(\mathcal{F}_{\mathbf{x}} = y)$ for the unknown precise output. If $\Pr(\mathcal{F}_{\mathbf{x}} = y)$ takes a high value, the GRFM predicts that y is more likely the precise result. The statistical motivation of $\mathcal{F}_{\mathbf{x}}$ is further explicated in the next section. For a basic understanding of subsequently introduced algorithmic approaches it widely suffices to understand the GRFM as a black box model, which outputs an gaussian distribution that describe the probability of presence for the true output.

Example: Figure 2.1.1 illustrates an example for the prediction of a result for a new input $\mathbf{x}'$ in a one dimensional input space. Two important characteristics of $\hat{s}(\mathbf{x}')$ can be observed: First, $\hat{s}(\mathbf{x}') = 0$ if $\hat{y}(\mathbf{x}')$ is equal to the known values at the location of the training patterns $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(m)}$. Second, the standard deviation $\hat{s}(\mathbf{x}')$ is zero at the

known points and it grows with the distance of $\mathbf{x}'$ to these points. □

GRFM allow to predict the output for multivariate output spaces and other metric input spaces. In case of multivariate output spaces GRFM yields the multivariate joint distribution describing the location of the output vector for each input vector.

2.2 Model assumptions

The basic assumption in output function modeling through GRFM is that the output function is a realization (*sample path*) of a gaussian random field $\mathcal{F}$. The latter is a mapping that assigns a one-dimensional gaussian distributed random variable $\mathcal{F}_{\mathbf{x}}$ with constant mean $\beta := \mathrm{E}(\mathcal{F}_{\mathbf{x}})$ and variance $s^2 = \mathrm{Var}(\mathcal{F}_{\mathbf{x}})$ to each point $\mathbf{x} \in \mathbb{S}$ of an input space $\mathbb{S}$ that is element of $\mathbb{R}^d$ (in case of $d = 1$ the term gaussian process is also in use). The theory of gaussian random fields was extensively studied by Adler [Adl81] and it was applied in several fields of science, like oceanography, neurodynamics, environmetrics and astrophysics [Adl81].

In contrast to other modeling techniques such as linear regression, a spatial correlation between the output variables is assumed. For two arbitrary inputs $\mathbf{x}$ and $\mathbf{x}'$ such a spatial correlation can be expressed by a correlation function

$$c(\mathcal{F}_{\mathbf{x}}, \mathcal{F}_{\mathbf{x}'}) \equiv c'(\mathbf{x}, \mathbf{x}'). \tag{2.2.4}$$

Typically the correlation function is assumed to be stationary, i. e.

$$c(\mathcal{F}_{\mathbf{x}}, \mathcal{F}_{\mathbf{x}'}) \equiv c'(\mathbf{x} - \mathbf{x}'), \tag{2.2.5}$$

or even isotropic, i. e.

$$c(\mathcal{F}_{\mathbf{x}}, \mathcal{F}_{\mathbf{x}'}) = c(\mathcal{F}_{\mathbf{x}'}, \mathcal{F}_{\mathbf{x}}) \equiv c'(|\mathbf{x} - \mathbf{x}'|), \tag{2.2.6}$$

in which case the method depends only on the distance $|\mathbf{x} - \mathbf{x}'|$ between inputs.

In the literature on design and analysis of computer experiments, the isotropic gaussian correlation

$$c(\theta) = \exp(-\theta \cdot |\mathbf{x} - \mathbf{x}'|) \tag{2.2.7}$$

and the gaussian product kernel are often used. The latter reads

$$c(\theta_1, \ldots, \theta_d) = \prod_{i=1}^{d} \exp(-\theta_i \cdot |x_i - x_i'|), \tag{2.2.8}$$

and it allows for independent correlation factors for each coordinate of the search space. In order to completely specify the GRFM, the parameters of the correlation function $\theta_1, \ldots, \theta_d$ and the parameters s^2 and β of the random field have to be estimated. These parameters are usually estimated from the sample during a *calibration phase* (section 2.5).

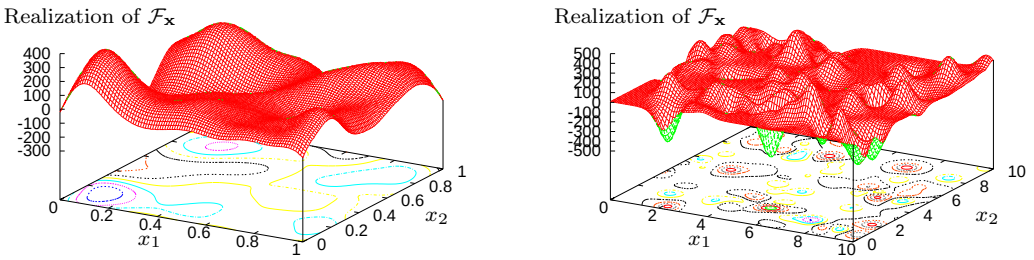


Figure 2.2.2: Sample path of a two-dimensional gaussian random field with strong correlation (left) and weak correlation (right) approximated with Krigifier [Pad00].

Example: Two examples of (predicted) sample paths for stationary GRF are given in figure 2.2.2. They were obtained with the Krigifier software developed by Padula [Pad00]. It becomes clear that for a high correlation of neighboring points (which corresponds to a low correlation parameter θ) the sample paths are very smooth and they become more bumpy for low correlation values. Hence, the estimated θ value is also an indicator of the predictability of a landscape and the amount of information that is needed to model it. □

After the calibration has been done, the parameters of the GRF are completely specified. Now, on basis of the calibrated model parameters, predictions can be computed for every input vector. The predicted distribution is the *conditional distribution* of $\mathcal{F}_{\mathbf{x}}$, given the evaluated sites $\mathbf{X}$ and $\mathbf{y}$. We will denote the corresponding family of random variables with

$$\mathcal{F}_{\mathbf{x}|\mathbf{X},\mathbf{y}}, \mathbf{x} \in \mathbb{S} \tag{2.2.9}$$

Hence, the conditional mean reads

$$\forall \mathbf{x} \in \mathbb{S} : \hat{y}(\mathbf{x}) := \mathrm{E}(\mathcal{F}_{\mathbf{x}|\mathbf{X},\mathbf{y}}), \tag{2.2.10}$$

and the conditional standard deviation reads

$$\forall \mathbf{x} \in \mathbb{S} : \hat{s}(\mathbf{x}) := \sqrt{\mathrm{Var}(\mathcal{F}_{\mathbf{x}|\mathbf{X},\mathbf{y}})}. \tag{2.2.11}$$

Note that the given information at sites $\mathbf{x} \in \mathbf{X}$ needs not necessarily to be given precisely. It also suffices to specify mean value and variance of a gaussian distribution here. This can be helpful in modeling noisy data.

2.3 Regression and Kriging models

Next, we examine the correspondence between GRFM and regression models and discuss combinations of both approaches, which are referred to as Kriging.

When working with *Kriging models* (a term that is used for GRFM and related techniques in the geostatistics community) it is common practice to use the GRFM as an offset to a regression term with regression parameters $\beta_i, i = 1, \ldots, n_r$:

$$\mathcal{F}_{\mathbf{x}} = \underbrace{\sum_{i=1}^{n_r} \beta_i \cdot r_i(\mathbf{x})}_{\text{Global trend}} + \underbrace{\mathcal{R}_{\mathbf{x}}}_{\text{Local deviation}} \tag{2.3.12}$$

The first part of this expression corresponds to a global regression model that is superposed by a spatially correlated homoscedastic GRFM $\mathcal{R}_{\mathbf{x}}$ with mean 0 and variance s^2. The latter corresponds to the noise term in regression. The regression functions $r_i : \mathbb{R}^d \to \mathbb{R}, i = 1, \ldots, n_r$ are assumed to be deterministic. They are also called trend functions.

The expression in equation 2.3.12 looks similar to the general functional regression model. In contrast to the Kriging techniques in regression no spatial correlation is assumed for the noise term. Uncorrelated noise terms can serve as a good assumption in the presence of noisy measurements, but in case of continuous deterministic input-output mappings these assumptions are unreasonable [SWMW00].

Depending on the choice of the trend functions, three types of the Kriging approach are typically distinguished:

- *Simple Kriging*: No trend is assumed, i.e. $\mathcal{F}_{\mathbf{x}} = \mathcal{R}_{\mathbf{x}}$.
- *Ordinary Kriging*: A constant trend is assumed, i. e. $\mathcal{F}_{\mathbf{x}} = \beta + \mathcal{R}_{\mathbf{x}}$
- *Universal Kriging*: A general linear trend function is assumed (expression 2.3.12)

Following a suggestion of Schonlau et al. [JSW98] ordinary Kriging will be used in this work whenever there is no justification to assume a particular trend function. However, it shall be noted here that the assumption of non-stationarity sometimes can be useful. Consider for example a model for rainfall intensity at the slope of a hill. Here, it seems reasonable to work at least with a linear trend function as a prior assumption, since the rainfall intensity probably correlates with the height of the hill.

2.4 Calibration and prediction

The parameter(s) θ as well as β and s^2 are invariant with respect to $\mathcal{F}$. Their values can be estimated through a sample by a generalized least squares method; θ is usually estimated by the maximum likelihood heuristic [SWMW00].

Without knowing the parameters of the random fields, we can express the likelihood of a sample $\mathbf{X}, \mathbf{y}$ via the probability density functions (PDF) of $\mathcal{F}_{\mathbf{x}_i}, i = 1, \ldots, m$:

$$\text{PDF}(\mathcal{F}_{\mathbf{x}_1} = y_1 \wedge \cdots \wedge \mathcal{F}_{\mathbf{x}_m} = y_m) = \tag{2.4.13}$$

$$\frac{1}{(2\pi)^{m/2} \cdot (\hat{s})^{m/2} \cdot \sqrt{\det(\mathbf{C})}} \exp\left[-\frac{(\mathbf{y} - \mathbf{1}\hat{\beta})^T \cdot \mathbf{C}^{-1} \cdot (\mathbf{y} - \mathbf{1}\hat{\beta})}{2\hat{s}} \right]$$

with

$$\mathbf{C} = \begin{bmatrix} c_\theta(\mathbf{x}_1, \mathbf{x}_1) & \cdots & c_\theta(\mathbf{x}_1, \mathbf{x}_m) \\ \vdots & \ddots & \vdots \\ c_\theta(\mathbf{x}_m, \mathbf{x}_1) & \cdots & c_\theta(\mathbf{x}_m, \mathbf{x}_m) \end{bmatrix}, \ \mathbf{1} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \tag{2.4.14}$$

using the following generalized least squares estimates [KO96] of β and s^2

$$\hat{\beta} = \frac{\mathbf{1}^T \cdot \mathbf{C}^{-1} \cdot \mathbf{y}}{\mathbf{1}^T \cdot \mathbf{C}^{-1} \cdot \mathbf{1}} \tag{2.4.15}$$

$$\hat{s} = \frac{(\mathbf{y} - \mathbf{1} \cdot \hat{\beta})^T \cdot \mathbf{C}^{-1} (\mathbf{y} - \mathbf{1} \cdot \hat{\beta})}{m} \tag{2.4.16}$$

For the maximization of the likelihood term (expression 2.4.13) it suffices to minimize the expression:

$$m \log \hat{s}(\theta) + \log \det \mathbf{C}(\theta) \tag{2.4.17}$$

Proof. The expression is derived as follows: By substituting $\hat{s}$ (expression 2.4.16) in the exponential power term of expression 2.4.13 we get:

$$\frac{1}{2\pi^{m/2} + (\hat{s})^{m/2} \cdot \det \mathbf{C}(\theta)^{1/2}} \exp(m/2) \to \max \tag{2.4.18}$$

This can be expressed as the minimization of the reciprocal term:

$$2\pi^{m/2} \cdot (\hat{s})^{m/2} \cdot \det \mathbf{C}(\theta)^{1/2} \cdot \exp(-m/2) \to \min \tag{2.4.19}$$

Finally, through logarithmization and elimination of constant values, the logarithmic likelihood expression 2.4.17 is obtained. □

The cost of the maximization of the likelihood expression depends on the number of θ-variables. Generally, due to nonlinearities, it is not always possible to solve this optimization problem in closed form. However, quasi Newton methods (cf. section 3.2) are often employed for its solutions. Partial derivatives need not to be obtained numerically. For the likelihood formula (Eq. 2.4.17) they are given in [KO96]. However, since the problem is multimodal (cf. Mac Kay [Mac98]), it cannot be guaranteed that its precise solution can be obtained with gradient based optimization. In order to achieve more robust search characteristics Torczon and Trosset [TT97] suggest to use a multidimensional pattern search algorithm (cf. section 3.3). In this work, we prefer to use a $(1+1)$ evolution strategy with 1/5th success rule for calibrating the model parameters [Sch95], which is also known as a robust and reasonably fast search heuristic.

Having estimated all parameters of the GRFM, we can calculate the mean and variance of the conditional gaussian random variable $\mathcal{F}_{|\mathbf{X},\mathbf{y}}$ at any known and unknown site $\mathbf{x} \in \mathbb{S}$:

$$\hat{y}(\mathbf{x}) = \beta + (\mathbf{y} - \mathbf{1}\beta)^T \mathbf{C}^{-1} \mathbf{c}(\mathbf{x}) \tag{2.4.20}$$

$$\mathbf{c}(\mathbf{x}) = [c_\theta(\mathbf{x}, \mathbf{x}_1), \ldots, c_\theta(\mathbf{x}, \mathbf{x}_m)]^T \tag{2.4.21}$$

The latter equation can be restated in the form of a linear predictor

$$\beta + \sum_{i=1}^{m} \lambda^{(i)} \cdot c(\mathbf{x}, \mathbf{x}_i) \tag{2.4.22}$$

with

$$[\lambda^{(1)}, \ldots, \lambda^{(m)}] = (\mathbf{y} - \mathbf{1}\beta) \cdot \mathbf{C}^{-1} \tag{2.4.23}$$

Assuming that the value of β is known, the local variance $s^2(\mathbf{x}) = \mathrm{Var}(\mathcal{F}_{\mathbf{x}'|X,y})$ for this random variable is given by

$$s^2(\mathbf{x}) = s^2 \cdot (1 - \mathbf{c}(\mathbf{x})^T \cdot \mathbf{C}^{-1} \cdot \mathbf{c}(\mathbf{x})) \tag{2.4.24}$$

Whenever the maximum likelihood estimate $\hat{\beta}$ is used instead of β, Schonlau et al. [JSW98] suggest to use instead of $s^2(\mathbf{x})$ the more exact expression $\hat{s}(\mathbf{x})$ defined as

$$\hat{s}(\mathbf{x}) = s^2 \cdot \left[1 - \mathbf{c}(\mathbf{x})^T \cdot \mathbf{C}^{-1} \cdot \mathbf{c}(\mathbf{x}) + \frac{(1 - \mathbf{1}^T \cdot \mathbf{C}^{-1} \cdot \mathbf{c}(\mathbf{x}))^2}{\mathbf{1}^T \cdot \mathbf{C}^{-1} \cdot \mathbf{1}}\right]. \tag{2.4.25}$$

The derivation of these two equations can be found in Sacks et al. [SWMW00]. Schonlau et al. [KO96] interpret the term $\mathbf{c}^T(\mathbf{x}) \cdot \mathbf{C}^{-1} \cdot \mathbf{c}(\mathbf{x})$ as the reduction in prediction error due to the fact that $\mathbf{x}$ is correlated with the sampled points. The term $(1 - \mathbf{1} \cdot \mathbf{C}^{-1} \cdot \mathbf{c}(\mathbf{x}))^2/(\mathbf{1}^T \cdot \mathbf{C}^{-1} \cdot \mathbf{1})$ is added, since the true value of β is unknown and it is estimated only from the sample.

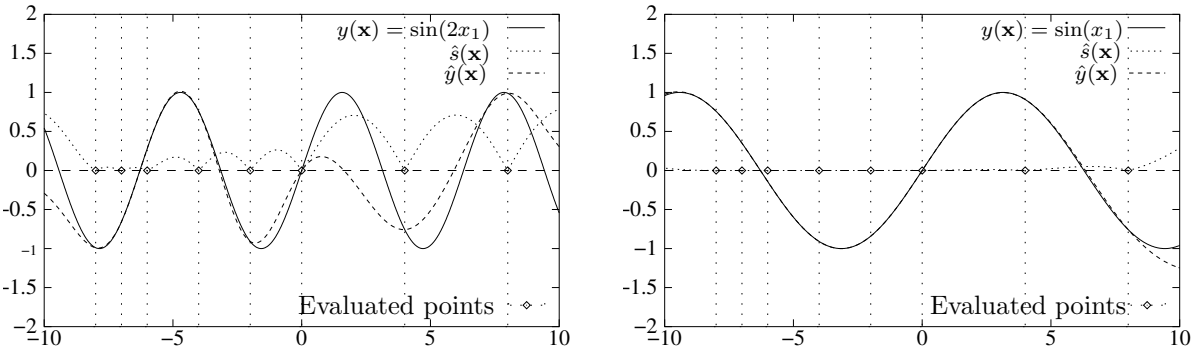


Figure 2.4.3: Interpolation with GRFM: Original function, approximation and error estimation for the one-dimensional function $\sin(2 \cdot x)$ (left) and $\sin(x)$ (left). The dashed lines mark positions that have been precisely evaluated.

Example: As an example for interpolation with GRFM two sinus functions have been approximated with $m = 8$ training points. They are depicted in Fig. 2.4.3. The approximation for the more rugged function $\sin(2x)$ (left plot) is less precise than that for the smooth function $\sin(x)$ (right plot). This is taken into account by the error predictor $\hat{s}$, which predicts higher deviations of the predicted values $\hat{y}(\mathbf{x})$ from the true values $y(\mathbf{x})$ for the first function than for the second function. It can also be observed, that all training points are precisely interpolated and that the deviation from the precise value of $y(\mathbf{x})$ grows with the distance from $\mathbf{x}$ to its neighboring training points. The error measure $\hat{s}(\mathbf{x})$ also grows with this distance and takes the value of zero at known points, which demonstrates $\hat{s}(\mathbf{x})$ is a measure for the degree of exploration for a region in the search space. □

Now, the model assumptions of GRFM have been clarified. Furthermore, we have outlined the methods for calibrating the metamodel and predicting output values by means of it. The time consumption of the different stages of computation can be significant, as will be revealed in the next section.

2.5 Computational efficiency of GRFM

Next, a detailed discussion of the time and space complexity for the Kriging method is given. It will turn out that the time complexity grows rapidly with the number of points in the database. In order to limit calculation time, training point selection methods will be suggested that reduce the number of training points, yielding in so called *local GRFM metamodels*.

The prediction of a new point with the Kriging methods can be subdivided into three phases:

1. Calibration phase: Estimation of the GRFM parameters, comprising the correlation parameter(s) θ, global variance s^2 and regression term parameter(s) $\beta_1, \ldots, \beta_{n_r}$
2. Training phase: Determination of the weights of the linear predictor (expression 2.4.22).
3. Prediction phase: Calculation of $\hat{y}(\mathbf{x})$ and – optionally – of $\hat{s}(\mathbf{x})$

Before starting with the complexity analysis for the calibration phase some notes shall be given on the algebraic time complexity of matrix operations that play an important role in the procedures for prediction with GRFM. It has been found that the algebraic complexity of matrix multiplication, matrix inversion and the determination of the determinant are essentially of the same order of magnitude [JGV03]. It has been recently obtained that a lower bound for the three operations on $m \times m$ real valued matrices can be given by $\Omega(m^2 \cdot \log m)$ (cf. [Tve03], [Ran03]). An upper bound of the algebraic time complexity for the matrix operations is $\mathcal{O}(m^3)$, which is a rough estimation that stems from the time complexity of gaussian elimination. A well known result on a tighter bound for matrix inversion of $\mathcal{O}(m^{\log 7})$ has been published by Strassen [Str69]. More recently, tighter upper bounds for matrix inversion have been obtained with $\mathcal{O}(m^{2.376})$ [CW90]. However, due to numerical instabilities it is not recommended to use Strassen's algorithm and its follow-ups [Wei04]. However, in the forthcoming analysis we will replace the precise value for the exponent by a variable w, noting that for practical implementations for $m < 50$ the value $w = 3$ is a good guess. In particular, we can exploit the symmetry of the correlation matrix and use LU factorization for determining its inverse [Pad00].

Next, the different phases of the Kriging algorithm shall be studied in more detail. The calibration phase is the most time consuming part of the three phases. For all considerations we assume an isotropic correlation function (equation 2.2.7) and alternatively an stationary correlation function with individual correlation parameters for the design variables (equation 2.2.8).

Corrolar 1. Suppose that $\mathbf{X}, \mathbf{y}$ comprises m data sets and N_{opt} evaluations of the maximum likelihood term are spent for the optimization of the GRF parameters. Then an upper bound for the operational time complexity of the calibration step of the Kriging algorithm is $\mathcal{O}(d \cdot m^2 + N_{max} \cdot m^w)$. A lower bound is given by $\Omega(d \cdot m^2 + N_{max} \cdot m^2 \cdot \log m)$.

Proof. The first term $d \cdot m^2$ originates from the time needed to calculate the entries of the distance matrix 2.4.14 for all $m \cdot (m-1)$ distinct pairs of points by means of equation

2.2.8. The second factor $N_{max} \cdot m^w$ of the expression bounds the time needed for the calculation of $\det(\mathbf{C})$. □

If we are given a fully parameterized GRFM, we can use the training data in order to estimate the weights of the linear predictor.

Corrolar 2. The operational time complexity of the training phase is upper bounded with $\mathcal{O}(d \cdot m^2 + m^w)$ and lower bounded with $\Omega(d \cdot m^2 + m^2 \cdot \log m)$.

Proof. Again, the first factor stems from the determination of the correlation matrix. The second factor in the sum originates from the bounds for matrix inversion in 2.4.23. □

Note that, if the training phase follows directly after the calibration phase, its cost reduces, since the distance matrix and the inverse matrix have already been calculated in the calibration and can be transferred from there. Hence, we get the new upper bound $\mathcal{O}(m^2)$ for the training step, which stems from the calculation of the matrix operations in Eq. 2.4.23.

Corrolar 3. The operational time complexity of the prediction of $\hat{y}$ for q new points $\mathbf{x}'_i \in \mathbb{R}$, $i = 1, \ldots, q$ is $\Theta(q \cdot d \cdot m)$.

Proof. Once the training phase has been done, the linear predictor (Eq. 2.4.22) can be applied in order to estimate all new points. □

Corrolar 4. The operational time complexity of the prediction of $\hat{s}$ for q new points $\mathbf{x}'_i \in \mathbb{R}$, $i = 1, \ldots, q$ is $\Theta(q \cdot d \cdot m^2)$.

Proof. For the matrix multiplications in 2.4.25 the algorithm needs $m^2 \cdot d$ steps. These multiplications need to be performed anew for each input vector. The inverse matrix does not have to be obtained anymore, since it can be transferred from the training phase. □

Putting these results together we get the following theorem:

Theorem 1. Let m denote the number of input points, N_{opt} denote the number of steps in the calibration, d denote the number of search space dimensions, and q the number of points that are to be predicted. Then the operational time complexity for the whole calibration and the prediction of the mean values ($\hat{y}$) and standard deviations ($\hat{s}$) for q new input vectors is bounded by

$$\mathcal{O}(d \cdot m^2 + N_{opt} \cdot m^w + q \cdot d \cdot m^2) \tag{2.5.26}$$

and

$$\mathcal{O}(d \cdot m^2 + N_{opt} \cdot m^w + q \cdot d \cdot m), \tag{2.5.27}$$

if only mean values are to be computed. The corresponding lower bounds are $\Omega(d \cdot m^2 + N_{opt} \cdot m^2 \cdot \log m + q \cdot d \cdot m^2)$ and $\Omega(d \cdot m^2 + N_{opt} \cdot m^2 \cdot \log m + q \cdot d \cdot m)$, respectively.

It becomes apparent that the number of training points has the most significant effect on the cost of the training phase, and not the dimension of the search space, as one might have assumed before. Thus, for large databases of evaluations we suggest to use only a subset of the total number of points available in the database for the metamodel training. A simple heuristic that proved to work well in empirical studies [EGN05], is to choose the k-nearest neighbors of a point $\mathbf{x}$ for training the metamodel and train a new metamodel at each point. Such a strategy would be called *local metamodeling* in contrast to *global metamodeling*, for which all evaluated points in the search space are used in order to build the model. In order to measure at least the impact of each variable it is recommended to choose m at least proportional to d.

Proposition 1. If k is chosen proportional to d, the time complexity for approximately evaluating q points is bounded by $\mathcal{O}(q \cdot (m \cdot d + m \cdot \log m + N_{opt} \cdot d^w))$, both for prediction of mean values only and prediction for standard deviations and mean values. A corresponding lower bound would be $\Omega(q \cdot (m \cdot d + m \cdot \log m + N_{opt} \cdot d^2 \log d)$.

Proof. First, we determine the distance of the new point to all other points in the database. The effort for this grows asymptotically with $\mathcal{O}(m \cdot d)$. Next, they are sorted with some efficient sorting algorithm. The time for sorting is estimated as $\mathcal{O}(m \log m)$. □

This is considerably faster, whenever $d \ll m$. In this work we will focus on search spaces with low or medium dimension.

However, local metamodels can be implemented more efficiently using for example a spatial database for storing the evaluated points. However, for the studies performed in this work, a straightforward implementation suffices to obtain predictions within a reasonable time, that is still orders of magnitudes lower than a precise evaluations.

Note, that GRFM can also be used for the *prediction of multiple output values*. Assuming independent response functions, this can be simply achieved by maintaining separate models for the different responses.

Artificial neural networks, that can have multiple outputs, might provide an alternative technique. Next, a detailed conceptual comparison of both approaches, shall reveal their essential difference.

2.6 Comparison to radial basis function networks

Frequently used techniques for nonlinear function approximation are artificial neural networks (ANN) [BL88]. An ANN is defined as a data processing system consisting of a large number of simple, interconnected processing units. The architecture of ANN has been inspired by information processing structures found in the multilayered cerebral cortex of the human brain. Besides multilayered perceptrons, *radial basis function networks (RBFN)* are the most common ANN used for the approximation of functions. In particular the latter are typically used for interpolation.

Radial basis function networks [Gia02, BL88] are three layer fully connected feedforward networks (cf. Figure 2.6.4). They perform a nonlinear mapping ($\mathbb{R}^d \to \mathbb{R}^m$) from the d

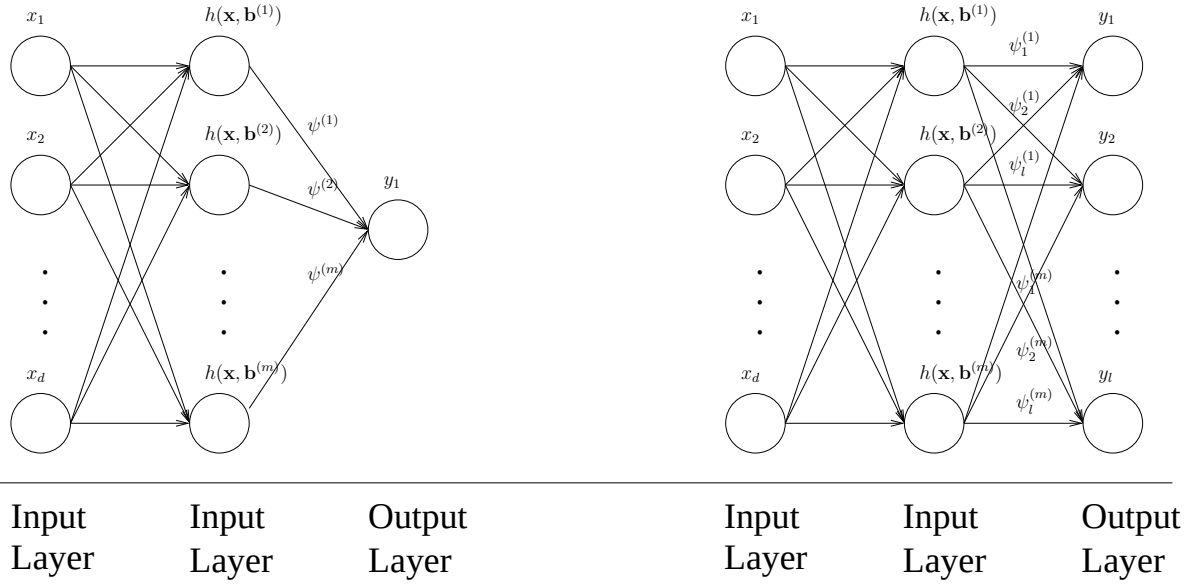


Figure 2.6.4: Visualization of a RBFN with single (left) and multiple (right) outputs.

inputs to the m hidden units followed by a linear mapping ($\mathbb{R}^m \to \mathbb{R}^l$) from the hidden units to the l outputs. For reasons of simplicity the typical case of $l = 1$ will be first considered and the multivariate case ($l > 1$) will be discussed later.

When applied for function approximation the neural network is trained in a *training phase* with data from known function evaluations. The weights of the linear function from the hidden layer to the output are adapted in a way that the deviations between the known output values to the predicted output values are minimized. Then, in the *prediction phase*, a point $\mathbf{x} \in \mathbb{R}^d$ is presented to the neural network and the neural network predicts the response.

Giannakoglou [Gia02] introduced a straightforward approach on how to employ RBF networks for function interpolation in the sense that results for points in the training set shall be reproduced exactly. It will be demonstrated that this kind of RBFN leads essentially to the same equations as they are used in the prediction step of Simple Kriging.

Its architecture is described as follows: Let again $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(m)}$ denote the evaluated points of the database, and $y^{(1)} = y(\mathbf{x}^{(1)}), \ldots, y^{(m)} = y(\mathbf{x}^{(m)})$. Then define for each evaluated point $\mathbf{x}^{(i)}$ a *RBF center*:

$$\mathbf{b}^{(i)} := \mathbf{x}^{(i)}, i = 1, \ldots, m \tag{2.6.28}$$

Let $|.| : \mathbb{R}^d \to \mathbb{R}_0^+$ denote a norm on $\mathbb{R}$ and $r : \mathbb{R}_0^+ \to \mathbb{R}_0^+$ a positive definite function on $\mathbb{R}_0^+$, then we define the activation function of the hidden layer via:

$$h(\mathbf{x}, \mathbf{b}^{(i)}) := r(|\mathbf{x} - \mathbf{b}^{(i)}|), i = 1, \ldots, m \tag{2.6.29}$$

The activation function based on r is called a *radial basis function* because it depends on the distance to the RBF center. For $r : \mathbb{R} \to \mathbb{R}$ Giannakoglou [Gia02] suggests the

function

$$r(\mathbf{x}) = \exp(-|\mathbf{x} - \mathbf{x}'|^q), \text{ with } q = 2 \tag{2.6.30}$$

As an alternative, a weighted distance measure $|\mathbf{x} - \mathbf{x}'|_\theta = \sum_{i=1}^{d} \cdot \theta_i \cdot |x_i - x_i'|$ for some θ_i provided by the user or derived from local gradients of f estimated by the RBFN, whether the RBFN becomes repeatedly trained [Gia02].

The function from the output values of the hidden layer to the output value of the RBFN is defined as a linear function with a-priori unknown weights:

$$\hat{y}(h^{(1)}, \ldots, h^{(m)}) = \sum_{i=1}^{m} \psi^{(i)} h(\mathbf{x}, \mathbf{b}^{(i)}) \tag{2.6.31}$$

The values of $\psi^{(i)}$ need to be adapted in the training phase. The output values of the training points have to be reproduced by the neural network, whenever we demand for exact interpolation of the results. This is expressed by the system of equations:

$$\sum_{i=1}^{m} \psi^{(i)} h(\mathbf{x}^{(j)}, \mathbf{x}^{(i)}) \stackrel{!}{=} y^{(j)}, j = 1, \ldots, n \tag{2.6.32}$$

Rewritten in matrix form this reads:

$$\underbrace{\begin{bmatrix} h(\mathbf{x}^{(1)}, \mathbf{b}^{(1)}) & \cdots & h(\mathbf{x}^{(1)}, \mathbf{b}^{(m)}) \\ \vdots & \ddots & \vdots \\ h(\mathbf{x}^{(m)}, \mathbf{b}^{(1)}) & \cdots & h(\mathbf{x}^{(m)}, \mathbf{b}^{(m)}) \end{bmatrix}}_{\mathbf{H}} \underbrace{\begin{bmatrix} \psi^{(1)} \\ \vdots \\ \psi^{(n)} \end{bmatrix}}_{\psi} \stackrel{!}{=} \underbrace{\begin{bmatrix} y^{(1)} \\ \vdots \\ y^{(m)} \end{bmatrix}}_{\mathbf{y}} \tag{2.6.33}$$

Note that $\mathbf{H}$ is a symmetric $m \times m$ matrix. The symmetry of the matrix $\mathbf{H}$ follows immediately from the equivalence of the RBF centers $\mathbf{b}^{(i)}, i = 1, \ldots, m$ with the input patterns $\mathbf{x}^{(i)}, i = 1, \ldots, m$ and the symmetry of the distance measure.

Assuming that there are no equal points in the database and that the RBF is positive definite, the weights $\psi^{(i)}, i = 1, \ldots, m$ are given by the solution of this system, i.e.

$$\psi = \mathbf{H}^{-1}\mathbf{y} \tag{2.6.34}$$

The correspondence of this approach to Simple Kriging is established, if we replace the correlation function c of the GRFM (Eq. 2.4.14) by the activation functions h in the RBFN. Under this condition we find $\mathbf{H} \hat{=} \mathbf{C}$ and $\psi \hat{=} \lambda$. As stated before, the special case of Kriging with an a priori given value of β is called Simple Kriging. Thus we can conclude that the prediction with the RBFN type introduced here is equivalent to the prediction of the mean value of the conditional distribution with Simple Kriging.

It might be claimed that the ANN approach has the advantage of multiple outputs and that all output values can be obtained within one training phase. Thus, we shall have a closer look at modeling multiple responses (cf. Figure 2.6.4) with the RBFN approach and with GRFM. Let

$$\mathbf{y}^{(i)} = \mathbf{y}(\mathbf{x}^{(i)}) := [y_1(\mathbf{x}^{(i)}), \ldots, y_l(\mathbf{x}^{(i)})]^T \tag{2.6.35}$$

denote the vector of responses for a single evaluation of $\mathbf{x}^{(i)}, i = 1, \ldots, m$. Moreover, let us define the $m \times l$ result matrix $\mathbf{Y}$ and the a priori unknown $m \times l$ weight matrix $\mathbf{\Psi}$ via

$$\mathbf{Y} = \begin{bmatrix} y_1^{(1)} & \cdots & y_l^{(1)} \\ \vdots & \ddots & \vdots \\ y_1^{(m)} & \cdots & y_l^{(m)} \end{bmatrix} \qquad \mathbf{\Psi} = \begin{bmatrix} \psi_1^{(1)} & \cdots & \psi_l^{(1)} \\ \vdots & \ddots & \vdots \\ \psi_1^{(m)} & \cdots & \psi_l^{(m)} \end{bmatrix} \tag{2.6.36}$$

such that $\mathbf{H} \cdot \mathbf{\Psi} = \mathbf{Y}$. Analogously to the single output case $\mathbf{\Psi} = \mathbf{H}^{-1} \cdot \mathbf{Y}$ describes the solution of this system, provided that the inverse exists, and via $\hat{\mathbf{y}} = \mathbf{\Psi} \cdot [h(\mathbf{x}, \mathbf{x}^{(1)}), \ldots, h(\mathbf{x}, \mathbf{x}^{(m)})]^T$ predictions for $\mathbf{y}(\mathbf{x}) \in \mathbb{R}^l$ can be obtained.

This algorithm is not equivalent to the prediction step of the simple co-Kriging algorithm [Mye92], since correlations between the output values are not considered. In fact it corresponds to the repeated prediction of different output values with the same correlation parameters and thus the same inverse correlation matrix $\mathbf{C}^{-1}$ in simple Kriging.

In conclusion, the RBF approach safes time, since the calibration phase is omitted. On the other hand we may loose accuracy, since the parameters of the activation function are not estimated from the sample. Over and above, standard implementations of the RBF do not provide us with any confidence information along with the prediction.

2.7 Prediction error

It would be interesting to know, which accuracy can be expected from a GRFM. In order to specify the maximal prediction error it makes sense to restrict the search space to an interval box of finite size, e.g. by $\mathbb{S} = [\mathbf{a}, \mathbf{b}]^d$ and define the error as $e_t = \max_{\mathbf{x} \in \mathbb{S}}(|y(\mathbf{x}) - \hat{y}_t(\mathbf{x})|)$, whereas $\hat{y}_t$ is the prediction based on t evaluations that are optimally placed in the search space, with regard to error minimization.

The so-called 'curse of dimension' states that in order to keep the error constant the number of sample points has to grow exponentially with dimension. As we will see, this general statement holds only, if certain assumptions about the function to be approximated are given.

The relationship between smoothness, dimension and approximation error can be studied for deterministic function classes, modeled by GRFM. A very general result is stated in Koehler et al. [KO96] and will be discussed next.

The Hölder class of functions with parameters k and α is defined as

$$F = \{f : [a, b]^d \to \mathbb{R} | \quad |D^{\mathbf{r}} f(x) - D^{\mathbf{r}} f(y)| = ||x - y||^{\alpha}\}, \\ \forall \mathbf{r} \text{ with } |r| = k, 0 < \alpha \leq 1. \tag{2.7.37}$$

Here we use here the notation of Schwartz for partial derivatives ([Joh81], pp. 54): $\mathbf{r} = (r_1, \ldots, r_d)$ is a n-tuple of non-negative integers with $|\mathbf{r}| = r_1 + \cdots + r_d$. The general partial differential monomial $D^{\mathbf{r}}$ is then defined as

$$D^{\alpha} := \frac{\partial^{|\mathbf{r}|}}{\partial x_1^{r_1} \ldots \partial x_d^{r_d}}. \tag{2.7.38}$$

For the Hölder class of functions it is possible to prove the lower bound for the maximal error e_t that we have to expect for an approximation based on t evaluated points in an search space $[0,1]^d$:

$$e_t \geq cn^{-(k+\alpha)/d} \tag{2.7.39}$$

Accordingly, the prediction quality of the metamodel not only depends on the dimension but also on the smoothness of the functions. Provided the parameters α and k are constant, the number of points that we need to guarantee a certain value of e_t growth exponentially with d.

If we stick more closely to the model assumptions of GRFM, and consider the function to be indeed a realization of a gaussian random field, the maximum of the error prediction $\hat{s}(\mathbf{x})$ for all $\mathbf{x} \in \mathbb{S}$ provides us with an upper confidence bound for e_t. Since the GRFM is assumed to be homoscedastic, the error variance is always limited by the global variance s^2. For highly correlated functions the distance of new points to known points governs the approximation error. Recall, that the correlation decreases exponentially with distance (expression 2.2.8). As with the same number of sample points the maximal distance between sample points and prediction points grows with the square root of the search space dimension d, also the correlation between the corresponding random variables decreases exponentially.

Recently, Büche et al. [BSK05] tested the effect of sampling size of metamodels in the context of metamodel-assisted evolutionary algorithms for optimization problems in the search space $\mathbb{R}^d$. They used local metamodels and recommended sample sizes ranging from $2d$ sampling points, for simple quadratic test problems, up to $8d$ sampling points, for some difficult multi-modal test problems. By these sampling sizes they achieved approximations that have a sufficiently good quality for providing predictions of satisfactory quality within the context of metamodel-assisted evolutionary algorithms. These recommendations were derived on typical test functions in the domain of evolutionary algorithms. All tests that led to these recommendations were carried out in 2–32 dimensions.

In conclusion, the results of this brief discussion shall highlight that the 'curse of dimension' can limit the use of approximation techniques to low dimensional spaces. However, the impact of the curse of dimensions can be weakened, if the functions considered are either very smooth or if the function value depends mainly on a small subset of active variables. Also, practical results on typical test problems may be taken into account, in order to find out about the sample size needed to achieve sufficiently well approximations.

2.8 Metamodel validation and diagnostics

Validation techniques can be used to assess the quality of a metamodel that has been trained on a given sample of points. In particular, the consistency of the given data with the model assumptions is verified.

For validating metamodels, we partition the set of known function evaluations $\mathbf{X}, \mathbf{y}$ into a training set and into a validation set. Then the metamodel is trained on the training set and predictions for points in the validation set are checked with the known results

from the validation set. A problem with this technique is that extra function evaluations are needed to form the validation set.

A technique that does not need extra function evaluations for checking the model is *leave-one-out cross-validation*. Let $\mathbf{X}_{-i}, \mathbf{y}_{-i}$ describe the training data set for which the i-th evaluation has been removed. This data set is used to train the metamodel. The predicted mean will be denoted with $\hat{y}_{-i}(\mathbf{x})$ and the predicted variance with $\hat{s}_{-i}(\mathbf{x})$, for any $\mathbf{x} \in \mathbb{S}$. Now, we compare the predicted mean $\hat{y}_{-i}(\mathbf{x})$ with $\hat{y}(\mathbf{x})$. This is done for any $i = 1, \ldots, m$.

Now, we can apply different plots in order to check the validity of the predictions.

First, we may ask, if the metamodel provides accurate estimations. Schonlau et al. [JSW98] suggest to use $y \sim \hat{y}$ plots for this purpose, plotting the predicted y values against the true y values. If the accuracy of predictions is good then all points in the $y \sim y$ plot are close to the bi-sector.

$$\epsilon_i = \sum_{i=1}^{m} (\hat{y}_{-i} - y_i)^2 \tag{2.8.40}$$

However, a low accuracy does not suggest that the model assumptions are wrong and have to be revised. It might also indicate the general difficulty of approximating a function due to low correlated data. It is thus also important to look for the self-assessment capabilities of the GRFM.

For the purpose of model validation it has to be tested, whether the errors are normally distributed with the predicted variance $\hat{s}(\mathbf{x})$. This can be done with the normal probability plot, which should be used to check the normal distribution hypothesis for the data set $\{\delta_1, \ldots, \delta_m\}$ with

$$\delta_i = \frac{\hat{y}_{-i}(\mathbf{x}_i) - y(\mathbf{x}_i)}{\hat{s}_{-i}(\mathbf{x}_i)}, i \in 1, \ldots, m. \tag{2.8.41}$$

It will turn out throughout this thesis that the error measure is often used to estimate the range for possible outputs for a given input. For checking, whether $\hat{y}_{-i}(\mathbf{x}) \pm \omega \cdot \hat{s}_{-i}(\mathbf{x})$ is a good confidence range for the true output or not, we can simply apply the $y \sim \mathrm{lb}_\omega$ diagram and the $y \sim \hat{ub}_\omega$ diagram. Within these diagrams we plot all true values against the predicted lower bounds (or the predicted upper bounds in case of the $y \sim \hat{\mathrm{ub}}_\omega$ plot). This shows us, whether the confidence bounds are valid with the confidence level, depending on ω.

2.9 Conclusions

GRFM metamodels have been introduced in this chapter. It has been shown how these models can be used for the prediction of single and multiple outputs. The runtime complexity of different procedures (calibration, training, prediction) has been investigated in detail. The main factor in the running time of Kriging is the number of training points. The results suggest that for large numbers of evaluations it is important to reduce the number of samples, before constructing the metamodel. This can be done, for example,

by only considering neighboring solutions of the search point that has to be predicted. Such techniques were termed local metamodels.

Moreover, the relationship between the GRFM approach and the RBFN approach for metamodeling has been explicated. It turned out that the commonly applied RBFN approach, which sets the RBF centers equal to the training patterns, can be mapped to the 'simple Kriging' approach by replacing the radial basis functions by correlation functions of GRFM and omitting the calibration step in the Kriging algorithm. Because of this formal equivalence, empirical comparisons of both approaches are dispensable. However, if we want to make use of the calibration of correlation parameters of GRFM, we need to compute more than one matrix inversion for building the GRFM. In this case we spend more time for building the model than in the case of simple RBFN and – in case of multiple outputs – also the time for training increases.

Finally, we reported on some results on the estimation of the prediction error and pointed out that besides the problem dimension also its smoothness governs the sample size needed to guarantee a certain prediction error. Moreover, cross-validation, as well as the use of $y \sim \hat{y}$ and $y \sim \mathrm{lb}_\omega$ diagrams were proposed for measuring the validity of a metamodel online or in practical experiments.

3 Continuous single-objective optimization

By asking for the impossible we obtain the possible.

Italian proverb

This chapter discusses the capabilities and limitations of state-of-the-art techniques for the solution of real valued single-objective optimization problems:

$$f(\mathbf{x}) \rightarrow \min, \mathbf{x} \in \mathbb{S} \subseteq \mathbb{R}^d \tag{3.0.1}$$

Due to the large number of techniques available for continuous optimization, our survey will necessarily be incomplete. Our coverage will focus on techniques that we consider as relevant within the context of this thesis. In particular, we introduce techniques that work with approximation models. For a broader overview of optimization techniques the reader is referred to [CDG99, GMW81, HP95] and [Sch95].

Firstly, in section 3.1 the black-box complexity of continuous box constrained optimization is discussed. Secondly, a sketch of four main methodologies for approximately solving the global optimization problem is given, all of which are closely related to the algorithmic approach proposed in this work. The survey starts with the discussion of *gradient based methods* (section 3.2). It continues with deterministic direct search methods, focussing on *pattern search* (section 3.3). *Bayesian global optimization methods* are discussed in section 3.4. Last but not least in section 3.5 we give a brief overview on *bio-inspired* and *stochastic methods* for optimization. There, we will focus on *evolution strategies* (ES) that provide the algorithmic framework for the methods discussed in the subsequent chapters.

3.1 Black box complexity of global optimization

Before introducing different algorithms to tackle single criterion optimization problems, it shall be motivated from a theoretical point of view, why heuristic procedures are often needed for the global optimization of black box functions. The concept of *information based complexity* or *black box complexity* will be introduced first, since it fits well within the context of optimization with time consuming computer experiments. The discussion will first provide some remarks on general optimization of regular functions and then focus on some general conditions of continuity that allow to get sharper bounds for estimating the information based time complexity of the optimization problem. These results also

limit what we might expect from some kind of general purpose optimization tool in the continuous domain. Though the results presented here are not new, they have rarely been discussed within the context of heuristic optimization.

In industrial design, computer experiments are usually very time consuming. Thus, if computer experiments are scheduled in order to perform a design optimization, the running time of the algorithm is mainly determined by the running time of the computer experiments. In these cases, the number of objective function evaluations is a good measure for the running time of an algorithm.

The *black box complexity* (or *information based complexity* [Nov99]) is defined as the asymptotic number of function evaluations that are needed to determine approximations to the global optimum with a certain precision. The term is also used in the context of approximation, where it is desired to achieve a certain approximation quality.

In the following an algorithm will be denoted by '$\mathtt{a}$' and it is assumed that it generates a sequence of points $\mathbf{x}_1^{\mathtt{a}}, \ldots, \mathbf{x}_t^{\mathtt{a}}$ and evaluates them by obtaining their function values $f(\mathbf{x}_1^{\mathtt{a}}), \ldots, f(\mathbf{x}_t^{\mathtt{a}})$. From this information an approximation for the global optimum is determined, usually by determining $\mathbf{x}_{opt}^{\mathtt{a}}(t) = \arg\min\{f(\mathbf{x}_1^{\mathtt{a}}), \ldots, f(\mathbf{x}_t^{\mathtt{a}})\}$. Generally, a distinction can be made between non-adaptive algorithms that determine all t sample locations a-priori and adaptive algorithms that make use of the $i-1$ previous function evaluations in order to determine the i-th sample location [NR96] for $i = 1, \ldots, t$.

According to Novak and Ritter [NR96], it is easy to construct algorithms that converge to the global optimum for functions f, which fulfil the property that for every positive ϵ the set

$$\{\mathbf{x} \in [\mathbf{x}^{\min}, \mathbf{x}^{\max}] | f(\mathbf{x}) < \inf_{\mathbf{x} \in [\mathbf{x}^{\min}, \mathbf{x}^{\max}]} f(\mathbf{x}) + \epsilon\} \tag{3.1.2}$$

contains an open set. Among others, this class of functions contains all continuous functions.

The sole condition that needs to be assured is that the sequence of points $\mathbf{x}_t^{\mathtt{a}}, t = 1, 2, \ldots$ generated by the algorithm $\mathtt{a}$ is dense in $[\mathbf{x}^{\min}, \mathbf{x}^{max}]$. This can be easily confirmed for methods working with grid refinement.

For Monte Carlo methods and other stochastic methods a similar result can be obtained [NR96] for convergence to the global optimum with probability of almost one. It suffices to show that for each positive ϵ the ϵ-ball

$$B_\epsilon(\mathbf{x}) := \{\mathbf{x}' | \ |\mathbf{x} - \mathbf{x}'| < \epsilon\} \tag{3.1.3}$$

around any search point $\mathbf{x}$ is sampled with a probability $p \geq p^{\min} > 0$ at least after each $n_0 < \infty$ samples. In this case a lower bound for the probability $p_{\epsilon,n_0}(t)$ that a sample is placed in the ϵ-ball around the optimum is given by

$$1 - (1 - p^{\min})^{t/n_0} \leq p_{\epsilon,n_0}(t) \leq 1 \tag{3.1.4}$$

and hence

$$\lim_{t \to \infty} (p_{\epsilon,n_0}(t)) = 1 \tag{3.1.5}$$

However, in practical optimization we are also interested in the number of objective function evaluations (t) needed to achieve a certain precision $\Delta(\mathtt{a}_t, f) = f(\mathbf{x}_{\mathtt{a}}^*(t)) - f^*$

for the approximation of an optimum. Meaningful bounds below infinity can only be achieved, if we make further assumptions on the black box function: Let $F : \wp([0,1]^d \rightarrow \mathbb{R})$ denote a class of functions, then

$$\Delta(\mathtt{a}_t, F) = \sup_{f \in F} \Delta(\mathtt{a}_t, f) \tag{3.1.6}$$

is the approximation precision that can be guaranteed for any function in F with t objective function evaluations when using algorithm '$\mathtt{a}$'.

More generally, we can ask for the maximal precision that can be achieved with any algorithm for a class of functions:

$$e_t(F) = \inf_{\mathtt{a} \in \mathcal{A}} \Delta(\mathtt{a}_t, F) \tag{3.1.7}$$

where $\mathcal{A}$ denotes the set of all possible algorithms.

For many function classes it is possible to prove the lower bound of this function. A very general class of functions is the Hölder class of functions (expression 2.7.38 on page 26).

For the Hölder class of function with parameters α and k a sharp error bound has been discovered [NR96], that is

$$e_t(F) \geq cn^{-(k+\alpha)/d} \tag{3.1.8}$$

or, if we are interested in the number n of objective function evaluations needed to achieve the approximation error $e_t(F)$:

$$n = \left(\frac{c}{e_t(F)} \right)^{d/(k+\alpha)} \tag{3.1.9}$$

The result demonstrates, that even for this *continuous* class of functions the number of objective function evaluations that is needed to achieve a certain approximation accuracy grows exponentially with the number of dimensions. This observation is often referred to as the '*curse of dimension*'. The effect of the dimension can be compensated to a certain extent, if F contains only very smooth functions. The latter is an important observation that has often been neglected in the literature. It provides us with reasonable hope that even for high dimensional spaces good approximations of global optima can be obtained, whether the partial derivatives of the function are bounded by small values compared to the size of the search space. In this context, it shall be noted that many simulation systems in physics and engineering are based on nonlinear partial differential equations. Thus, it is sometimes easy to obtain bounds for the derivatives, even though the solution of the equations is difficult.

Note, that this 'curse' also weighs on the problem of function approximation we stepped across in chapter 2 (page 26). Accordingly, the problem of black box optimization and approximation are closely related and their maximal performance seems to be limited by the same properties of the function. Note, however, that we have nothing said yet about the operational run time complexity (cf. [Nov99]) of optimization algorithms, which still might grow exponentially wo

Even, if the black box complexity is restricted by the bound given above, the operational run time complexity of the optimization algorithm may grow even faster than the black-box complexity with the number of dimensions d. It has been stated by Ritter and

Novak [NR96] that sampling on a regular grid method is among the asymptotically best methods for solving this problem, if we consider the worst case of the approximation error. However, in the average case[1] adaptive methods can outperform such simple strategies [NR96].

The error bound in expression 3.1.8 can be achieved with non-adaptive grid methods. However, adaptive methods outperform non-adaptive methods in the average case. This has been studied by Ritter [Rit90]. In special cases like Lipschitz optimization and the optimization of convex quadratic functions much better error bounds can be obtained (cf. [NR96]).

Expression 3.1.9 suggests that for several relevant classes of high dimensional or low correlated functions there exists no algorithm that guarantees to find a sufficiently precise approximation of the global optimum in polynomial time. This is, why the practitioner has to rely upon heuristic optimization algorithms. These algorithms might not find the global optimum but they are more or less sophisticated strategies for improving the quality of a solution measured by the objective function value.

Heuristic search methods range from *path oriented methods* that converge quickly to a nearby local optimum to *volume oriented methods* that do a *coarse sampling* of the search space. As we will see in the following, pattern search methods and evolution strategies are some kind of compromise between these two paradigms. They start with a coarse sampling of the search space and adaptively refine the sampling in promising regions of the search space.

3.2 Gradient based optimization methods

Path oriented or local search methods can be defined by a general iterative formula:

$$\mathbf{x}_{t+1} = \mathbf{x}_t + \sigma_t \mathbf{d}_t \tag{3.2.10}$$

Here $\mathbf{x}_t$ is the vector of design variables, σ_t denotes a step size, and $\mathbf{d}_t$ a direction vector at time step t. The subsequent input vector $\mathbf{x}_{t+1}$ is considered to have a better objective function value. Note, that for the determination of $\mathbf{d}_t$ and σ_t a limited number of trial evaluations of the objective function are conducted. If the local optimization method is successfully applied, the series $(\mathbf{x}_t)_{t=1,2,\ldots}$ converges to the optimal solution. It is said that $(\mathbf{x}_t)_{t=1,2,\ldots}$ describes a path of points with decreasing objective function values to the optimum, why these methods are often referred to as *descent methods*.

Gradient based methods make use of the gradient of the objective function

$$\nabla f(\mathbf{x}) = (\frac{\partial f(\mathbf{x})}{\partial x_1}, \ldots, \frac{\partial f(\mathbf{x})}{\partial x_d})^T \tag{3.2.11}$$

in order to determine the step direction $\mathbf{d}_t$ in Eq. 3.2.10.

In cases, where the gradient cannot be provided by the evaluation tools, it may be approximated by a finite difference method from $f(\mathbf{x})$ plus n or $2 \cdot n$ objective function evaluations at small perturbations of $\mathbf{x}$ in all coordinate directions [GMW81].

[1] Note, that this raises the question of how to define some average case. For details on average case analysis of numerical problems on continuous functions we refer to Ritter [Rit90].

The most straightforward gradient based minimization method is the steepest descent method. In the steepest descent method the direction $\mathbf{d}_t$ in Eq. 3.2.10 is substituted by the normalized negative gradient:

$$\mathbf{d}_t = -\frac{\nabla f(\mathbf{x}_t)}{||\nabla f(\mathbf{x}_t)||} \tag{3.2.12}$$

The step size is then chosen as the result (so-called relative minimum) of an one dimensional optimization in this direction (so-called line search):

$$\sigma_t = \arg\min_{\sigma>0} f(\mathbf{x}_t + \sigma \mathbf{d}_t) \tag{3.2.13}$$

Another possibility for choosing σ_t is to start with an arbitrary step size and reduce or extend the step size by certain rules that are applied in each iteration, according to the success or failure of previous trial steps [Sch95].

Although the concept of steepest descent looks appealing at first glance, this method has some drawbacks. Besides the disadvantage that convergence to the optimum of f can only be guaranteed for strictly convex and differentiable functions there is another problem: If there is strong interaction between the variables, the strategy runs into danger to perform a 'zig-zag' course and therefore many line searches will be needed to approximate the local optimum sufficiently well.

The latter drawback can be offset by considering second order information in the search. These methods exploit also pieces of information from second order derivatives in order to determine the search direction.

Second order methods build a local quadratic model of the objective function by means of its Taylor expansion

$$f(\mathbf{x}) \approx f(\mathbf{x}_t) + \nabla f(\mathbf{x}_t)(\mathbf{x} - \mathbf{x}_t)^T + \frac{1}{2}(\mathbf{x} - \mathbf{x}_t)^T \nabla^2 f(\mathbf{x}_t)(\mathbf{x} - \mathbf{x}_t)^T \tag{3.2.14}$$

Then, $\mathbf{x}_{t+1}$ is set to the optimum of this approximation. This can be determined by using Newton's method in order to find the zero of the gradient, supposed that the Hessian matrix is positive definite (i. e. the problem is strictly convex). Thus we get the Newton Raphson optimization strategy:

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \nabla f(\mathbf{x}_t)[\nabla^2 f(\mathbf{x}_t)]^{-1} \tag{3.2.15}$$

In practical applications the objective function is often far from being quadratic or convex, why it might be wise to limit the length of a trial step. This can be done by replacing $\mathbf{d}$ in Eq. 3.2.10 by the second factor in equation 3.2.15 and perform like it has been described for the steepest descent method. In order to decrease the cost of the approximation of the inverse Hessian matrix, it is also a common strategy to approximate it from a succession of gradient approximations in gradient based approximation. This is the main idea in the so-called *quasi Newton methods* like Stewart's modification of the Davidon Fletcher Powell method (DFPS) (cf. [Sch95], pp. 78) or by conjugate directions and conjugate gradients methods [Sch95]. Note, that the DFPS method approximates the local gradient

by forward differences or central differences. Thus the DFPS method is a derivative-free method that can be applied in cases where gradient values cannot be obtained from the black box evaluator.

Recent Newton optimization methods determine a so-called trust region [CGTT00], i .e. a local environment of the current search point in that the predictions are being trusted. The radius of the trust region is adjusted during the optimization run by comparing the objective function value at the predicted optimum with the predicted objective function value.

In several studies ([Sch95, ESB02]) it has been demonstrated that Newton methods and quasi-newton methods like DFPS are not capable of dealing with complex search spaces involving multimodalities and discontinuities. On the other hand, they perform very well on quadratic and near quadratic functions.

3.3 Deterministic direct search methods

Besides gradient based methods various derivative-free optimization methods have been frequently used for the optimization with computer models. A large number of those derivative-free methods belong to the class of *direct search methods*. Hooke and Jeeves were the first, who used the term "direct search methods". In a publication that was released in 1961 they defined it as follows [HJ61]:

> *We use the phrase 'direct search' to describe sequential examination of trial solutions involving comparisons of each trial solution with the best obtained up to that time together with a strategy for determining (as a function of earlier results) what the next trial solution will be. The phrase implies our preference, based on experience, for straightforward search strategies which employ no techniques from classical analysis except when there is a demonstrable advantage of doing so.*

The origin of many direct search methods dates back to the late fifties and early sixties, when there was some kind of "boom" for constructing numerical optimization methods. This was closely related to the upcoming of digital computers and the availability of the first computer models. For a survey on classical direct search methods that have been originated in those days the interested reader is referred to [GMW81] and [Sch95]. A brief overview is given also in [VT00] and [ESB02].

One reason for the success of many direct search methods is that they have been invented to a time, where gradient based methods were suffering from much more serious drawbacks than they do today and a theoretical foundation of these algorithms had not yet been established. However, despite the significant progress in gradient based optimization, direct search method still have their regular place in the optimization of computer models. A reason for this is certainly the simplicity and robustness of some of these methods. It does not require an expert in numerical optimization to implement methods like the downhill simplex or the method of Hooke and Jeeves [HJ61], and to gain a basic insight into their search principles and how to control them.

But also for experts in numerical optimization it is justified to consider these methods. The fault tolerance and the less rigid iteration schemes of the direct search methods that are used today, make them easy to apply for automatic optimization in parallel computing environments that have become very apparent in research and development. Moreover, they are considered to be less sensitive to numerical noise due to rounding errors and discontinuities – both being difficulties that are frequently encountered when optimizing with computer experiments.

The theoretical foundations of direct search methods have been further developed since the early 60ties and, today for many constrained and unconstrained problem classes the convergence of these methods to a local minimum has been proven [Tor97]. In recent years also the concept of *generalized pattern search (GPS)* has been developed [VT00] that allows to integrate many direct search methods into a common framework and to derive a unified theory for these methods (cf. [AD00]).

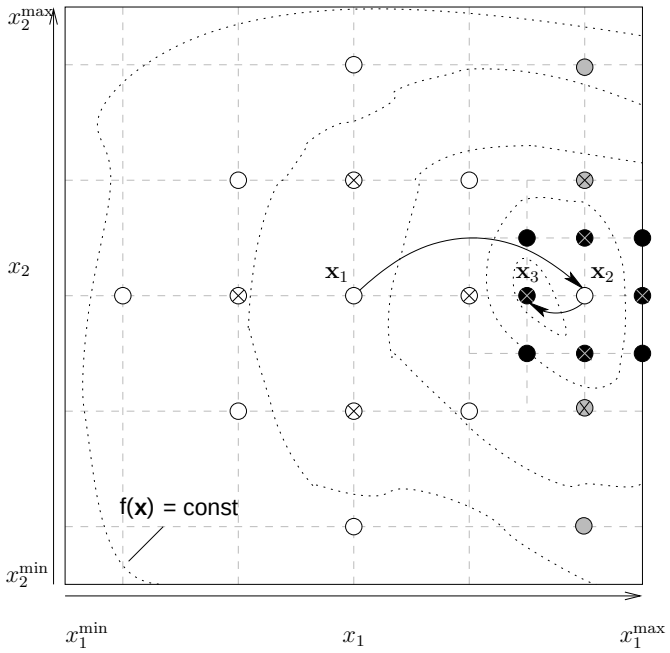


Figure 3.3.1: Visualization of some search steps of an algorithm that fits into the generalized pattern search (GPS) framework. In the example the GPS algorithm has to obtain an optimum in a two dimensional search space $[\mathbf{x}^{\max}, \mathbf{x}^{\min}] \subset \mathbb{R}^2$. The trial points for the first step are the white filled points placed around the search point $\mathbf{x}_1$. The largest improvement among these trial points due to f is $\mathbf{x}_2$. Thus $\mathbf{x}_2$ serves as the next search point. For $\mathbf{x}_2$ again trial points are placed on the mesh in form of a diamond, excluding those points that have already been evaluated in the first step or that extend the search space boundaries. Since no improvement has been found the step size is halved and a new pattern of trial points is generated for a refined mesh. Among the points on the refined mesh, an improvement $\mathbf{x}_3$ has been found and it is taken as the new search point. Note that the cross-marked points belong to the core pattern of one of the search space, which is a minimal sub-pattern that is important to assure stationary point convergence.

Within methods that fall into the framework of generalized pattern search, trial points are placed on selected nodes of a d-dimensional grid around the current search point $\mathbf{x}_t$. Then these points are evaluated using the objective function f. If an improvement is found, a

new iteration starts with points placed around a new search point $\mathbf{x}_{t+1}$ that is set to the trial point where the improvement has been obtained. Depending on the implementation of the pattern search method, not necessarily the first improvement found on the pattern needs to be the starting point of the next iteration. Furthermore, there can be significant differences in the order in that search point are evaluated. If no improvement has been found among all trial points of the basis (a subset of the selected grid points including trial points in different coordinate directions), the grid size can be refined by dividing the mesh size by a constant factor. It is definitely refined if none of the points on the pattern led to an improvement.

It has been proven that, if the set of trial points meets the condition that a set of difference vectors between a subset of the trial points and $\mathbf{x}_t$ is a positive basis of $\mathbb{R}^d$, the pattern search method converges to a stationary point, i. e. a point $\mathbf{x}$ with $\nabla f(\mathbf{x}) = \mathbf{0}$, for any differentiable objective function and $t \to \infty$. Practically speaking, this means that pattern search will converge either to a local optimum or to a saddle point for any differentiable function. A proof and a more detailed discussion of this result and a further discussion of convergence properties of generalized pattern search can be found in [Tor97]. For box constrained search spaces further requirements for the set of trial points need to be fulfilled as it has been demonstrated by Lewis et al. [LT99].

Example: Figure 3.3.1 visualizes three search steps within a pattern search algorithm that works with a diamond-shaped 2-D pattern. First trial points for $\mathbf{x}_1$ are determined by the algorithm (the white filled points). They are all placed on a regular pattern (in the example this has the form of a diamond) around the search point $\mathbf{x}_1$. Among the trial points are the points that belong to the core pattern (the crossed white circles). Now, all trial points are evaluated and the pattern search moves to the largest improvement (here it is $\mathbf{x}_2$) and a new pattern of trial points is generated, excluding those points that have already been evaluated. Among the new trial points placed around the new search point $\mathbf{x}_2$ no improvement can be found. Thus, the mesh is refined by halving the mesh size. On the refined mesh around $\mathbf{x}_2$, again trial points are placed in form of a diamond. Again all new points are evaluated. This time with $\mathbf{x}_3$ an improvement has been obtained and it will serve as the new search point. □

Pattern search methods are regarded to converge slower than gradient based methods (for simple nearly quadratic functions and in terms of objective function evaluations), but they are also considered to be much more robust and flexible (cf. [ESB02]) than the latter. In order to accelerate direct search methods, metamodeling techniques have been proposed [TT97, DV97].

The working principle of *metamodel-assisted pattern search* is to evaluate search points in the sequence that is suggested by rank ordering obtained for the approximations provided by a Kriging metamodel. It is easy to prove, the circumstance that the stationary point convergence is still satisfied for these methods. An empirical investigation of metamodel-assisted pattern search techniques can be found in [Sie00].

3.4 Bayesian global optimization

Several global optimization methods have been proposed in literature. Due to Dixon and Szegö [DS78] these algorithms distinguish from local optimization algorithms, as they do not aim at finding a single local optimum but at finding the best among several local optima. Törn et al. [TZ89] developed several multi-start clustering techniques for that purpose. These algorithms run several local search procedures in different parts of the search space. These parts are identified using cluster analysis techniques. Though it yields high quality results, this approach needs a large number of objective function evaluations and thus it is not applicable for optimization with expensive objective function evaluations.

Zilinskas and Mockus [TZ89] developed *bayesian global optimization (BGO)* procedures that are often used for the optimization with time consuming computer experiments.

Within BGO the random field metamodels are employed to model the response surface from objective function evaluations of points sampled in the search space. A 'figure of merit' is used in order to decide which points have to be evaluated. This figure of merit could be the expected outcome of the experiment for an unknown point. It might also take into account the local variance of the prediction in order to re-sample less explored regions or to find regions with high potential for improvements.

However, it is common practice in the BGO literature to suggest the use of error measures along with the predicted response in order to define ranking criteria. A straightforward approach to BGO termed *statistical global optimization (SGO)* has been suggested by Cox and John [CJ97].

Algorithm 1 Statistical global optimization

1: $D_0 \leftarrow \mathbf{evaluate}_f(\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(m_0)})$ {Initialize database}
2: $t \leftarrow m_0$ {Initialize evaluation counter}
3: **while** $t < t_{eval,max}$ **do**
4: Search for $\mathbf{x}_t^* = \text{argmin}_{\mathbf{x}\in\mathbb{S}} \hat{f}_{sc}(D_t, \mathbf{x})$
5: $y_t = f(\mathbf{x}_t^*)$
6: **if** $y_t < y_{\min}^t$ **then**
7: $\mathbf{x}_{\min}^t = \mathbf{x}_t^*$
8: $y_{\min}^t = y_t$
9: **end if**
10: $D_{t+1} = D_t \cup \{(\mathbf{x}_t^*, y_t)\}$
11: **end while**
12: **return** $y_{\min}^t, \mathbf{x}_{\min}^t$

In algorithm 1 an outline of the SGO algorithm is given. The algorithm starts with evaluating a user defined number of design points evenly distributed in the design space. Then repeatedly the following steps are proceeded: First the minimum of a *utility criterion* $\hat{f}_{sc}$ is searched for on the metamodel. The minimizer is then evaluated precisely by means of an computer experiment and used in order to update the database of objective function evaluations that are used for metamodeling.

3.4.1 Utility functions in bayesian optimization

Different criteria for the utility criterion $\hat{f}_{sc}$ have been suggested for pre-screening the search space by means of the metamodel (line 4). They all more or less refer to the trade-off already described by Kushner [Kus62]:

> *The purpose of the utility function is to find trade-off between sampling in known promising regions versus sampling in under-explored regions or regions where the variation in function values is high.*

If the focus is put to the promising regions and the degree of exploration is not considered, the expected function value

$$\hat{f}_{sc}(\mathbf{x}) = \hat{y}(\mathbf{x}) \tag{3.4.16}$$

would serve as a good criterion. Also, this allows us also to use metamodels that are not capable of providing an estimation for the approximation error.

Cox and John [CJ97] suggested a compromise between the exploration and the exploitation objective by proposing the criterion

$$\mathrm{lb}_\omega(\mathbf{x}) = \hat{y}(\mathbf{x}) - \omega \cdot \hat{s}(\mathbf{x}), \omega > 0 \tag{3.4.17}$$

for $\hat{f}_{sc}(\mathbf{x})$. A visualization of this criterion can be found in Fig. 3.4.2. The lb_ω criterion looks very appealing since it takes into account the expected function value as well as the confidence factor attached to this value. The factor ω can be adjusted in order to balance between global exploration and local search.

A high value of ω rewards regions in the search space that are relatively unexplored, which entails a high value for $\hat{s}$. Driving the search to unexplored regions, can help to escape from local optima in multimodal optimization and thus increases the robustness of the approach. On the other hand, a greedy approach would focus on a high progress within the next step and thus samples values with the minimal expected values. This might be faster but entails a higher risk of convergence to a local optimum. The influence of the mean value in expression 3.4.17 is, of course, highest whenever ω takes values close to zero.

The expression of 3.4.17 can be interpreted as a one-sided (lower) confidence bound. Accordingly, we will use the abbreviation lb_ω. In this context, the adjustment of ω can be related to a confidence level quantifying the probability that the true output value is above the lower confidence bound, i. e. $p_\alpha = \Pr(\mathrm{lb}_\omega(\mathbf{x}) > y(\mathbf{x}))$ provided the model assumptions are true. For GRFM the relationship between p_α and ω can be expressed by $p_\alpha = \Phi(-\omega)$ where Φ denotes the cumulative gaussian distribution.

Mockus et al. [MTZ78] proposed a criterion based on the expected improvement. Jones et al. [JSW98, SWJ98] further developed algorithms based on this criterion. The expected improvement criterion is defined as follows: Let $y^t_{\min} = \min\{y(\mathbf{x}_i)|i = 1, \ldots, t\}$ denote the best so-far found result after t objective function evaluations. Moreover, let $\mathbf{x}$ denote a potential new search point considered for evaluation. Then the improvement at this new search point is measured by:

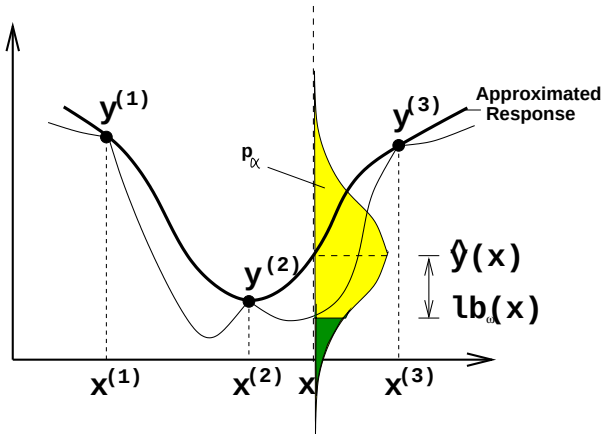


Figure 3.4.2: The figure visualizes the lb_ω criterion $\mathrm{lb}_\omega(\mathbf{x}) = \hat{y}(\mathbf{x}) - \omega\hat{s}(\mathbf{x})$ for a point $\mathbf{x} \in \mathbb{S}$. By means of the confidence factor ω the width of the confidence interval can be adjusted, in order to achieve a desired confidence level.

$$\mathrm{I}(y) = \begin{cases} 0 & \text{if } y > y^t_{\min} \\ y^t_{\min} - y & \text{otherwise} \end{cases} \tag{3.4.18}$$

Now, the expected improvement is defined as:

$$\mathrm{ExI}(\mathbf{x}) = \int_{-\infty}^{y^t_{\min}} (y^t_{\min} - y) \cdot \varphi(\frac{y - \hat{y}(\mathbf{x})}{\hat{s}(\mathbf{x})}) dy$$

This expression can be simplified by algebraic operations to

$$\mathrm{ExI}(\mathbf{x}) = (y^t_{\min} - \hat{y}) \cdot \Phi(\frac{y^t_{\min} - \hat{y}(\mathbf{x})}{\hat{s}(\mathbf{x})}) + \hat{s}(\mathbf{x}) \cdot \varphi(\frac{y^t_{\min} - \hat{y}(\mathbf{x})}{\hat{s}(\mathbf{x})}) \tag{3.4.19}$$

In this expression φ denotes the probability density function of the standard gaussian distribution. In their paper Jones et al. [JSW98] provide a deterministic procedure how to determine the minimum of the expected improvement, based on Floudas' α-branch and bound algorithm for nonlinear optimization ([JSW98], pp. 27). However, they state that this algorithm is only applicable for small data-sets. The resulting BGO they termed *efficient global optimization (EGO)*. They conjectured that – provided the θ_i in the gaussian model (cf. expression 2.2.8) remain strictly positive – the EGO algorithm creates a dense subset within the search space ([JSW98], page 25) and thus converges to the global optimum of continuous functions. Empirical studies of EGO suggest a good performance especially for low dimensional search spaces and low numbers of evaluations.

There are many other BGO algorithms using different kinds of iteration schemes and criteria for selecting subsequent search points. Also a lot of work has been spent on how to design the initial sample. For an overview of BGO methods the interested reader is referred to [TZ89] and to [SWN03].

3.5 Bio-inspired optimization algorithms

Like in the field of approximation with artificial neural networks, also in the field of optimization it proved to be successful to build heuristic algorithms inspired by abstractions of collective processes observed in nature. The general idea of bio-inspired optimization has been extensively exploited in the last decades in order to build more or less specific heuristics for black box optimization, like particle swarm optimization [KES01], genetic algorithms [Gol89] and ant colony algorithms [DMC99]. Moreover, there are many stochastic heuristic optimization methods like *simulated annealing* and *tabu search* [CDG99], that are not directly motivated from biological systems but work with similar mechanisms. It would extend the scope of this work to give a comprehensive overview of all these heuristics. Instead, we recommend the books [CDG99] and [ZM00] as surveys on ideas in bio-inspired and related heuristic optimization methods.

Many bio-inspired optimization algorithms belong to the class of *evolutionary algorithms* (EA). These algorithms have been inspired by theories of evolution, in particular by those theories that are referred to as the *modern synthesis* [Dob37] that unifies the natural selection theory of Darwin and Wallace with Mendel's theory of genetics. Within the modern synthesis, *evolution* is defined as a change in the frequency of an allele within a gene pool (or population of genes). This change may be caused by mechanisms like natural selection, genetic drift or changes in population structure (gene flow), etc. Common evolutionary algorithms are genetic algorithms (GA), evolutionary programming (EP) and evolution strategies (ES). For an comprehensive survey of classical evolutionary algorithms we refer to [BFM97], thereby noting that the development of evolutionary algorithms is still an field of extensive research.

It shall be remarked here that the view of evolution as 'survival of the fittest' is way to simple to account for many evolutionary adaptation processes in nature, though it applies well to some simple designs for evolutionary algorithms that are used for optimization purposes. But even there it turned out that elitism is not always the best way of how to achieve good solutions in the long term [OBS98, Sch02]. For a rich source of material on theories of natural evolution the interested reader is referred to [Pag02].

Within this work, EA are designed as randomized search heuristics for tackling difficult black box optimization problems and not in the first instance as models of natural processes. Therefore, it requires caution to translate the results back into a biological context.

3.5.1 Evolution strategies

In this work Evolution Strategies (ES) will provide the algorithmic framework for the optimization studies on single-objective problems. After explaining why this subclass of an evolutionary algorithm has been chosen, a detailed outline of the (μ, κ, λ)-ES will be given. Later we will show how this algorithm can be supported by metamodels. Note that the metamodel-assistance techniques developed for the ES can be easily transferred to similar EA incarnations that work with rank based selection schemes.

Evolution Strategies have been chosen, since they have proven to be very robust for opti-

mization of high dimensional ($d \gg 7$) continuous optimization problems. Another reason, why ES have been chosen as basic algorithms is that, unlike other commonly used EA like for example genetic algorithms, ES have been extensively studied on continuous search spaces. However, they were also used for other problem domains, like mixed integer optimization [ESGG00] or under certain assumptions for general metric spaces [Wie01]. Modern incarnations of ES incorporate most of the typical features of EA like rank based selection, population based search and the usage of mutation and recombination operators. There are also variants that are working on structured population models. A main feature of ES is their capability to adapt control *strategy parameters* such as mutation step sizes online, which makes them user friendly and allow for a gradual refinement of search.

It will be impossible to present a comprehensive survey of the results that have been found in more than 40 years of research on evolutionary strategies. For further discussion of this algorithm the reader is referred to literature: A comparison of classical EA compared to ES for parameter optimization can be found in [Bäc96]. The theory of evolution strategies has been described by Beyer [Bey01]. For a recent overview on ES the reader is referred to [BS02]. Empirical investigations of the ES variants discussed in this work can be found in Kursawe [Kur99]. For a comparison to classical search strategies the interested reader is referred to Schwefel [Sch95] and within the context of industrial design optimization to Emmerich et. al. [ESB02].

Next, the (μ, κ, λ)-ES as it is used throughout this work shall be outlined in detail. Algorithm 2 specifies the algorithmic framework for the design of a (μ, κ, λ)-Evolution Strategy for a general search space. The following notations are used for the description of the algorithm: $P = \{\mathbf{a}^{(1)}, \dots \mathbf{a}^{(m)}\} \in \mathbb{I}^m$ denotes a *population* of m *individuals*. Here, $\mathbb{I}$ denotes the space of individuals. An individual

$$\mathbf{a} = (\mathbf{x}, \mathbf{s}, y, k) \in \mathbb{S} \times \mathbb{R}^{n_s} \times \mathbb{F} \cup \{\oslash\} \times \mathbb{N}_0^+ \tag{3.5.20}$$

consists of the search point $\mathbf{x} \in \mathbb{S}$, the components of which are called *object variables* and a vector of n_s *strategy parameters* $\mathbf{s} \in \mathbb{R}^{n_s}$. Moreover, a set of function values y (the results of an evaluation) can be stored in the individual along with its age k, i. e. the number of generations it has survived. In single criterion optimization this value is the result of the objective function evaluation, i. e. $y = f(\mathbf{x})$.

Algorithm 2 General Framework for an Evolutionary Strategy

1: $t \leftarrow 0$
2: $P_0 = \mathbf{evaluate}(\mathbf{init}_\mu())$
3: $\mathbf{x}_{best}^0 \leftarrow \arg\min(\{f(\mathbf{x}) | \mathbf{x} \in P_0\}); f_{best}^0 \leftarrow f(\mathbf{x}_{best}^0)$
4: **while terminate**() $= false$ **do**
5: $\quad G_t \leftarrow \mathbf{mutate}_{\lambda\to\lambda}(\mathbf{recombine}_{\mu\to\lambda}(P_t))$
6: $\quad O_t \leftarrow \mathbf{evaluate}(G_t)$
7: $\quad \mathbf{x}_{best}^{t+1} \leftarrow \arg\min(\{f(\mathbf{x}) | \mathbf{x} \in O_t\} \cup \{f(\mathbf{x}_{best}^t)\}); f_{best}^{t+1} \leftarrow f(\mathbf{x}_{best}^{t+1})$
8: $\quad P_t^{sel} \leftarrow \mathbf{replace}_{\lambda\to\mu}^{\kappa}(P_t \cup O_t)$
9: $\quad P_{t+1} = \mathbf{increase\_age}(P_t^{sel})$
10: $\quad t \leftarrow t + 1$
11: **end while**
12: **return** $\mathbf{x}_{best}^t, f_{best}^t$

The EA starts with the initialization of the first population P_0. Let Ω_ω describe a probability space. Then the initialization is done by an initialization operator $\mathbf{init}_\mu : \Omega_\omega \to \mathbb{I}^\mu$ which usually initializes the object variables uniformly distributed within the search space. The generation of λ new individuals in O_t from individuals in P_t (summarized as $\mathbf{generate}_{\mu\to\lambda} : \mathbb{I}^\mu \to \mathbb{I}^\lambda$) can be subdivided into the recombination and the mutation phase: In the first phase, the recombination operator $\mathbf{recombine}'_{\mu\to\lambda} : \Omega_\omega \times \mathbb{I}^\mu \to \mathbb{I}^\lambda$ generates λ individuals by iterative calling of a reduced recombination operator $\mathbf{recombine}_{\rho\to 1} : \Omega_\omega \times \mathbb{I}^\rho \to \mathbb{I}$ that mixes the information of ρ individuals from P_t in order to generate a new individual. In the second phase, individuals of the resulting population of λ individuals are further modified by application of the mutation operator. The mutation operator $\mathbf{mutate}'_{\lambda\to\lambda} : \Omega_\omega \times \mathbb{I}^\lambda \to \mathbb{I}^\lambda$ iteratively applies the reduced mutation operator $\mathbf{mutate}_{1\to 1} : \Omega_\omega \times \mathbb{I} \to \mathbb{I}$, that randomly modifies (mutates) the search point. The intensity and random distribution of the variation is often controlled by the individual's strategy parameters. For some EA variants also these strategy parameters are mutated. After the generation of λ variations from the parent population, all new individuals are evaluated. Then a new parent population is composed by the deterministic operator $\mathbf{replace}^\kappa_{\mu+\lambda\to\mu} : \mathbb{I}^{\lambda+\mu} \times \mathbb{N} \to \mathbb{I}^\mu$. This operator favors individuals that have a good function value and it might also takes into account the age of the individuals, e. g. by comparing it to a maximal life span κ. Finally, the operator $\mathbf{increase\_age}_{\mu\to\mu} : \mathbb{I}^\mu \to \mathbb{I}^\mu$ is applied, that does nothing else but to increase the age of all individuals in the just now composed population P_{t+1} by one. The generational loop is then repeated with this new parent population until some termination criterion is fulfilled. The termination criterion might be related to a measure of diversity in the population or simply the exceed of a maximum number of evaluations.

Next, we are going to specify a common instantiation of ES for continuous search spaces with box constraints:

Representation

In continuous optimization with ES the space of individuals is described as

$$\mathbb{I} = \mathbb{R}^d \times (\mathbb{R}^+)^{n_s} \times (\mathbb{R} \cup \oslash) \times \mathbb{N} \tag{3.5.21}$$

For an individual

$$\mathbf{a} = (\mathbf{x}, \mathbf{s}, y, k) \in \mathbb{I} \tag{3.5.22}$$

the continuous vector $\mathbf{x} = (x_1, \ldots, x_d) \in \mathbb{R}^d$ denotes the vector of d object variables and $\mathbf{s} = (s_1, \ldots, s_{n_s})$ denotes the vector of n_s strategy parameters, that determine the shape of the random distribution that is employed for generating mutations of the object variables. The strategy parameters play a decisive role in controlling the granularity of the search process during an optimization run.

Furthermore, a single cost value is denoted with y and the individual's age (in generations) is denoted with t. Since we deal with minimization problems, the fitness value will be set equal to $y = f(\mathbf{x})$ after the evaluation of an individual.

Initialization

Given a box constrained search space with upper bounds $\mathbf{x}_{\min} \in \mathbb{R}^d$ and upper bounds $\mathbf{x}_{max} \in \mathbb{R}^d$, the initialization operator $\mathbf{init}_\mu : \Omega_\omega \to \mathbb{I}^\mu$ may sample μ individuals, which are uniformly distributed within the bounds, i. e.:

$$x_j^{(i)} = x_{min,j} + \mathrm{U}(0,1) \cdot [x_{max,j} - x_{min,j}], i = 1, \ldots, \mu, j = 1, \ldots, d. \tag{3.5.23}$$

This type of initialization will be called *uniform initialization*.

Another possibility, that can also be used in unconstrained optimization, is, to initialize the start population by a gaussian distributed sample around a start point $\mathbf{x}^{init}$:

$$x_j^{(i)} = x_j^{init} + s_j^{(i)} \cdot \mathrm{N}(0,1), i = 1, \ldots, \mu, j = 1, \ldots, d. \tag{3.5.24}$$

This kind of initialization will be referred to as *Monte Carlo initialization*. It can also be applied for unconstrained optimization.

The strategy parameters in $\mathbf{s}$ are typically initialized by the user. If interval constraints are provided it is suggested [Bäc96] to choose

$$s_j^{(i)} = s_{min,j} + 0.06[s_{max,j} - s_{min,j}], j = 1, \ldots, \mu \tag{3.5.25}$$

as initial strategy parameters. This allows for a coarse grained search in the beginning of the optimization.

Recombination

The recombination $\mathbf{recombine}_{\mu \to \lambda} : \Omega_\omega \times \mathbb{I}^\mu \to \mathbb{I}^\lambda$ is carried out by generating λ individuals by λ times generating a single individual by means of the reduced recombination operator $\mathbf{recombine}'_{\mu \to 1} : \Omega_\omega \times \mathbb{I}^\mu \to \mathbb{I}$. Before stating the recombination operator, two vector operations shall be introduced:

For a set of continuous vectors $\mathbf{u}_1 \in \mathbb{R}^d, \ldots, \mathbf{u}_\mu \in \mathbb{R}^d$ the intermediate recombination function $r_I : (\mathbb{R}^d)^\rho$ will be defined as

$$r_I(\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(\rho)}) = (\frac{1}{\rho} \sum_{i=1}^{\rho} u_1^{(i)}, \ldots, \frac{1}{\rho} \sum_{i=1}^{\rho} u_d^{(i)}) \tag{3.5.26}$$

Accordingly, discrete recombination (or dominant recombination) $r_D : \Omega_\omega \times (\mathbb{R}^d)^\rho \to \mathbb{R}^d$ will be defined as

$$r_D(\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(\rho)}) = ((u_1^{(1)} \text{ or } \ldots \text{ or } u_1^{(\rho)}), \ldots, (u_d^{(1)} \text{ or } \ldots \text{ or } u_d^{(\rho)})), \tag{3.5.27}$$

where the operator ' or ' symbolizes that each entry in the list gets chosen with the same probability. Next, with algorithm 3, we present an outline of the recombination operator that is used throughout this thesis.

Algorithm 3 Reduced ES recombination operator.

1: **input:** $P \in \mathbb{I}^{\mu}$
2: **Choose** ρ individuals $\{(\mathbf{x}_1, \mathbf{s}_1, y_1, k_1), \ldots, (\mathbf{x}_\mu, \mathbf{s}_\mu, y_\mu, k_\mu)\}$ randomly out of P
3: $\mathbf{x}' = r_D(\mathbf{x}_1, \ldots, \mathbf{x}_\rho)$
4: $\mathbf{s}' = r_I(\mathbf{s}_1, \ldots, \mathbf{s}_\rho)$
5: **return** $(\mathbf{x}', \mathbf{s}', \oslash, 0)$

Within the ES, recombination is applied for the object variables as well as for the strategy parameters. Typically discrete recombination is recommended for the first and intermediate recombination is recommended for the latter [Kur99]. However, Beyer [Bey01] frequently uses a different setting with $\mu = \rho$ and intermediate recombination for the object variables, for which it is easier to analyze the convergence dynamics. Another important version of the recombination operator discussed in literature is the *global recombination* [Sch95], where for each vector position the ρ individuals are chosen anew. Global recombination allows for an elegant way for the online adaptation of individual step-sizes, which works with a single mutation rate ([Sch95], page 147).

Mutation

In an ES, all individuals that result from the recombination operator are randomly modified by means of the reduced mutation operator $\mathbf{mutate}'_{1\to1}$. Formally the global mutation operator $\mathbf{mutate}_{\lambda\to\lambda} : \mathbb{I}^{\lambda} \to \mathbb{I}^{\lambda}$ reads:

$$\mathbf{mutate}_{\lambda\to\lambda}(\mathbf{a}^{(1)}, \ldots, \mathbf{a}^{(\lambda)}) = \underbrace{\{\mathbf{mutate}'_{1\to1}(\mathbf{a}^{(1)}), \ldots, \mathbf{mutate}'_{1\to1}(\mathbf{a}^{(\lambda)})\}}_{\lambda \text{ times}} \tag{3.5.28}$$

There have been different suggestions for the choice of the reduced mutation operator. Within the canonical ES a gaussian distributed value is added to the object variables and the strategy parameters are multiplied with a log-normal distributed value.

Let $\mathbf{a} = (\mathbf{x}, \mathbf{s}, y, k)$ denote an arbitrary individual that has to be mutated. Then the following procedure is applied in order to produce the result of the mutation $\mathbf{a}' = \mathbf{mutate}'_{1\to1}(\mathbf{a})$.

Algorithm 4 Reduced ES mutation operator.

1: **input:** $\mathbf{a} = (\mathbf{x}, \mathbf{s}, y, k) \in \mathbb{I} = \mathbb{R}^d \times (\mathbb{R}^+)^{n_s} \times (\mathbb{R} \cup \oslash) \times \mathbb{N}$
2: $N_{global} \leftarrow N(0,1)$
3: **for all** $i \in \{1, \ldots, d\}$ **do**
4: $\quad s'_i \leftarrow s_i \cdot \exp(\tau_{local} \cdot N(0,1) + \tau_{global} \cdot N_{global})$
5: $\quad x'_i \leftarrow x_i + s'_i \cdot N(0,1)$
6: **end for**
7: **return** $\mathbf{a}' = (\mathbf{x}', \mathbf{s}', y, k)$

The values of τ_{local} and τ_{global} are called local and global learning rate. In [Sch95] and [BS02] it has been proposed to choose $\tau_{local} = (\sqrt{2 \cdot \sqrt{d}})^{-1}$ and $\tau_{global} = (\sqrt{2d})^{-1}$ as default values for the learning rates. However, Kursawe [Kur99] found that the optimal choice of these parameter very much depends on the problem at hand. A simple but effective method to mutate a single step-size is the two-point operator as proposed in [Rec94] and later analyzed by Beyer (cf. [Bey01] p. 325). More details on this operator will be provided in 4.5.2.

Replacement

The replacement operator $\mathbf{replace}^{\kappa}_{\mu\to\lambda} : \mathbb{I}^{\mu+\lambda} \to \mathbb{I}^{\mu}$ in ES is deterministic and can be easily described. Let $\mathbf{a}_{1:\lambda+\mu}, \ldots, \mathbf{a}_{\lambda+\mu:\lambda+\mu}$ denote the in increasing order due to the comparison operator $<_\kappa: \mathbb{I} \times \mathbb{I} \to \{true, false\}$ with

$$\begin{aligned} \mathbf{a} <_\kappa \mathbf{a}' \text{ iff } \quad & k < \kappa \wedge k' < \kappa \wedge f(\mathbf{x}) < f(\mathbf{x}') \\ \vee \quad & k \geq \kappa \wedge k' \geq k \\ \vee \quad & k < \kappa \wedge k' \geq \kappa \end{aligned} \tag{3.5.29}$$

Then replace is defined as follows:

$$\mathbf{replace}^{\kappa}_{\lambda+\mu\to\mu}(\{\mathbf{a}_{1:\mu+\lambda}, \ldots, \mathbf{a}_{\mu+\lambda:\mu+\lambda}\}) = \{\mathbf{a}_{1:\mu+\lambda}, \ldots, \mathbf{a}_{\mu:\mu+\lambda}\} \tag{3.5.30}$$

In case of $\kappa = 1$ the resulting strategy is called (μ, λ)-ES and in case of $\kappa = \infty$ it is called $(\mu + \lambda)$-ES. Although, the setting $\kappa = 1$ has the advantage that it is easy to escape local optima and that it allows for a better functioning of the step size adaptation, experience shows that for limited numbers of evaluations higher values for κ should be favored.

3.6 Existing work on metamodel-assisted evolutionary optimization

Recently, there have been some first efforts to introduce approximate function also in evolutionary optimization. A good summary of them can be found in Jin et al. [Jin05]. Jin terms approaches that use fitness approximations to accelerate evolutionary algorithms *controlled evolution*. There are two main approaches for model-assistance applied within the context of controlled evolution. The first approach, that we will also adopt in this thesis, is called *individual based* approach. The basic principle of this approach is to pre-screen the generated individuals by means of approximate evaluations and then to evaluate the most promising individuals precisely and consider them for the next generation. The *generation based* approach works in a different way. Here, for some generations all individuals are evaluated entirely on the metamodel. Then a control step takes place and one generation is evaluated precisely. The individual based control [GGK01, GEN+01, Rat98] is similar to the model assisted pattern search algorithm that also uses the metamodel for pre-evaluation in each iteration (section 3.3), whereas the

generation based pre-screening [EBNK99, ND03] is similar to the bayesian global optimization approach (section 3.4). Typically, the metamodels are used as simple predictors, i. e. no use is made of the uncertainty information as it is provided with the variance of Kriging. An exception can be found in the paper of Keane et al. [EBNK99]. Here, Kriging predictors and their estimated variance have been employed in the context of a genetic algorithm. Some individuals in the proposed procedure are pre-selected by solely by means of the variance and some individuals are pre-selected due to the predicted value. The selection by means of the variance is mainly motivated by the goal to increase the model quality, regardless if the sampled region contains promising solution.

The first evolution strategies that make use of metamodels were introduced by Giotis et al. [GEN+01]. Here, the authors adopted a radial basis function network for the purpose of metamodel. A measure that combines the variance with the mean value of the prediction has been proposed for evolution strategies by Emmerich et al. [EGÖ+02] and later adopted by other authors [USZ03, BSK05], using a different variant of the evolution strategy working with a derandomized covariance matrix adaptation [HO01] that works with comparably small population sizes. Within the context of evolution strategies, Emmerich et al. [EGÖ+02] proposed to use the Kriging method for metamodeling. Note, that in this thesis the Kriging method is referred to as gaussian random field metamodel, in order to emphasize on the gaussian distribution that describes the predictions. Before more details on the relationship between different proposed metamodel-assisted evolution strategies are discussed, it seems suitable to introduce the general framework of metamodel-assisted evolution strategies, as it is done in the next chapter.

4 Metamodel-assisted evolution strategies

Whosoever wishes to know about the world must learn about it in its particular details.

Herakletos of Ephesos.

In this chapter we propose evolution strategies assisted by metamodels. Unlike in the previous work on metamodel-assisted strategies, emphasis is given to the treatment of the variance. It will be demonstrated that the use of the variance information can be of crucial importance in order to prevent premature stagnation of the search. The chapter includes a thorough description of the metamodel-assisted ES (MAES) framework and a detailed discussion of its alternative implementations.

Firstly, in section 4.1 we introduce the algorithmic framework of the MAES. Then, in section 4.2, we propose and discuss solution filters for the MAES, the choice and design of which is a crucial part in the development of the MAES. In section 4.3 we proceed with a brief discussion of theoretical results on the algorithm, including a proof of convergence to a global optimum. Accuracy measures, needed for the empirical analysis of the MAES, are proposed in section 4.4. Based on these measures studies of the algorithmic behavior are provided in section 4.5 for various artificial landscapes. The chapter ends with a concluding discussion on the use of uncertainty measures in metamodel-assisted evolution strategies (section 4.6).

4.1 Algorithmic framework

There are many possibilities of integrating metamodels into evolutionary algorithms. For example, one could use generation control or individual control (cf. section 3.6). Generation control faces the difficulty that it enforces to work with approximations, even if it is known that the information is insufficient to guide the search. Especially when dealing with high dimensional problems, the global quality of the metamodel can be very poor, and in that case a steady update of the database would be of crucial importance. This is why we favor an individual-based control scheme. In particular, for each candidate solution it shall be decided, whether it is evaluated precisely or rejected, depending on its expected value and the confidence value used in the approximate evaluation.

In this work, the screening of new individuals will be done by means of *imprecise evaluation filters* (briefly: IPE-filters or just *filters*). IPE filters are defined as operators on sets of solutions that reduce a set of imprecisely evaluated candidate solutions G to a subset

of promising solutions $Q \subseteq G$. Only those candidate solutions that pass the filter are processed further within the algorithm, while all other solutions are rejected.

A straightforward way to install filters in the EA is to employ them as a pre-selection operator before the precise evaluation and replacement procedures, in order to reduce the number of solutions that are precisely evaluated. Alternatively, filters might also be installed in other parts of the algorithm. For example, they may assist the variation operators to produce a high ratio of promising solutions. However, in this work we will focus on the use of filters as a pre-selection operator in an ES. The concepts derived, can easily be transferred to other optimization algorithms that work with a selection-variation scheme, like other types of EA and also many algorithms that can be assigned to the category of generalized pattern search (cf. section 3.3).

The MAES incorporates a filter before the evaluation of the offspring population. According to Jin's classification scheme [Jin05], it can be classified as an EA with individual based control (cf. section 3.6). Algorithm 5 displays an outline of the main loop of an MAES, which is a modified version of the basic $(\mu + \lambda)$-ES described in algorithm 2.

Two features distinguish the MAES from the standard ES. Firstly, any exactly evaluated individual is recorded in a database (lines 4 and 14 of Algorithm 5). Secondly, a pre-screening step is added and new candidate solutions are imprecisely evaluated using GRFM before deciding whether they should be re-evaluated by the exact evaluation software. By means of the filter, at time t, the set of offspring solutions G_t is reduced to the set of offspring solutions Q_t that which will be evaluated and considered in the final selection procedure.

The suggested filters for the MAES are parameterized operators. Typical parameters are the confidence factor ω and the number of pre-selected individuals ν. This is the case for fixed cardinality filters that let always pass the same number of individuals. Furthermore, some of these filters compare the approximated solution candidates to individuals in the parent population.

Positioning the filter before the replacement has the advantage that the algorithm remains transparent and thus the effect of the filter can easily be measured, as we will see in the subsequent sections. For filters with fixed cardinality the number of precise evaluations can be calculated from the number of generations, and vice versa. This is not the case for filters, for which the cardinality of the output set is determined by the quality of the input set. In that case, it is sometimes difficult to decide whether the performance of the algorithm is due to the quality of the filter or due to the number of generations. Hence, we will study both: Filters with fixed output size and filters with variable output size. In any case, the number of solutions that enter a subsequent generation will be denoted with ν_t.

In case of fixed cardinality filters we shall denote the output size with ν and term the resulting algorithm a $(\mu, \kappa, \nu < \lambda)$-*MAES*.

Before discussing different types of filters, the online learning of the metamodel will be addressed briefly. In line 8 of algorithm 5, all offspring individuals are evaluated by means of the GRFM. For each offspring solution an approximation is calculated by means of a metamodel. In particular, we are interested in the prediction of the mean value $\hat{y}(\mathbf{x})$ and the standard deviation $\hat{s}(\mathbf{x})$. The training of a metamodel from the entire database

Algorithm 5 $(\mu + \nu < \lambda)$**-MAES.**

$t \leftarrow 0$
$P_0 = \textbf{evaluate}(\textbf{init}_\mu())$ {Initialize population}
$\mathbf{x}^0_{best} \leftarrow \arg\min(\{f(\mathbf{x})|\mathbf{x} \in P_0\}); f^0_{best} \leftarrow f(\mathbf{x}^0_{best})$
$D_0 \leftarrow P_0$
while terminate$() = false$ **do**
  $G_t \leftarrow \textbf{mutate}_{\lambda\rightarrow\lambda}(\textbf{recombine}_{\mu\rightarrow\lambda}(P_t))$
  /* Pre-screening phase - start */
  evaluate G_t approximately with metamodel derived from D_t
  $Q_t \leftarrow \textbf{filter}_{\omega,\nu}(G_t, P_t)$ {Select subset Q_t of size ν from G_t by means of an IPE filter with parameter ω}
  /* Pre-screening phase - end */
  $O_t \leftarrow \textbf{evaluate}(Q_t)$
  $\mathbf{x}^{t+1}_{best} \leftarrow \arg\min(\{f(\mathbf{x})|\mathbf{x} \in O_t\} \cup \{f(\mathbf{x}^t_{best})\}); f^{t+1}_{best} \leftarrow f(\mathbf{x}^{t+1}_{best})$
  $P^{sel}_t \leftarrow \textbf{replace}^{\kappa}_{\nu+\mu\rightarrow\mu}(P_t \cup O_t)$
  $D_{t+1} \leftarrow O_t \cup D_t$
  $P_{t+1} = \textbf{increase\_age}(P^{sel}_t)$
  $t \leftarrow t + 1$
end while

of points can be very time consuming. Thus, local metamodels are used, as suggested in section 2.5. For each solution $\mathbf{x}$ in G_t, a metamodel is trained from its nearest local neighbors in the database. Leave-one-out cross validation [JSW98] is used in order to assure that all predictions are feasible. Furthermore, the inversions of the correlation matrix in the GRFM procedure are checked by multiplying the correlation matrix with its inverse and comparing it to the unity matrix[1]. If significant numerical problems are encountered for the approximation, precise evaluations are enforced. In order to avoid numerical problems caused by inversion, a minimal distance between neighboring solutions is required.

Furthermore, it is noteworthy that the number of neighboring solutions used for the training has been set to $2d$, where d is the dimension of the search space. Further training points usually enhance the quality of the metamodel. On the other hand, extra points might increase the approximation time significantly. However, it has been found in practical experiments, which will be presented later, that the approximations obtained with the proposed local metamodeling technique are already sufficient to establish a good approximate order among the function values of the individuals in the offspring population.

4.2 Imprecise evaluation filters

The incorporation of a filter used as pre-selection operator before the precise evaluation of individuals is the main operator that distinguishes the MAES from the ES. Various interesting alternatives of how to design a filter can be considered. We will present and

[1] A maximal deviation of 10^{-12} for each entries of the product matrix from the unity matrix shall be accepted.

evaluate some of them in the subsequent chapters, including filters that have already been discussed in past publications as well as yet unpublished approaches.

Fixed cardinality filters let pass always the same constant number of solutions. These solutions are typically selected by means of a scalar criterion that is derived from the approximate evaluation. Filters of that kind are introduced in section 4.2.1. *Improvement-based filters* compare the approximations to a threshold value (e.g. the best found solution found so far). Four different improvement-based filters are discussed in section 4.2.2. Last but not least, the concept of *interval filters* is introduced in subsection 4.2.3. Unlike the filters presented before, these filters decide whether an individual is selected or not by taking into account the whole ensemble of candidate solutions presented to the filter. Interval filters handle approximations as confidence intervals with lower and upper confidence bounds and are based on the theory of interval orders.

4.2.1 Mean value and lower confidence bound filters

The simplest strategy for filtering out less promising solutions is to select a constant number of $\nu < \lambda$ individuals due to the predicted mean value $\hat{y}(\mathbf{x})$ with the metamodel. This strategy will be termed *mean value filter* or briefly $\hat{y}$-filter. This technique does not make use of the confidence information and thus can be used with metamodeling techniques that do not provide this piece of information. This is probably the reason why it is most frequently used in literature (cf. [Jin05]).

Emmerich et al. [EGÖ+02] suggested to incorporate the confidence information for filtering candidate solutions in the evolution strategy. This has been done using the lb_ω criterion that has been suggested as a optimization criterion in bayesian Global Optimization (section 3.4). In particular, they use a so-called *lower confidence bound (lb_ω) filter* that establishes a ranking due to the lb_ω criterion (equation 3.4.17) and select the ν best solutions according to this ranking. They provide first empirical evidence that an MAES using the lb_ω filter is more robust against premature stagnation than an MAES working with a mean value filter. This has been ascertained, empirically, on multimodal problems.

Example: As an example, figures 4.2.1 and 4.2.2 visualize the $\hat{y}$ filter and the lb_ω filter, respectively. In both figures the approximately evaluated offspring population G_t consists of the five individuals `xo1` ... `xo5`, and the parent population P_t consists of the three individuals `xp1`, `xp2`, and `xp3`. Unlike for the filters that will be introduced later in this work, the parent population is not considered by the mean value and the $\mathbf{l}$ filter. For the mean value filter and the lb_ω filter only the five individuals of the offspring population G_t are considered. If ν is set to three, the mean value filter accepts the individuals `xo1`, `xo2`, and `xo3`, that have the three smallest mean values. The solutions `xo4` and `xo5` are rejected. The lb_ω filter accepts `xo1`, `xo2`, and `xo5`, which have the three lowest values for the lower confidence bound.

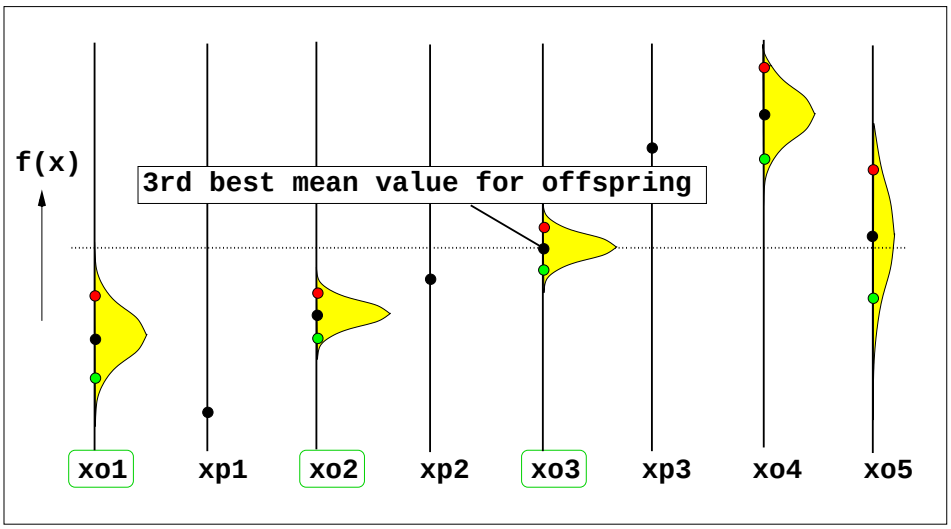


Figure 4.2.1: Mean value filter with a number of pre-selected individuals of $\nu = 3$. The precisely evaluated parent population is given by $P_t = \{\texttt{xp1}, \texttt{xp2}, \texttt{xp3}\}$ and the offspring population by $G_t = \{\texttt{xo1}, \ldots, \texttt{xo3}\}$. Only the three offspring solutions with the lowest mean value pass the filter. These are `xo1`, `xo2` and `xo5`. Note that the values of the parent individuals are not considered by the mean value filter. They have been included in the drawing to simplify comparisons to other filters.

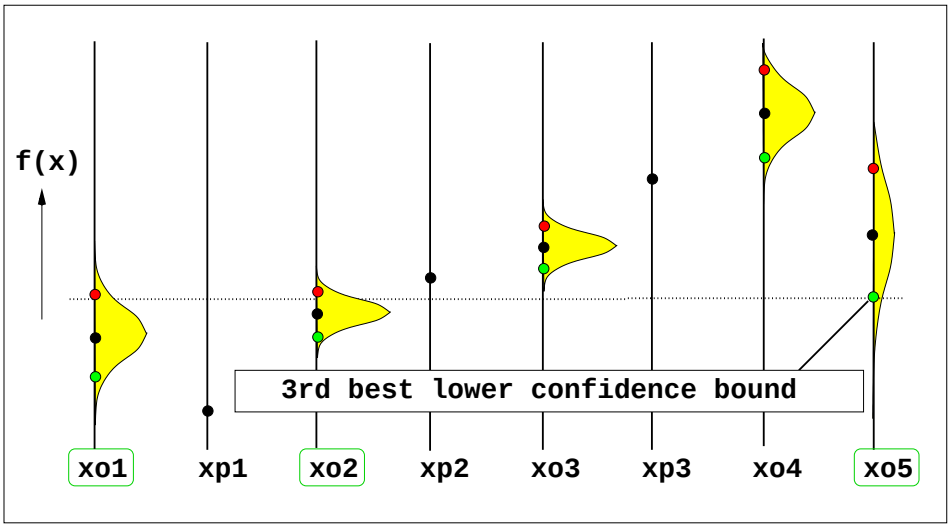


Figure 4.2.2: Lower confidence bound filter with $\nu = 3$. The precisely evaluated parent population is given by $P_t = \{\texttt{xp1}, \texttt{xp2}, \texttt{xp3}\}$ and the offspring population by $G_t = \{\texttt{xo1}, \ldots, \texttt{xo5}\}$. Only the three offspring solutions with the lowest lower confidence bounds pass the filter. These are `xo1`, `xo2` and `xo5`.

4.2.2 Improvement-based filters

The $\hat{y}$ and lb_ω filters previously mentioned are independent of the set P_t, to which the new candidate solutions in G_t will be compared for the replacement.

Improvement-based filters diminish this shortcoming by comparing new individuals in G_t to a reference value (threshold). As a default this reference value is chosen as the worst function value for an individual in the parent population P_t.

Let $\mathbf{x}^t_{best}$ denote such a reference solution and $y^t_{ref} = y(\mathbf{x}^t_{ref})$. Then, according to expression 3.4.18 $\text{I}(y(\mathbf{x})) = \max\{f^t_{best} - y(\mathbf{x}), 0\}$ defines the improvement for any new set of design variables $\mathbf{x} \in \mathbb{S}$. Next, we will define four criteria that are based on the notion of improvement and compare them on a conceptual level.

Probability of improvement. According to Ulmer et al. [USZ03] the *probability of improvement* is defined as

$$\text{PoI}(\mathbf{x}) := \Pr(\text{I}(y(\mathbf{x})) > 0) \tag{4.2.1}$$

in case of $\hat{s}(\mathbf{x}) > 0$. It will turn out to be convenient to define the probability of improvement also for precise evaluations, i. e. for evaluations with $\hat{s}(\mathbf{x}) = 0$. In that case we define $\text{PoI}(\mathbf{x}) = 1$, if $\text{I}(\hat{y}(\mathbf{x})) > 0$, and $\text{PoI}(\mathbf{x}) = 0$, if $\text{I}(\hat{y}(\mathbf{x})) = 0$. The term in expression 4.2.1 can be calculated via

$$\text{PoI}(\mathbf{x}) = \int_{-\infty}^{f^t_{best}} \text{PDF}_{\mathbf{x}}(y)dy = \Phi(\frac{f^t_{best} - \hat{y}(\mathbf{x})}{\hat{s}(\mathbf{x})}). \tag{4.2.2}$$

The probability of improvement has been studied as a ranking criterion in the ES by Ulmer et al. [USZ03]. Like the lower confidence bound criterion, it depends on the confidence measure $\hat{s}(\mathbf{x})$ and on the predicted value $\hat{y}(\mathbf{x})$. Ulmer et al. [USZ03] claim that an advantage to the lb_ω criterion is that the PoI does not depend on a user-specified parameter like the parameter ω for the lb_ω criterion. Another important difference between the PoI and the lb_ω criterion is that the PoI criterion does not take into account the quantity of an improvement. Thus its numerical value is invariant to monotone transformations of the GRFM. A potential problem with the PoI is that it tends to favor small improvements around existing solutions and thus the MAES avoids more risky steps that might lead to large improvements. This entails the danger of premature stagnation of the search.

Another, more subtle, difference of the PoI in comparison to the lb_ω is, that a higher variance does not lead necessarily to a equal or better ranking among other solutions, as it is the case for the lower confidence bound criterion. In fact, it depends on the value of $\hat{y}$ relative to f^t_{best} if an increased value of $\hat{s}$ leads to a higher value of the PoI. It can be easily obtained from a graphical visualization, that an increased variance leads to a decreased value of the PoI, iff $\hat{y} < f^t_{best}$, and to an increased value of the PoI, iff $\hat{y} > f^t_{best}$ (cf. figure 4.2.3).

Within the MAES there are two options how to employ the PoI criterion as a filter. One option is to sort the population by means of the PoI and only select the ν candidate solutions from G_t with the highest values for the PoI criterion. For this fixed cardinality filter, the user has to provide the number of pre-selected individuals ν. Another possibility is to keep the size of the output set variable and to define a probability threshold. Only

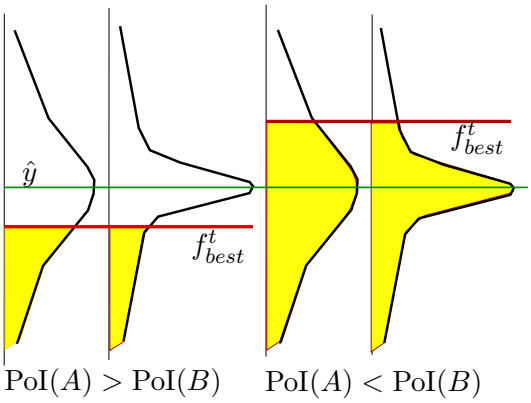


Figure 4.2.3: Schematic draw illustrating the influence of the variance $\hat{s}$ on the PoI. The filled area depicts the area, the size of which determines the PoI. If $\hat{y} > f^t_{best}$ (left figure) an increased variance corresponds to an increased PoI. Otherwise, if $\hat{y} < f^t_{best}$ (right figure) an increased variance corresponds to a decreased PoI.

candidate solutions with a value below this threshold will be rejected. The latter filter will be termed PoI$_\tau$ filter. Here the τ parameter denotes the threshold probability.

Expected Improvement. Another filter that is based on an integral expression is the *expected improvement filter* (ExI filter), which is based on the expected improvement (ExI) criterion (cf. section 3.4). In contrast to the PoI criterion this criterion takes into account the expected quantity of an improvement.

Slightly different from equation 3.4.19 we will define the ExI criterion as follows:

For $\hat{s}(\mathbf{x}) > 0$

$$\mathrm{ExI}(\mathbf{x}) = \mathrm{E}(\mathrm{I}(\mathbf{x})) = \int_{-\infty}^{f^t_{best}} \mathrm{I}(y) \cdot \mathrm{PDF}_{\mathbf{x}}(y) dy = \int_{-\infty}^{f^t_{best}} \mathrm{I}(y) \cdot \varphi(\frac{f^t_{best} - \hat{y}(\mathbf{x})}{\hat{s}(\mathbf{x})}) dy \qquad (4.2.3)$$

Again φ denotes the probability density function of the standard gaussian distribution. For $\hat{s}(\mathbf{x}) = 0$ we will deviate from the definition by Schonlau et al. [JSW98] and define the expected improvement as equal to $\mathrm{I}(\hat{y}(\mathbf{x}))$. This makes sense, because, if we know the improvement precisely, it will be not a good assumption to set it to zero, like they do in [JSW98]. In the context of pre-selection, it turns out that it is better to set its value exactly the value of the improvement we will get. Recall, that the integral in equation 4.2.3 can directly be calculated by means of the closed analytical expression given in equation 3.4.19.

The expected improvement always takes positive values, provided that $\hat{s}(\mathbf{x}) > 0$. Only in case of $\hat{s}(\mathbf{x}) = 0$ and $\hat{y}(\mathbf{x}) > y^t_{ref}$ the expected improvement is equal to zero. Furthermore, the expected improvement is a monotonously decreasing function in $\hat{y}(\mathbf{x})$. Moreover we note that $\mathrm{ExI}(\mathbf{x})$ has a limit value of zero as its argument approaches infinity. Like the PoI the ExI integrates the confidence information and does not require a user specified parameter, like the ω-parameter for the lb$_\omega$ criterion. Furthermore it is usually more optimistic than the mean value.

In order to integrate the ExI criterion as a filter into the MAES, there are two options: One option is to only accept solutions that have ExI values above a certain threshold and the other option is to work with a constant number of ν pre-selected individuals that are the individuals with the ν highest values for the ExI. The latter option has clearly to be preferred to the first option, since when working with the first criterion, it is by no means clear how the threshold value should be controlled during the search.

Most likely improvement. Next, we propose a straightforward design of a threshold filter based on the mean value criterion and relate it to the previously defined improvement based filters.

Taking a statistical stance, under the GRFM assumptions the value $\hat{y}(\mathbf{x})$ is not only the mean value, but also the most likely outcome of the computer experiment for an input vector $\mathbf{x}$. Accordingly, the most likely improvement would be defined as:

$$\mathrm{MLI}(\mathbf{x}) = \mathrm{I}(\hat{y}(\mathbf{x})) = \begin{cases} \hat{y}(\mathbf{x}) - f^t_{best} & \text{if } \hat{y}(\mathbf{x}) > y^t_{\min} \\ 0 & \text{otherwise} \end{cases} \tag{4.2.4}$$

Like the $\hat{y}$ criterion, the MLI criterion does not take into account the uncertainty of the approximation, and thus it can be used with metamodels that do not provide a confidence measure. Like the ExI criterion it also measures the quantity of an improvement. In order to establish the relationship between the MLI and the ExI, we can prove the following lemma:

Lemma 1. For any $\mathbf{x} \in \mathbb{S}$, we find that $\mathrm{MLI}(\mathbf{x}) \leq \mathrm{ExI}(\mathbf{x})$, and in case of $\hat{s}(\mathbf{x}) > 0$ we find that $\mathrm{MLI}(\mathbf{x}) < \mathrm{ExI}(\mathbf{x})$.

Proof. In case of $\hat{y}(\mathbf{x}) < f^t_{best}$ and $\hat{s}(\mathbf{x}) > 0$, the statement follows from the positivity of the $\mathrm{ExI}(\mathbf{x})$ and from the fact that – by definition – in the considered case $\mathrm{MLI}(\mathbf{x})$ equals zero. In case of $\hat{s}(\mathbf{x}) = 0$ we always get $\hat{y}(\mathbf{x}) = f^t_{best}$, and hence $\mathrm{MLI}(\mathbf{x}) = \mathrm{ExI}(\mathbf{x})$.

Let us now turn to the remaining case: $\hat{y}(\mathbf{x}) > f^t_{best}$ and $\hat{s}(\mathbf{x}) > 0$. By definition of the MLI we get $\mathrm{MLI}(\mathbf{x}) = f^t_{best} - \hat{y}(\mathbf{x})$. Since $\hat{y}(\mathbf{x})$ is the expected value of the normal distribution, we can rewrite the equation of the MLI in the form

$$\mathrm{MLI}(\mathbf{x}) = f^t_{best} - \int_{-\infty}^{\infty} y \cdot \mathrm{PDF}(y) dy. \tag{4.2.5}$$

Now, we can pull the f^t_{best} value inside the integral, by making use of the equivalence

$$f^t_{best} = \int_{-\infty}^{\infty} f^t_{best} \cdot \mathrm{PDF}(y) dy \tag{4.2.6}$$

. Hence,

$$\mathrm{MLI}(\mathbf{x}) = \int_{-\infty}^{\infty} (f^t_{best} - y) \cdot \mathrm{PDF}(y) dy. \tag{4.2.7}$$

This integral can be decomposed into two addends

$$\mathrm{MLI}(\mathbf{x}) = \underbrace{\int_{-\infty}^{f^t_{best}} (f^t_{best} - y) \cdot \mathrm{PDF}(y) dy}_{=\mathrm{ExI}(\mathbf{x})} + \underbrace{\int_{f^t_{best}}^{\infty} (f^t_{best} - y) \cdot \mathrm{PDF}(y) dy}_{<0}. \tag{4.2.8}$$

Here, the first term in the sum equals the ExI expression, and the second is an integral that takes a value clearly below zero due to our assumption $f^t_{best} < \hat{y}$. Hence, the resulting MLI is smaller than the value of the ExI for any $\mathbf{x} \in \mathbb{S}$. □

According to lemma 1, the MLI criterion can be regarded as more pessimistic than the expected improvement criterion. The ranking produced by the MLI criterion generally can differ from that produced by the ExI and PoI criterion, since it does not make use of the variance.

Unlike in the case of the ExI and PoI criteria, there is a straightforward strategy of how to filter individuals by means of the MLI criterion. This is, only to let pass those individuals that have a positive MLI. Such a filter shall be termed *MLI filter*. However, a minimum and maximum number of pre-selected individuals should be defined in order to avoid stagnation or, on the other hand, too many evaluations per generation.

Taking a closer look, the MLI criterion is equivalent to the PoI_τ criterion for $\tau = 0.5$, since the most likely value of the prediction is exactly below the threshold, if the probability of improvement is one half. However, this equivalence will not be found by generalizations of the MLI and PoI for multi-objective and constrained problems.

Potential improvement. As in case of the $\hat{y}$ criterion, we can also re-define the lb_ω criterion as an improvement-based criterion: The *potential improvement* LBI_ω then reads:

$$\text{LBI}_\omega(\mathbf{x}) = \text{I}(\text{lb}_\omega(\mathbf{x})) = \text{I}(\hat{y}(\mathbf{x}) - \omega \cdot \hat{s}(\mathbf{x})), \omega \in \mathbb{R}^+. \tag{4.2.9}$$

It follows from positive values of ω and $\hat{s}(\mathbf{x})$ that the potential improvement is always larger than the most likely improvement. With either $\omega = 0$ or $\hat{s}(\mathbf{x}) = 0$ both criteria are equivalent.

In contrast to the PoI, ExI and MLI criteria, the LBI_ω criterion asks for the parameter ω. This might be considered as a drawback, because it requires a decision by the user. On the other hand, it gives the user the possibility to scale between global search and local search, as an increased ω amplifies the influence of $\hat{s}(\mathbf{x})$ and thus favors solution candidates in less explored regions.

Also for the potential improvement criterion there is a straightforward strategy of how to use the LBI_ω criterion as a filter. We suggest, to let pass only those individuals that obtain a positive potential improvement. The filter operator – that we will term LBI_ω-filter – should also be equipped with a minimal and a maximal number of individuals that can pass the filter.

Example: In Figures 4.2.4 and 4.2.5 the operations performed by the MLI and LBI_ω-filters are illustrated. The highest objective function value among individuals of the parent population $P_t = \{\texttt{xp1}, \texttt{xp2}, \texttt{xp3}\}$ is set as a threshold. Now, the MLI-filter selects only those individuals from the set $G_t = \{\texttt{xo1}, \ldots, \texttt{xo5}\}$ that have a mean value below this threshold. These are the solutions `xo1`, `xo2`, `xo3` and `xo5`. The LBI_ω-filter only accepts individuals with a lower confidence bound below the threshold. In the example, these are all individuals in the example population G_t. Note that, unlike in the example of the mean value and lb_ω filters, the information of the parent population is considered in the MLI and LBI_ω-filters.

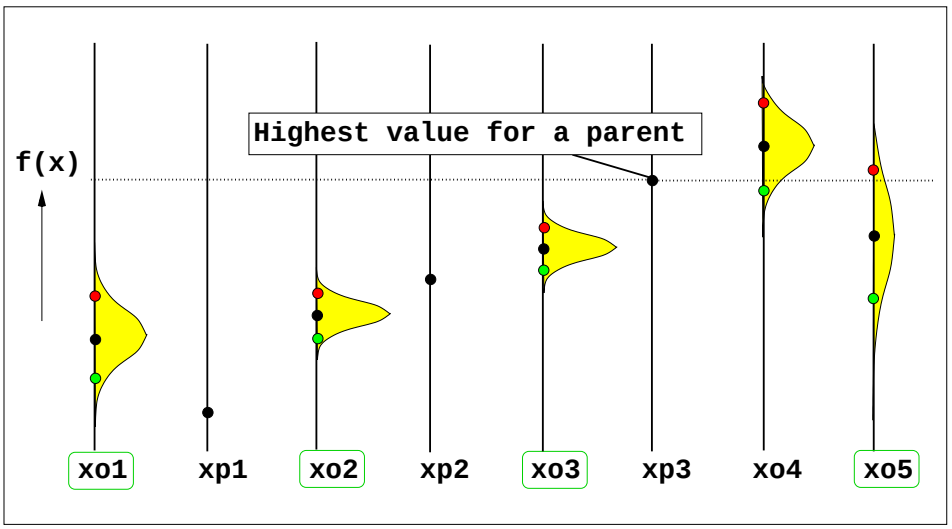


Figure 4.2.4: Illustration of the operation of an MLI-filter: The precisely evaluated parent population is given by $P_t = \{\texttt{xp1}, \texttt{xp2}, \texttt{xp3}\}$ and the offspring population by $G_t = \{\texttt{xo1}, \ldots, \texttt{xo5}\}$. The worst function value for a parent provides a threshold. Only the offspring solutions with mean value below this threshold pass the filter. These are `xo1`, `xo2`, `xo3` and `xo5`.

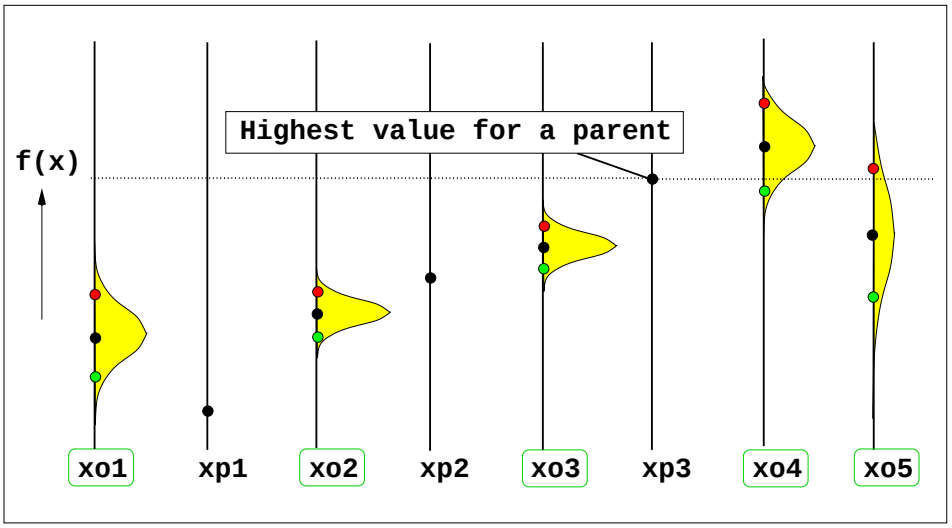


Figure 4.2.5: Illustration of the operation of an LBI_ω-filter: The precisely evaluated parent population is given by $P_t = \{\texttt{xp1}, \texttt{xp2}, \texttt{xp3}\}$ and the offspring population by $G_t = \{\texttt{xo1}, \ldots, \texttt{xo5}\}$. The worst function value for a parent provides a threshold. All offspring solutions have a lower confidence bound that is below this threshold and thus pass the filter.

4.2.3 Interval filters

Up to now, all filter procedures have been based on scalar criteria. In contrast to this interval filters, that will be proposed next, do not work with a scalar criterion but by comparing two-sided confidence intervals assigned to each candidate solution. In order to obtain confidence intervals for evaluations Emmerich et al. [EN04a] proposed to calculate for each solution a lower confidence bound $\mathrm{lb}_\omega(\mathbf{x})$ and an upper confidence bound $\mathrm{ub}_\omega(\mathbf{x})$ with

$$\mathrm{lb}_\omega(\mathbf{x}) = \hat{y}(\mathbf{x}) - \omega \cdot \hat{s}(\mathbf{x}),\ \mathrm{ub}_\omega(\mathbf{x}) = \hat{y}(\mathbf{x}) + \omega\hat{s}(\mathbf{x}). \tag{4.2.10}$$

Provided the GRFM assumptions are true, it can be said that the probability p_α that the true value is inside the specified interval reads

$$p_\alpha = \Pr(\mathcal{F}_\mathbf{x} \in [\mathrm{lb}_\omega(\mathbf{x}), \mathrm{ub}_\omega(\mathbf{x})]) = 1 - 2\Phi(-\omega). \tag{4.2.11}$$

This formula stems from the gaussian distribution assumption. The normalized gaussian random variable for $\mathcal{F}_\mathbf{x}$ reads $(\mathcal{F}_\mathbf{x} - \hat{y}(\mathbf{x}))/\hat{s}(\mathbf{x})$. The probability that the true value is below $\mathrm{lb}_\omega(\mathbf{x})$ is $\Phi(-\omega)$ and the probability that the true value is above ub_ω is $1 - \Phi(\omega)$, which is equal to $\Phi(-\omega)$ due to the symmetry of the gaussian distribution.

We make the convention that $\mathrm{lb}_\omega(\mathbf{x}) = \mathrm{ub}_\omega(\mathbf{x}) = \hat{y}(\mathbf{x})$, if $\hat{s}(\mathbf{x})$ is estimated as 0, which might happen on plateaus or if the result of $\mathbf{x}$ is precisely known. It is now possible to compare approximate and precise results with the same criterion.

Given these intervals, the question arises, which of the solutions of a set G_t of candidate solutions in algorithm 5 would be among the μ individuals selected in the replacement, if all function values would be evaluated precisely.

When determining this set, we can make two kinds of mistakes:

(A) We may select solutions, which are not among the set of the μ best solutions.

(B) We may reject solutions, which are among the set of μ best solutions.

A filter is said to work with a high *precision*, if it avoids mistake A. Accordingly, a filter is said to work with a high *recall*, if it avoids mistake B. The terms precision and recall stem from the theory of information retrieval ([vR79], pp. 112). There, the recall measures the percentage of relevant solutions that have been retrieved, and the precision measures percentage of relevant solutions among the retrieved solutions. Usually, in the presence of uncertainties there is a trade-off between these two objectives.

It is important to note, that in the following we will first develop concepts for the filters on basis of precise intervals bounding the objective function value (scenario 1). Then we apply them to two-sided confidence intervals that bound the precise objective function value only with a certain probability, depending on the confidence level (scenario 2). Hence, the statements derived for the scenario 1 are only valid for scenario 2 under the condition that the confidence intervals bound the realization of the random variables. This is true with a probability of at least $(1 - p_\alpha)^k$, if k denotes the number of approximations that we compare and p_α is the confidence level, which should be equally chosen for each confidence interval.

Example: Before starting to discuss the design of interval filters that take into account this trade-off, let us give a motivating example on how to reason with intervals that bound solutions. Let us consider we have a set of five solutions $\mathbf{x}_1, \ldots, \mathbf{x}_5$ and corresponding intervals $b(\mathbf{x}_1) = [-1, 1]$, $b(\mathbf{x}_2) = [2, 4]$, $b(\mathbf{x}_3) = [1, 2]$ and $b(\mathbf{x}_4) = [4, 10]$, $b(\mathbf{x}_5) = [5, 12]$ that bound the precise function values $f(\mathbf{x}_1), \ldots, f(\mathbf{x}_5)$ for these solutions. The question is, which of the solutions have to be selected, if we want to select for sure only solutions that are among the two minimal solutions. In the special case of solutions with equal function value, we assume that these solutions get sorted randomly.

Actually, the sole solution that can be selected is $\mathbf{x}_1$ with $b(\mathbf{x}_1) = [-1, 1]$, because only this one clearly dominates at least three other solutions. Likewise, we may ask what are the solutions that we need to select, if we want to be sure to have selected the two best solutions. This is the solution $\mathbf{x}_1$ with $b(\mathbf{x}_1) = [-1, 1]$, but also solutions $\mathbf{x}_2$ and $\mathbf{x}_3$ have to be considered, since they may not be dominated by more than two other solutions. □

Considerations like the one stated above gave rise to the development of filters that are highly adaptive and take into account the whole information about interval boundary values of the individuals presented to them, when deciding on whether to accept or reject one solution.

Next, we are going to derive two IPE filters that work on the basis of interval information and that operate on the two extremes of this trade-off. The filter that avoids mistake A will be called *P_ω-filter* since it maximizes the precision of the result. Accordingly the filter that avoids mistake B will be called *R_ω-filter* since it maximizes the recall of the result.

In order to estimate the subset of μ-best solutions from G_t with a good precision, the following lemma will prove to be useful:

Lemma 2. Let M denote a set of at least $\mu + 1$ solutions for that precise interval boundaries $\mathrm{lb}(\mathbf{x}) < \infty$ and $\mathrm{ub}(\mathbf{x}) < \infty$ for the function value $y(\mathbf{x})$ are known for all $\mathbf{x} \in M$. The intervals are considered to be closed, i. e. $y(\mathbf{x}) \in [\mathrm{lb}(\mathbf{x}), \mathrm{ub}(\mathbf{x})]$. Furthermore, let $\Upsilon_\mu(M) = \{\mathbf{x} \in M| \quad |\{\mathbf{x}' \in M | y(\mathbf{x}) < y(\mathbf{x}')\}| \geq |M| - \mu - 1\}$ denote a set of solutions that dominate at least $|M| - \mu - 1$ other solutions in M, and hence would be among the μ selected solutions, if only precise evaluations were used. Then

$$\zeta_\mu^A(M) := \{\mathbf{x} \in M| \quad |\{\mathbf{x}' \in M | \mathrm{ub}(\mathbf{x}) < \mathrm{lb}(\mathbf{x}')\}| \geq |M| - \mu\} \tag{4.2.12}$$

is the maximal subset of M that contains only elements that belong to the set Υ_μ.

Proof. Provided the intervals are valid, each element that dominates at least $|M| - \mu$ elements is for sure part of the μ best solutions. No other element can be included, because whenever this would be done it is possible that $\mathbf{x}$ does not belong to the μ best solutions. For example $\mathbf{x}$ falls out of the set $\Upsilon_\mu(M)$, if the true value of $y(\mathbf{x})$ is realized at the upper bound of the interval $[\mathrm{lb}(\mathbf{x}), \mathrm{ub}(\mathbf{x})]$ and for any other $\mathbf{x}' \in M$ the true value of $y(\mathbf{x}')$ is realized at the lower bound of the confidence interval $[\mathrm{lb}(\mathbf{x}'), \mathrm{ub}(\mathbf{x}')]$. In that case, μ solutions can be found in M that have an equal or smaller function value than $\mathbf{x}$ and thus it cannot be decided, whether $\mathbf{x}$ is selected or not. □

From lemma 2 we can easily derive a filter that can be used for the MAES. Let $P_t^{-\kappa} \subseteq P_t$ denote the set of individuals from P_t with an age lower than κ. Then, the set $Q_t =$

$\zeta_\mu^A(G_t \cup P_t^{-\kappa}) \cap G_t$ is determined by means of algorithm 6:

Algorithm 6 P_ω-filter (G_t, P_t): Reduces the set G_t to Q_t, aiming at a high precision.

```
Q_t ← ∅
for all x ∈ G_t do
  c ← 0
  for all x' ∈ P_t^{-κ} ∪ G_t do
    if ub_ω(x) < lb_ω(x') then
      c ← c + 1
    end if
  end for
  if c ≥ |P_t^{-κ} ∪ G_t| − μ then
    Q_t ← Q_t ∪ {x}
  end if
end for
return Q_t
```

It is possible – and also likely for large intervals and small populations – that the P_ω-filter selects no individual at all. Furthermore, the P_ω-filter never selects more than μ elements.

A similar filter algorithm can be found, if the aim is to maximize the recall, i.e. to avoid the rejection of individuals that might be selected. Again, we start with a lemma on solution sets with precise intervals for the evaluation of their members.

Lemma 3. Let M denote a set of solutions for the precise interval boundaries $\mathrm{lb}(\mathbf{x})$ and $\mathrm{ub}(\mathbf{x})$ for the true function value $y(\mathbf{x})$ of which are known for all $\mathbf{x} \in M$. Furthermore, let us define $\Psi_\mu(M) := \{\mathbf{x} \in M| \quad |\{\mathbf{x}' \in M|y(\mathbf{x}') < y(\mathbf{x})\}| < \mu\}$ as the set of individuals that are not strictly dominated by at least μ other individuals in M. Then the set

$$\zeta_\mu^B(M) := \{\mathbf{x} \in M| \quad |\{\mathbf{x}' \in M|\mathrm{ub}(\mathbf{x}') < \mathrm{lb}(\mathbf{x})\}| < \mu\} \tag{4.2.13}$$

contains the minimal subset of M that includes all elements from $\Psi_\mu(M)$. □

Proof. Provided the intervals are valid, each element that is dominated by less than μ elements possibly belongs to the μ smallest solutions. This can be verified by making the most optimistic assumption about the true value for $y(\mathbf{x})$ that is $y(\mathbf{x}) = \mathrm{ub}(\mathbf{x})$ and the most pessimistic assumption about all other individuals that is $y(\mathbf{x}') = \mathrm{lb}(\mathbf{x}')$ for all $\mathbf{x}' \in M \setminus \{\mathbf{x}\}$. In this case $\mathbf{x}$ would be among the solutions in Ψ_μ^B and – with regard to the $\mathbf{replace}_{\infty,|M|\to\mu}$ operator – it cannot be sure that the element is rejected. Furthermore, it is clear that no other element from M needs to be considered, since any element from M that is dominated by more than μ solutions is for sure not selected. □

Again, we can derive a filter algorithm from the lemma. This time lemma 3 provides the principle for the R_ω-filter algorithm, that aims at a high recall. It rejects only those individuals that would for sure be rejected by the replace operator, under the condition that all realizations of the random variables are enclosed by their confidence intervals. This is done by detecting the solutions $\zeta_\mu^B(P_t \cup G_t) \cap G_t$ by means of algorithm 7.

The computation time of algorithm 6 and algorithm 7 is given by $\mathcal{O}(((|G_t| + |P_t|) \cdot |G_t|)$. This quadratic time complexity is practically of no significant importance for the MAES, if

Algorithm 7 R_ω-filter (G_t, P_t): Reduces the set G_t to Q_t, aiming at a high recall.

```
Q_t ← ∅
for all x ∈ G_t do
  c ← 0
  for all x' ∈ (P_t^{-κ} ∪ G_t) − {x} do
    if ub_ω(x') < lb_ω(x) then
      c ← c + 1
    end if
  end for
  if c < μ then
    Q_t ← Q_t ∪ {x}
  end if
end for
return Q_t
```

we work with small sets of Q_t and P_t. However, if these sets get larger, a modified version of these algorithms should be used that reduces the calculation time to $\mathcal{O}(\log|G_t| \cdot |G_t|)$ for $|P_t| < |Q_t|$. This algorithm is based on the following lemma:

Lemma 4. Let M be a set of solutions which contains at least $\mu + 1$ elements. For all elements upper bounds $\mathrm{ub}(\mathbf{x})$ and lower bounds $\mathrm{lb}(\mathbf{x})$ for the true function value $y(\mathbf{x})$ are known. Furthermore, for $m = |M|$ and $\mu \leq m$ let us define the μ smallest lower bound as

$$\mathrm{lb}_{\mu:m}(M) = \min_{\mathbf{x}\in M}\{\mathrm{lb}(\mathbf{x})| \quad |\{\mathbf{x}' \in M|\mathrm{lb}(\mathbf{x}) \geq \mathrm{lb}(\mathbf{x}')\}| \geq \mu\}. \tag{4.2.14}$$

This is the smallest lower bound for which no more than μ elements are better or equal than $\mathrm{lb}(\mathbf{x})$. Likewise, let us define the μ lowest upper bound as:

$$\mathrm{ub}_{\mu:m}(M) = \min_{\mathbf{x}\in M}\{\mathrm{ub}(\mathbf{x})| \quad |\{\mathbf{x}' \in M|\mathrm{ub}(\mathbf{x}) \geq \mathrm{ub}(\mathbf{x}')\}| \geq \mu\} \tag{4.2.15}$$

Then

$$\zeta_\mu^A(M) = \{\mathbf{x} \in M|\mathrm{ub}(\mathbf{x}) < \mathrm{lb}_{\mu+1:|M|}(M)\} \tag{4.2.16}$$

and

$$\zeta_\mu^B(M) = \{\mathbf{x} \in M|\mathrm{lb}(\mathbf{x}) \leq \mathrm{ub}_{\mu:|M|}(M)\}. \tag{4.2.17}$$

Proof. Let us first prove that equation 4.2.16 is true. Consider a solution $\mathbf{x} \in M$ for which it should be decided, whether it belongs to $\Upsilon_\mu(M)$ or not. It is clear that all solutions that have a upper bound below the threshold $\mathrm{lb}_{\mu+1:|M|}(M)$ have a smaller true function value than $\mathrm{lb}_{\mu+1:|M|}(M)$ and thus they dominate at least one element from M. For $|M| = \mu + 1$, we are done. For $|M| = \mu + 2$ from the transitivity of the linear order it follows that there must be $|M| - \mu - 2$ further solutions in M that have a lower bound higher than $\mathrm{lb}_{\mu+1:|M|}(M)$ and thus are dominated by $\mathbf{x}$. Hence, $\mathbf{x}$ dominates more than $|M| - \mu - 2$ solutions as it is required for being member of $\Upsilon_\mu(M)$. Furthermore, it has to be proven that no further elements than the ones defined in expression 4.2.16 fulfill the requirements for being member of Υ_μ. For that purpose, consider an element $\mathbf{x}$ with $\mathrm{ub}(\mathbf{x}) \geq \mathrm{lb}_{\mu+1:|M|}(M)$. If we are pessimistic about the outcome of $\mathbf{x}$ by assuming that its function value is realized at its upper bound and optimistic about the function value

of all other $\mathbf{x} \in M$, then $\mathbf{x}$ dominates at most $\max\{0, |M| - \mu - 2\}$ solutions and thus cannot be part of $\Upsilon_\mu(\mathbf{x})$.

A similar argumentation can be provided in order to prove equation 4.2.17. Consider, a solution $\mathbf{x} \in M$ with $\mathrm{lb}(\mathbf{x}) \leq \mathrm{ub}_{\mu:|M|}(M)$. Then $\mathbf{x}$ might be selected for $\Psi_\mu(M)$ and thus is part of $\zeta_\mu^B(M)$, because the true function value for $\mathbf{x}$ is strictly dominated by less than μ function values for other solutions in M. Furthermore all solutions with $\mathrm{lb}(\mathbf{x}) > \mathrm{ub}(\mathbf{x})$ are not member of $\Psi_\mu(M)$ since any of these solutions is clearly dominated by more than μ other solutions from M. □

From these lemmata, we can derive modified versions of algorithm 6 and 7. The idea for the P_ω-filter is to sort the population $G_t \cup P_t$ by lower confidence bounds or upper confidence bounds for a given confidence level p_α, respectively. Thereby, we take into account the age of the individuals that correspond to the solutions and favor solutions that have an age lower than κ. Then, for the P_ω-filter algorithm, the lower confidence bound for the $\mu + 1$-th element is detected. All upper confidence bounds of elements in G_t are compared to this value and rejected, if they are not smaller than it.

Accordingly, the R_ω-filter first sorts the population $G_t \cup P_t$ by upper confidence bounds in order to detect the μ-th lowest upper confidence bound, thereby considering additionally, whether the age of the individuals in P_t is lower than κ or not. For all elements of $\mathbf{x} \in Q_t$, the lower confidence bound $\mathrm{lb}_\omega(\mathbf{x})$ is compared to the μ lowest upper confidence bound and the element is accepted, if its lower confidence bound is lower or equal than this upper confidence bound.

It could be remarked that algorithm 6 and 7 need not be introduced. However, the formerly introduced algorithms are more transparent and thus easier to implement. Moreover, unlike the algorithms based on lemma 4, algorithms 6 and 7 can be generalized for multi-objective optimization.

Example: The P_ω-filter and the R_ω-filter are visualized by means of examples in figure 4.2.7 and figure 4.2.6. The parent population P_t={xp1, xp2, xp3 } and the offspring population G_t= {xo1, ..., xo5 } are considered as one set and the $\mu = 3$ best solutions shall be filtered from all parent and offspring solutions (here the case of plus-selection is assumed). The parent solutions have been evaluated precisely, while the offspring solutions have been exactly evaluated. Only solution xo1 passes the P_ω-filter, since it has an upper confidence bound that is exceeded by the $\mu + 1$ lowest lower confidence bound. Contrary to this, three offspring individuals pass the R_ω-filter, because they have lower confidence bounds that do not exceed the μ lowest upper confidence bound. □

It becomes apparent that in comparison to the formerly introduced IPE-filters the interval based P_ω-filter and R_ω-filter make extensive use of the information provided by the parent population.

4.2.4 Invariant permeability relationships between filters with variable output size

Now, we will emphasize invariant relationships among the identified filters with variable output size. By invariant relationships, those relationships that do not depend on the

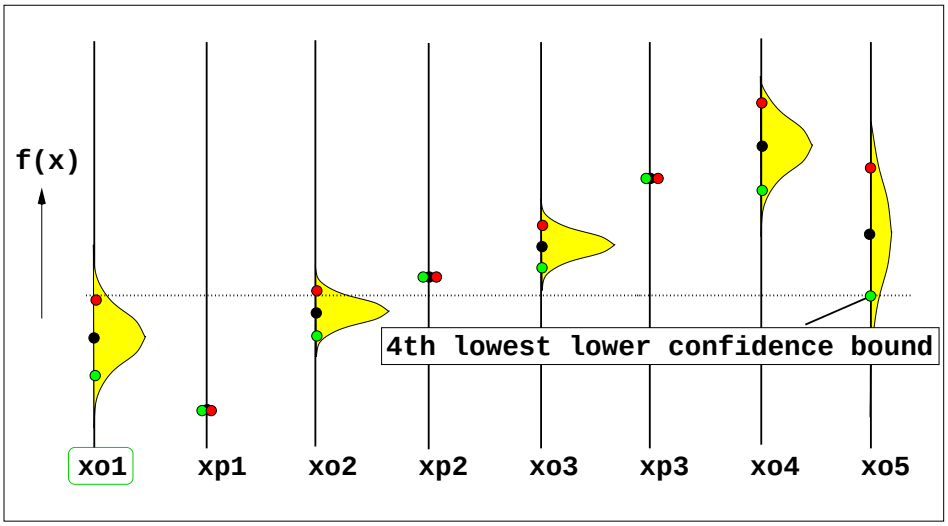


Figure 4.2.6: P_ω-filter for a parent population $P_t = \{\texttt{xp1}, \texttt{xp2}, \texttt{xp3}\}$ and offspring population $G_t = \{\texttt{xo1}, \ldots, \texttt{xo5}\}$. The offspring individual `xo1` will pass the filter, since its predicted upper confidence bound is smaller than the μ+1-th lowest predicted upper confidence bounds of all other solutions.

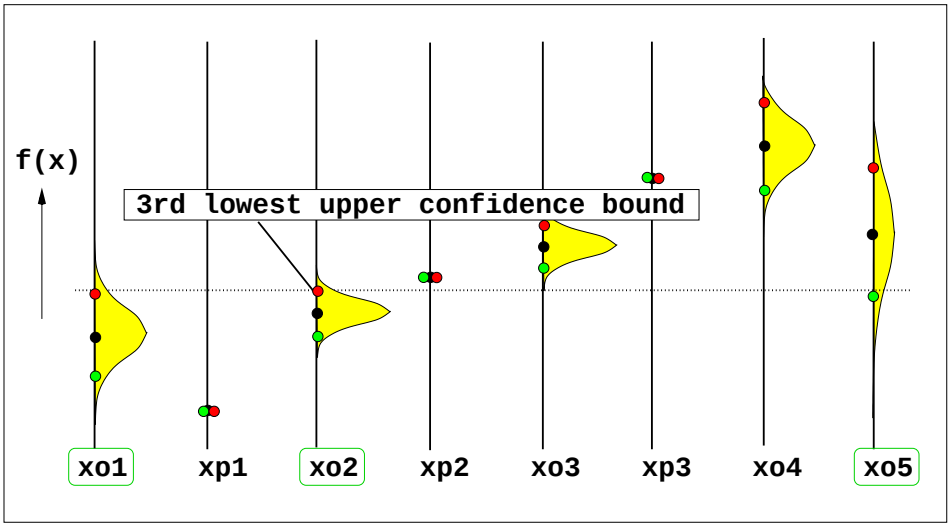


Figure 4.2.7: R_ω-filter for a parent population $P_t = \{\texttt{xp1}, \texttt{xp2}, \texttt{x3}\}$ and offspring population $G_t = \{\texttt{xo1}, \ldots, \texttt{xo5}\}$. The offspring individuals `xo1`, `xo2` and `xo5` will pass the filter, since their approximate lower confidence bounds do not exceed the predicted μ-th lowest upper confidence bound.

particular choice of the set of input solutions are meant. In particular, we are interested in relationships between output sets of filters for the same input sets G and P, without further specifying the input set.

Before, we start the discussion, the concept of *permeability* shall be introduced. Comparing two filters F_A and F_B we say F_A *is less permeable than* F_B, iff $\forall M \in \wp(\mathbb{I}), P \in \wp(\mathbb{I})$: $F_A(M,P) \subseteq F_B(M,P)$. Here, $\wp$ denotes the power set function that yields the set of all subsets for a given set.

The permeability forms a preorder on the set of IPE-filters. The sole minimal element is always the filter that rejects all solutions and the sole maximal element of this preorder is the filter that lets all solutions pass. We will summarize the permeability relationships among the filters with variable output size by means of the following lemma

Lemma 5. Let us assume G denotes a set of approximately evaluated offspring individuals and P denote a set of precisely evaluated parent individuals. Furthermore, consider LBI_ω-filter and MLI-filter for the case $\kappa = \infty$, only. Then we can establish the following relationships among IPE-filters with variable output size:

(1) The LBI_ω-filter is more permeable than R_ω-filter.

(2) With decreasing ω the LBI_ω and the R_ω-filter get less permeable.

(3) The MLI filter is less permeable than the LBI_ω filter.

(4) The P_ω-filter is less permeable than the MLI filter.

(5) With increasing ω the P_ω-filter gets less permeable.

Proof. We will consider the statements one by one: Statement (1) is true, since the μ lowest upper confidence bound will never be larger than the μ-th worst fitness value for the parent population, but it might be smaller. Hence, it is more difficult to pass the R_ω-filter than the LBI_ω-filter. Statement (2) is true, because if ω decreases, more lower confidence bounds will exceed the threshold value of the LBI_ω-filter. Also, for the R_ω-filter more individuals will be selected. The reason for this is, that with decreasing ω the μ-th lowest upper confidence bound will decrease. And, as also the lower confidence bounds of all solutions decrease or stay equal, more lower confidence bounds will exceed the μ-th lowest upper confidence bound and thus more solutions will be rejected by the filter. (3) is true since the MLI is equivalent to the LBI_ω-filter for $\omega = 0$. The rest follows from (2). In case of statement (4) it suffices to state, that the μ lowest lower confidence bound is always lower or equal to the μ-th best value. For proving statement (5), we can argue that more individuals are possibly selected, if the uncertainty about the approximation grows. Concretely speaking, if the lower confidence bounds decrease and the upper confidence bound increase, further individuals will have a upper confidence bound that is higher than the decreased μ lowest lower confidence bound. That is the reason, why less individuals might be identified as being selected by the replacement operator. □

The hierarchy of filters due to their permeability have been depicted for $\kappa = \infty$ in figure 4.2.8. It makes clear that the choice of the filter type as well as the ω parameter can be

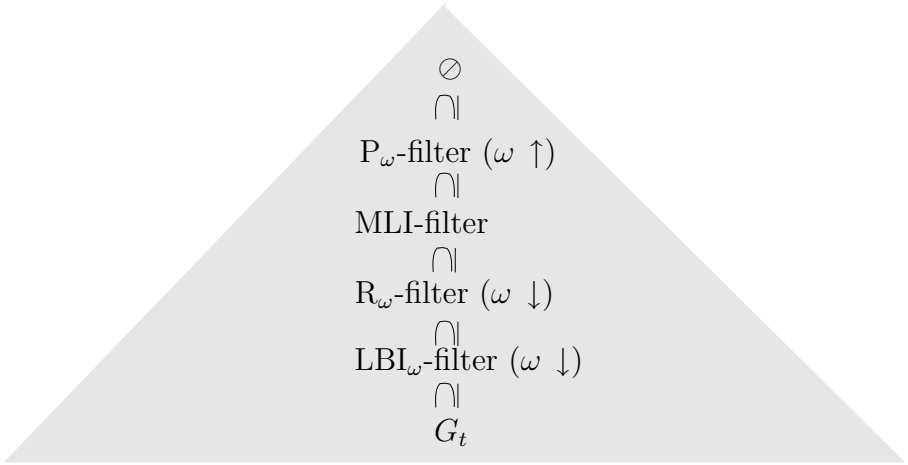


Figure 4.2.8: Permeability of filters with variable output size. Here, the symbol '$\omega \uparrow$' indicates that with increasing ω the permeability decreases. Likewise, the symbol '$\omega \downarrow$' indicates that with decreasing ω the permeability increases.

used for regulation of the permeability. Furthermore, it shows that in the one-dimensional case the considered filters are linearly ordered.

4.2.5 Refinements of interval based filters

After highlighting some relationships between stochastic ranking and selection procedures and IPE-filters, designs for filters with adaptive output size will be discussed that are closely related to such procedures. Such filters might be seen as refinements of interval-based filters.

Stochastic ranking and selection methods

Closely related to the field of metamodel-assisted optimization is the field of optimization with noisy function evaluations [AB03, BBPM05, BT05, JB05, Rud01]. Buchholz and Thümmler [BT05] proposed statistical selection procedures based on *two-stage sampling* (TSS) for evolution strategies that work with approximate, uncertain, evaluations. In their scenario, stochastic simulation procedures designate the source of uncertainty. The uncertain evaluations are modeled by gaussian distributions. The proposed evolution strategies aim at finding solutions with minimum mean response.

They based their selection procedures on a *two-stage selection procedure* for selecting the μ best individuals from a sample of k individuals. The ranking is based on the a-priori unknown mean values $\mathrm{E}(Y_i)$ of the random variables $Y_i, i = 1, \ldots, k$ that describe the responses.

A correct solution is a subset selection where indeed the μ best solutions are contained in the selected subset. Here, as 'best' μ solutions they define the best μ solutions according to the mean values of the response.

Law and Kelton [LK00] clarified that without further assumptions algorithms cannot guarantee correct solutions for a probability $p_\alpha > 0$. Therefore, it is common practise to introduce a *indifference-zone parameter* z^*, meaning that the user does not care for differences between the $\mathrm{E}(Y_1)$ and $\mathrm{E}(Y_2)$, whenever these differences stay below z^*.

Koenig and Law [KL85] proposed a two-stage sampling procedure for selecting a subset of size μ containing the l best (with regard to their mean value) out of k independent normal distributed random variables. In a first stage, their TSS procedures executes a constant number of n_0 replicated evaluations for each individual. From the resulting response values it estimates mean values and variances of the random variables. In a second stage, additional evaluations, the number of which can vary from individual to individual, are performed.

Relationship to filters in the metamodel-assisted evolutions strategy

Let us now relate TSS approach to the scenario in which we apply metamodel-assisted evolution strategies. Recall, that in this scenario the likelihood of the yet unknown outcome of a deterministic function evaluation is described by a gaussian distribution. Recall, that after the first sampling stage in TSS we obtain a set of k mean values and variances describing gaussian distributions. Within the MAES scenario, this corresponds to the information we get for the search points after evaluating the metamodel. Thus, at first glance both approaches seem to be closely related. But there are some important differences between both approaches:

- In the MAES approach the evaluation of an input vector with the simulator always means to obtain the precise result for that particular input vector.
- Rather than estimating the μ smallest mean values of the given random variables, in the MAES we are aiming at selecting those random variables which realizations are most likely the μ smallest solutions.

Clearly, this scenario differs from the scenario of stochastic selection and different subset selections are to be sought than the ones proposed by Koenig and Law [KL85].

There are now at least two options we can take:

1. Select a set of size $\nu \geq \mu$ that contains the μ best individuals with a probability equal or higher than p_α without any resampling.
2. Proceed with *resampling* until we can separate the μ best solutions from the $k - \mu$ solutions with a probability higher than p_α.

The first procedure seems to be more appealing, because it allows us to stay within the proposed MAES schema that separates between pre-selection and evaluation phase. A first approach for capturing such a set is provided with the interval-based filters. A conservative bound on p_α is given by $(1 - 2\Phi(-\omega))^k$, if k is the number of approximated individuals.

Note that in order to obtain separation procedures for which better bounds for p_α can be stated can be sought. However, they likely become very complicated as the solution of the following closely related problem may indicate.

Let $\mathcal{Y} = \{\mathcal{Y}_1, \ldots, \mathcal{Y}_k\}$ denote a set of independent gaussian distributed random variables with mean value $\hat{y}_i$ and some positive standard deviations $\hat{s}_i$. Our objective is to find a μ-sized subset $Y^*_\mu \subset Y$ that, with maximal probability, will provide μ smallest out of k realizations. Let $Pr(\mathcal{Y}_\mu \prec \bar{\mathcal{Y}}_\mu)$ denote the event that the variables of some μ-sized subset $Y_\mu \subset \mathcal{Y}$ generate samples that are all smaller than the samples from $\bar{\mathcal{Y}}_\mu = \mathcal{Y} - \mathcal{Y}_\mu$. An expression for $Pr(Y_\mu \prec \bar{Y}_\mu)$ is can be derived by integrating over all u the conditional probability density for the realizations of $\mathcal{Y}_\mu$ being below or equal u and realizations of $\bar{\mathcal{Y}}_\mu$ being above u under the condition that u is the value of the μ lowest realizations of random variables in $\mathcal{Y}$:

$$p(\mathcal{Y}_\mu \prec \bar{\mathcal{Y}}_\mu) = \int_{u=-\infty}^{\infty} p_{\mu;k}(u) \prod_{Y \in \mathcal{Y}} \Phi\left(\frac{x - \mathrm{E}(Y)}{\mathrm{S}(Y)}\right) \prod_{Y \in \bar{\mathcal{Y}}} \left[1 - \Phi\left(\frac{x - \mathrm{E}(Y)}{\mathrm{S}(Y)}\right)\right] du. \quad (4.2.18)$$

Here, as usual, E(.) denotes the mean value of the random variable and S(.) denotes the standard deviation. Moreover, $p_{\mu;k}(u)$ denotes the probability density function for the μ-th lowest coordinate from a realization of the joint distribution of $Y_1, \ldots, Y_k$. Borrowing a result from the theory of order statistics (cf. David [Dav81], page 22), we obtain

$$p_{\mu;k}(u) = \sum_{i=1}^{k} \sum_{S_i} \prod_{l=1}^{i} \Phi\left(\frac{\mathrm{E}(Y_{j_l}) - u}{\mathrm{S}(Y_{j_l})}\right) \prod_{l=i+1}^{k} \left[1 - \Phi\left(\frac{\mathrm{E}(Y_{j_l} - u)}{\mathrm{S}(Y_{j_l})}\right)\right]. \quad (4.2.19)$$

Here the summation S_i extends over all permutations with $(j_1, \ldots, j_k)$ for which $j_1 < \cdots < j_i$ and $j_{i+1} < \cdots < j_n$. Except in special cases, the computational effort for computing these expressions is considerable. Moreover, the maximization of expression 4.2.18 over all μ-sized subsets from $\mathcal{Y}$ has to be carried out, and it is at least not obvious to see how this procedure can be efficiently implemented.

Thus, more sharp separation techniques that take into account the precise shape of the gaussian distribution likely come at the cost of a significant computational overhead. Future studies will have to reveal if this overhead is justified by a considerable increase in algorithm performance, or if it is possible to increase the efficiency of the such separation algorithms.

Concerning the resampling approach (alternative 2), we note that it is also possible to resample points which are not among the set of search points in the offspring population, in order to reduce the variances. This opens up a large number of resampling possibilities resampling. A thorough discussion of them would exceed the scope of this thesis and is thus left to future research.

4.3 Global convergence behavior

Independent of the filter that is employed, it is relatively easy to prove the probabilistic global convergence of the MAES for regular functions under certain weak restrictions for the algorithm. This proof will be discussed in the first subsection. The second subsection

addresses the difficulty to develop a theory for the dynamics of the MAES and motivates, why experiments are needed in order to answer questions of practical importance about the algorithms behavior.

4.3.1 Proof of global convergence

If certain simple requirements are met, it is always possible to prove the probabilistic global convergence of the MAES for all *regular functions*.

Let us first define a regular function as a function $f : \mathbb{S} \to \mathbb{R}$ with

(A) f is continuous

(B) $\mathbb{S}$ is a closed set,

(C) $\forall \mathbf{x}' \in \mathbb{S} : \forall \epsilon > 0 :$ the set $\{\mathbf{x} \in \mathbb{S} | \mathbf{x} \neq \mathbf{x}' \wedge f(\mathbf{x}) \leq f(\mathbf{x}') + \epsilon\}$ is non-empty.

Now, we can prove the theorem about the global convergence of the MAES:

Theorem 2. Let us consider an regular function f and an arbitrary $(\mu, \kappa, \nu < \lambda)$ MAES with $\nu > 0$ and a minimal step-size of $\sigma_{\min} > 0$. In addition, let $\Delta_t = ||\mathbf{x}^t_{best} - \mathbf{x}^*||$ denote the distance of the search point $\mathbf{x}^t_{best}$ in iteration t to the global optimum $\mathbf{x}^*$ of f. Then

$$\Pr\{\Delta_t \quad \overrightarrow{t \to \infty} \quad 0\} = 1 \tag{4.3.20}$$

.

Proof. From the construction of the algorithm it follows:

$$\forall t \geq 0 : \Delta_{t+1} \leq \Delta_t \tag{4.3.21}$$

and from the definition of the global optimum we get

$$\forall t \geq 0 : \Delta_t \geq 0. \tag{4.3.22}$$

With proposition 4.3.21 and 4.3.22 it follows that there exists a limit value

$$\Delta_t \quad \overrightarrow{t \to \infty} \quad \Delta_\infty. \tag{4.3.23}$$

We show that $\Delta_\infty > 0$ leads to a contradiction and thus (with proposition 4.3.22) $\Delta_\infty = 0$ is true. Let f^∞_{best} denote the function value for f^t_{best} for $t \to \infty$. Then let

$$\epsilon = (f^\infty_{best} - f^*)/2 \tag{4.3.24}$$

and it follows from the regularity of f that

$$X^*_\epsilon = \{\mathbf{x} \in \mathbb{S} | |f(\mathbf{x}) - f^*| \leq \epsilon\} \tag{4.3.25}$$

is a nonempty set and thus there exists a hyper-sphere $K = \{\mathbf{x} \in \mathbb{R}^n | |\mathbf{x} - \mathbf{x}'|^2 \leq r^2\}$ with $r > 0$ and $\mathbf{x}' \in K$ such that $K \subseteq X^*_\epsilon$. Now, the probability that a new offspring generated by the EA in a d-dimensional search space is part of K is lower bounded by

$$p_\epsilon = \min_{\mathbf{x}' \in \mathbb{S}} \left(\frac{1}{\sqrt{2\pi}\sigma^2_{\min}} \right)^d \cdot \int_{\mathbf{x} \in K} \exp\left(\frac{1}{2\sigma^2_{\min}} \right) \cdot (\mathbf{x} - \mathbf{x}')^T \cdot (\mathbf{x} - \mathbf{x}') d\mathbf{x} \ > 0 \tag{4.3.26}$$

for a limited step-size of $\sigma_{\min}$. Given these preliminaries, it can be concluded that the probability that all individuals of a generation are placed inside of K can be calculated as $(p_\epsilon)^\lambda > 0$. Due to $\nu \geq 1$, this is a lower confidence bound for the probability that at least one individual is placed inside of K passes the filter in the pre-selection and thus is considered for replacing the currently best individual in one generation of the MAES.

Now, we can derive a lower bound for the probability that K is hit at least once after q generations as:

$$\Pr(\bigvee_{i=0}^{q}(\mathbf{x}_{best}^i \in K)) = 1 - (1 - (p_\epsilon)^\lambda)^q, \tag{4.3.27}$$

and hence

$$\Pr(\bigvee_{i=0}^{q}(\mathbf{x}_{best}^i \in X_\epsilon^*) \quad \overrightarrow{q \to \infty} \quad 1 \tag{4.3.28}$$

With expression 4.3.22 and expression 4.3.24 we get an contradiction to our assumption that $\Delta_\infty > 0$. □

The result assures that the MAES does what it is expected to do, that is to converges in probability to the global optimum. However, the result does not provide insights on how fast the global optimum is approached. The answer to this question is much more difficult to be obtained and will be addressed in the subsequent sections.

4.3.2 Convergence dynamics

In this subsection we will briefly study some aspects of the convergence dynamics for the MAES. A general expression for the convergence velocity of the MAES will be derived. Then it will be displayed, why standard techniques for the theoretical analysis of ES cannot be applied in the context of the MAES and empirical studies are needed.

We will compare the convergence speed of the ES and the MAES with the same settings of μ, κ, and λ. First, a general definition for the convergence velocity of ES (algorithm 2) and MAES (algorithm 5) shall be derived. Let t denote the number of generations of an EA. Suppose $\Delta_t = |\mathbf{x}_{best}^t - \mathbf{x}^*|$ denotes the distance to the global optimum in one iteration (generation) t of an EA and let τ_t denote the expected time for all objective function evaluations conducted during one generation. Then the convergence velocity of the EA can be defined as

$$v_t = \frac{\mathrm{E}(\Delta_{t+1} - \Delta_t)}{\tau_t}. \tag{4.3.29}$$

For standard techniques in the analysis of the ES (cf. Beyer [Bey01]) the value of τ_t is assumed to be constant for constant λ. In the context of approximate evaluations, this is the sum of evaluation times for individuals in a single generation. In the MAES every individual is evaluated once by using the approximate model and, if it passes the filter, also by using the precise model. Let T_a denote the expected time it takes to evaluate the individuals by means of the approximate model and T_p the expected time it takes to evaluate the precise model. Furthermore, let ν denote the number of pre-selected individuals for one generation. Then

$$\tau_t = \lambda \cdot T_a + \nu \cdot T_p. \tag{4.3.30}$$

Now, assuming that within one generation of the MAES the same progress is achieved as in one generation of the ES, we can compare the convergence speed of the MAES to the convergence speed of the ES by calculating the *convergence speed-up* s_t^{MAES}:

$$s_t^{MAES} = \frac{v_t^{MAES}}{v_t^{ES}} - 1 = \frac{\lambda \cdot T_p}{\lambda \cdot T_a + \nu \cdot T_p} - 1. \tag{4.3.31}$$

In order to find out the break-even point, where the speed-up is exactly zero, we introduce dimensionless numbers $\alpha = T_a/T_p$ and $\gamma = \nu/\lambda$. Now, equation 4.3.31 reads:

$$s_t^{MAES}(\alpha, \gamma) = \frac{1}{\alpha + \gamma} - 1. \tag{4.3.32}$$

Hence, whenever $\alpha + \gamma < 1$ we achieve an acceleration, otherwise the algorithm slows down. It is often assumed that T_a is orders of magnitude smaller than T_p. In this case the acceleration of the MAES depends mainly on γ since $\alpha \approx 0$. Whenever the acceleration is greater than this, we can conclude that this has to be attributed to another positive effect the metamodel-assistance has on the dynamics of the search, e. g driving the search into unexplored regions of the search space.

Among the pre-screening criteria that have been discussed in section 4.2 only the LBI_ω-filter aims at avoiding the rejection of individuals that might be successfully selected in the replacement, if evaluated precisely. Hence, only for this strategy it is a good assumption that $\text{E}(\Delta_{t+1}^{ES} - \Delta_t^{ES})$ and $\text{E}(\Delta_{t+1}^{MAES} - \Delta_t^{MAES})$ are the same. Here Δ_t^{ES} denotes the distance to the optimum for the ES, and Δ_t^{ES} the distance to the optimum for the MAES. In that case the extended expression for the acceleration coefficient should be used, that is defined as

$$s_t^{MAES} = \frac{\lambda \cdot T_p}{\lambda \cdot T_a + \nu \cdot T_p} \cdot \frac{\text{E}(\Delta_{t+1}^{MAES} - \Delta_t^{MAES})}{\text{E}(\Delta_{t+1}^{ES} - \Delta_t^{ES})} - 1. \tag{4.3.33}$$

If a positive acceleration is desired, it has to be assured that

$$\frac{1}{\alpha + \gamma} \geq \frac{\text{E}(\Delta_{t+1}^{ES} - \Delta_t^{ES})}{\text{E}(\Delta_{t+1}^{MAES} - \Delta_t^{MAES})} \tag{4.3.34}$$

.

The theoretical analysis of the convergence speed of the MAES faces the serious difficulty of determining the progress rate at a certain iteration. In literature, most analysis methods make use of the Markov property of the ES, viewed as a discrete dynamical system with index t. In particular, for the ES (algorithm 2) it can be assumed that the randomized generation of any new population P_{t+1} only depends on the previous population P_t. This feature can be used to derive terms for the expected progress rate for simple function classes. An extensive discussion of the theory of the ES dynamics can be found in [Bey01]. However, for the MAES the generation of population P_{t+1} is a random procedure that depends on all previous populations $P_0, \ldots, P_{t+1}$. Noting that already the rigorous analysis of the dynamic behavior of the standard ES on some simple functions is only possible if simplifying assumptions are made (cf. [Bey01]), loosing the Markov property makes it even more difficult to derive analytical expressions for the convergence velocity. Hence, computer experiments are needed to investigate the dynamical behavior of the MAES.

4.4 Performance and indicator measures

The experimental analysis of algorithms involves many decisions. The most important ones are the choice of the appropriate instantiations of the algorithms in charge, the set of test problems, the performance measures and last but not least measures that allow for a deepened understanding of why the algorithm behaves the way it does. The latter will be termed indicator measures.

In this section these aspects of the experimental setup for single-objective optimization with time consuming evaluations is addressed, before, in section 4.5, we will present results of the tests and discuss them.

4.4.1 Performance measures

Typically, the major CPU time during a run of the MAES is spent on the precise objective function evaluations. Thus the number of time-consuming objective function evaluations it takes to achieve a certain precision of the approximation to the optimum will be studied in the following, assuming that the time for approximate objective function evaluations can be neglected. Indeed, for problems in industrial design optimization it often takes minutes up to hours of time to evaluate one solution candidate, thus the time to evaluate a metamodel is insignificant. For a given test problem the history of the best found function value $f_{best}(n_e)$ after n_e objective function evaluations can be plotted against n_e. From this plot we can obtain the behavior of the MAES for different running times.

Since the ES, as well as the MAES, is a randomized algorithm, it is insufficient to provide only a single run for the performance test in order to gain insights into the typical behavior of the algorithm. Several runs have to be averaged in order to detect significant effects. A straightforward approach would be to calculate the average best function value achieved after n_e objective function evaluations for several runs of the same ES. Usually, it is better to plot the median instead of the arithmetic mean in order to prevent outliers to disturb the results. Especially for multimodal problems the arithmetic mean can be misleading, if there are substantial numerical differences between the local optima.

In order to measure robustness of a strategy, also some quantile should be displayed, e. g. the 80% quantile. This is the result for which 80% of runs have obtained a better function value after a given number of objective function evaluations t_{eval}. Again, we can give the same arguments as mentioned above for the arithmetic mean, why the 80% quantile is a better choice than the standard deviation. Generally, the confidence margins that are due to the standard deviation can be quite confusing, because they do not show the skewness of a distribution and thus are not very specific about the most frequent direction of the deviation.

4.4.2 Accuracy and selectivity measures

The selectivity of the metamodel-based predictions during the evolution can be displayed by plotting the difference between predicted and exact objective function values, such as in $y \sim \hat{y}$, $y \sim \hat{y}_{lb}$ diagrams (see e.g. [EN04a]). The $y \sim \hat{y}$ plot indicates the correlation

between predicted values and estimated values, while the $y \sim \hat{y}_{lb}$ plot indicates whether $\hat{y}(\mathbf{x}) - \omega \cdot \hat{s}(\mathbf{x})$ was an adequate choice for a lower confidence bound for the predicted values or not.

It is a well known fact, that ES are rank-based optimization strategies that are invariant to monotonic transformations of the objective function. Hence, a metamodel used in conjunction with ES can be regarded as successful, if it is able to predict whether or not a new individual is an improvement with respect to the parent population P_t. For instance in the $(\mu + \lambda)$ ES, any filter that can identify the subset G_t of the offspring population with $\nu \leq \mu$ individuals, that are worth entering the next generation, is fully adequate. The so-called *retrieval quality* of any filter can be measured by means of the *recall* and *precision* measures defined below.

Let $M_\mu(A)$ denote the subset defined by the μ best solutions in A. The filter operator in the MAES aims at identifying the members of $G_t \cap M_\mu(G_t \cup P_t)$ that will then enter the next generation. Thus, it is desirable that

$$Q_t \approx G_t \cap M_\mu(G_t \cup P_t). \tag{4.4.35}$$

It is reasonable that none of the filters can always retrieve the ensemble of relevant individuals out of G_t. As has already been described in a less formal manner in section 4.2 a non-satisfactory filter is one that: (A) selects too many individuals that do not belong to $G_t \cap M_\mu(G_t \cup P_t)$ and thus are irrelevant or (B) fails capturing a considerable part of $G_t \cap M_\mu(G_t \cup P_t)$.

The retrieval selectivity, in relation to the first of the two unpleasant situations that have been mentioned, is defined as precision(t), which expresses the ratio of the relevant solutions retrieved to the total number of retrieved solutions, as follows:

$$0 \leq \text{precision}(t) = \frac{|M_\mu(G_t \cup P_t) \cap Q_t|}{|Q_t|} \leq 1. \tag{4.4.36}$$

On the other hand, the ratio of the relevant solutions retrieved from G_t to the number of all relevant solutions in G_t is quantified as follows:

$$0 \leq \text{recall}(t) := \frac{|M_\mu(G_t \cup P_t) \cap Q_t|}{|M_\mu(G_t \cup P_t) \cap G_t|} \leq 1. \tag{4.4.37}$$

In contrast to quantitative measures such as $y \sim \hat{y}$-plots, specificity measures can not be evaluated without conducting extra objective function evaluations, which would cause extra computing cost. Hence, they are mainly useful for statistics on simple academic test cases and not for real world problems.

4.4.3 Indicator measuring the number of inversions

With regard to the criterion based filters it can also be interesting to measure the capability of the MAES to establish a proper order on the subset of solutions. For this we propose to count the number of inversions in the sorting of individuals due to their fitness values. This measure is calculated as follows:

Let the sequence $\pi(1), \ldots, \pi(n)$ denote a permutation of the sequence $1, \ldots, n$. Then, the pair $\mathbf{x}_{\pi(i)}, \mathbf{x}_{\pi(j)}$ is called an inversion, if $\pi(i) > \pi(j)$ and $i < j$. The number of inversions in a permutation can be counted via:

$$\mathrm{Inv}_n(\pi) = \frac{1}{2} \sum_{(i,j) \in \{1,\ldots,n\}^2} \iota(\pi(i) > \pi(j)), \qquad \iota(\pi(i) > \pi(j)) = \begin{cases} 1 & \text{if } \pi(i) > \pi(j) \wedge i < j \\ 0 & \text{otherwise} \end{cases} . \tag{4.4.38}$$

Additionally, we will define the *number of sorted pairs* as

$$\mathrm{Sort}_n(\pi) = n \cdot (n-1)/2 - \mathrm{Inv}_n(\pi). \tag{4.4.39}$$

The number of sorted pairs takes the value of $(n-1) \cdot n/2$, if the sequence is completely ordered and it gets zero, if π represents an inverse order.

In order to test wether an approximate order is significantly better than a pure random ordering, the theorem of Sachkov [Sac97] is useful:

Theorem 3. [Sac97] If ξ_n is a random variable representing the number of inversions in a random equiprobable permutation of n elements, then the random variable

$$\Xi_n = (\xi_n - \mathrm{E}(\xi_n))/\mathrm{Var}(\xi_n) \tag{4.4.40}$$

has a normal distribution with mean 0 and variance 1 as $n \to \infty$.

Margolius [Mar01] found that for $n \geq 10$ the approximation to a standard gaussian distribution is already very close. Hence, we can test the hypothesis, whether the sequence of individuals is ordered randomly or not by means of a simple test:

$$\Pr(\pi \text{ has been produced by random ordering}) = \Phi(\frac{-(N_{inv} + \mathrm{E}(\xi_n))}{\mathrm{Var}(\xi_n)}). \tag{4.4.41}$$

An expression for the mean and variance of the number of inversions in equiprobable permutations has been given in [Mar01]:

$$\mathrm{E}(\xi_n) = \frac{n \cdot (n-1)}{4} \tag{4.4.42}$$

and

$$\mathrm{Var}(\xi_n) = \frac{2n^3 + 3n^2 - 5n}{12} \tag{4.4.43}$$

Example: Consider the aim is to sort a population of 100 individuals by means of predictions for the objective function evaluations. After evaluating the correct ordering, it has been obtained that 3397 pairs of solutions are in the right order. In that case, the maximal number of ordered pairs is 4950. The mean value of ξ_{100} is 2475 and the variance is 169125, and hence the standard deviation is approximately 411. Hence, the probability that by chance more than $3397 (= 2475 + 2 \cdot 411)$ sorted pairs can be found in an random order is given approximately by $\Phi(-2)$ that is less than 2.3%. Thus, it is likely that by means of the approximation a better sorting has been achieved than by pure random sorting.

4.5 Studies on artificial test problems

The aim of the studies is to find out more about the characteristics of the convergence process of the MAES using different filter strategies. First, the test problems for the comparison are displayed. Then, the results of test runs will be described and analyzed.

4.5.1 Test functions for the first comparison

Six different test functions have been chosen in order to test different filters for the MAES for single criterion optimization. Visualizations of these functions can be found in figure 4.5.10.

Sphere problem

$$f(\mathbf{x}) = \sum_{i=1}^{d} x_i^2 \rightarrow \min . \tag{4.5.44}$$

The known minimum of the sphere function is $f(\mathbf{x}^*) = 0$ at the minimizer $\mathbf{x}^* = \mathbf{0}$.

The sphere problem is a simple reference test case. The objective function is strictly convex, quadratic, and also point-symmetric. The center of symmetry is the known minimizer $\mathbf{x}^* = \mathbf{0}$. From test runs on the sphere problem we approximate the best case behavior of the MAES algorithm and we learn about its local convergence speed.

Ellipsoid problem

The ellipsoid problem reads

$$f(\mathbf{x}) = \sum_{i=1}^{d} i \cdot x_i^2 \rightarrow \min . \tag{4.5.45}$$

The known optimum of the ellipsoid problem is $f(\mathbf{x}^*) = 0$ at the minimizer $\mathbf{x}^* = \mathbf{0}$.

The ellipsoid function can be used to study, whether the algorithm can deal with variables that have a different impact on the objective function. Furthermore, this test case is useful to answer the question whether a proper scaling can be learnt or not.

Double sum problem

A difficult quadratic problem is the double sum problem taken from Schwefel ([Sch95], pp. 326):

$$f(\mathbf{x}) = \sum_{i=1}^{d} (\sum_{j=i}^{d} x_j)^2 \rightarrow \min . \tag{4.5.46}$$

Its known minimum is $f(\mathbf{x}^*) = 0$ at the minimizer $\mathbf{x}^* = \mathbf{0}$.

Like the sphere problem and the ellipsoid problem, the double sum problem is unimodal and convex. The function has an elliptic contour hyper-surface ($f(\mathbf{x}) = const$). Let $a_{\max}$ denote the maximal length of one of the semi-axes of the elliptic contour hyper-surface and $a_{\min}$ denote its minimal length, then the condition number of the quadratic problem is given by $K = (a_{\max}/a_{\min})^2$. For moderate numbers of d the condition number increases nearly quadratically with the dimension of the problem (cf. [Sch95]). Hence, the ratio between the largest and smallest length of the semi-axis of the ellipsoids scale linearly. This characteristic is shared with the ellipsoid problem.

In contrast to the ellipsoid problem, the double sum problem is not decomposable. In geometrical terms, the semi axes of the ellipsoids of the contour hyper-surface are not parallel to the coordinate axes. Hence, it is not possible to progress to the optimal solution by adjusting each variable separately. Thus, the double sum problem allows to check the algorithm's capability to deal with interaction effects between input variables.

Step problem

$$f(\mathbf{x}) = \sum_{i=1}^{d} \lfloor x_i \rfloor^2 \rightarrow \min . \tag{4.5.47}$$

The known minimum of this function is $f(\mathbf{x}^*) = 0$ at the minimizer $\mathbf{x}^* = \mathbf{0}$. From a distance this surface of this problem looks like that of sphere model (cf. figure 4.5.10). However, it has a non-steady surface consisting of several plateaus. Like steps they are symmetrically placed around the center zero. The optimal region is the open interval box $]-\mathbf{1}, \mathbf{1}[$, with $\mathbf{1} = (1, \ldots, 1)^T \in \mathbb{R}^d$. For all inputs located in this region the function value is zero. Tests on the step function reveal whether or not the optimization algorithm is capable of dealing with discontinuities and plateaus.

Ackley problem

A multimodal problem with regularly distributed local optima is given by the Ackley problem ([Ack87], pp. 13):

$$-c_1 \cdot \exp\left(-c_2 \sqrt{(\frac{1}{d}\sum_{i=1}^{d} x_i^2)}\right) - \exp\left(\frac{1}{d}\sum_{i=1}^{d} \cos(c_3 \cdot x_i)\right) + c_1 + \exp(1) \rightarrow \min . \tag{4.5.48}$$

$$c_1 = 20;\ c_2 = 0.2;\ c_3 = 2\pi$$

The minimizer of the Ackley function is $\mathbf{x}^* = \mathbf{0}$ with $f(\mathbf{x}^*) = 0$. The function of Ackley is moderate in difficulty. It possesses local optima that are located in symmetrical patterns around the global optimum (cf. figure 4.5.10). The number of optima grows exponentially with the problem dimension.

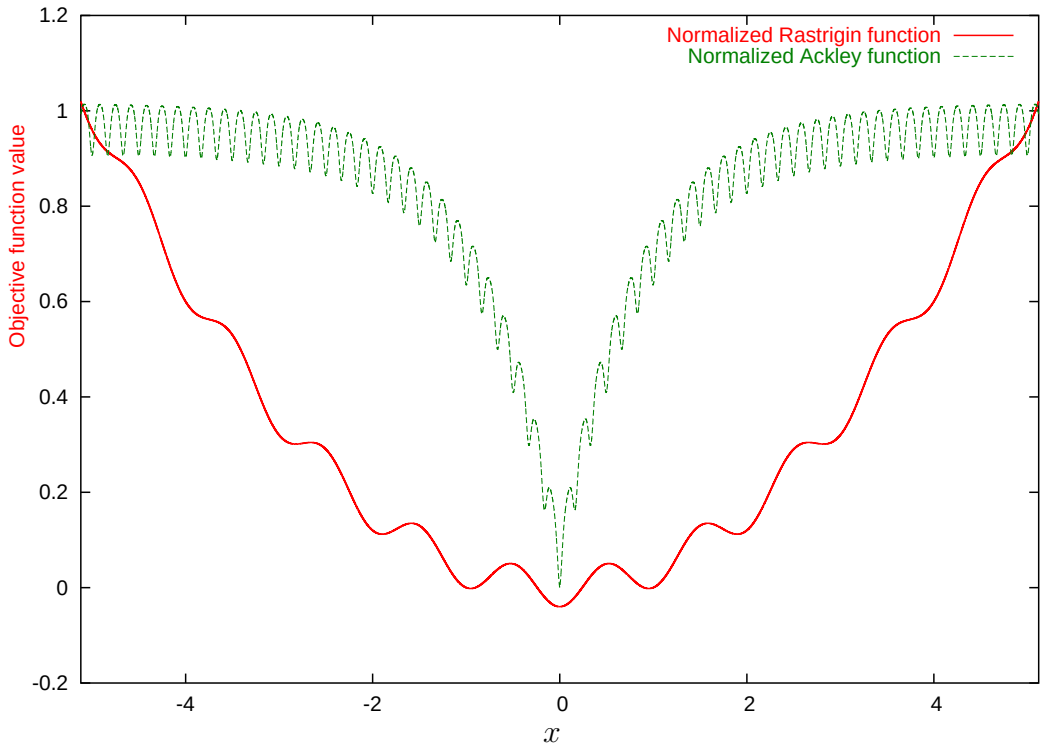


Figure 4.5.9: Comparison of the one-dimensional Rastrigin and Ackley function.

Rastrigin problem

Multimodal optimization is a typical application field for EA. Thus, a further test problem has been studied in the experiments. Like the Ackley problem, also the Rastrigin problem is characterized by regularly distributed local optima and a global trend. In contrast to the Ackley function, the trend of this function is not an exponential function but a quadratic sum. Hence, the global trend of the Rastrigin function is steep at its boundaries and gets flat to the middle of the interval boundaries (see figure 4.5.9).

$$10d + \sum_{i=1}^{d} (x_i^2 - 10\cos(2\pi \cdot x_i)) \rightarrow \min . \tag{4.5.49}$$

$$\mathbf{x} \in [-5.12, 5.12]^d$$

The minimizer of the Rastrigin function is $\mathbf{x}^* = \mathbf{0}$ with $f(\mathbf{x}^*) = 0$.

Fletcher Powell problem

The Fletcher Powell problem, originally proposed by Fletcher and Powell in 1963 [FP63], is also highly multimodal. In contrast to Ackley's function, its optima are irregularly spaced. This makes it difficult to optimize by strategies that exploit the symmetry of an objective function.

$$\sum_{i=1}^{n} (A_i - B_i)^2 \rightarrow \min . \tag{4.5.50}$$

$$A_i = \sum_{j=1}^{d} (a_{ij} \cdot \sin \alpha_j + b_{ij} \cdot \cos \alpha_j) \tag{4.5.51}$$

$$B_i = \sum_{j=1}^{d} (a_{ij} \cdot \sin x_j + b_{ij} \cdot \cos x_j) \tag{4.5.52}$$

The position of the random optima is determined by the random matrices $\mathbf{A} = (\alpha_{ij})$ and $\mathbf{B} = (b_{ij})$. The entries of these matrices have been chosen as suggested by Bäck ([Bäc96], pp. 143). The global optimum is obviously given by: $\mathbf{x}^* = (\alpha_1, \ldots, \alpha_d)^T$ with $f(\mathbf{x}^*) = 0$.

4.5.2 Implementation details

All tests in the comparison have been conducted for a maximum number of 1000 objective function evaluations. This choice was due to the fact that for many industrial problems with time consuming objective function evaluations only a few hundred objective function evaluations can be spent, even in cases where objective function evaluations can be performed in a distributed fashion. Note that we will not only discuss final results after 1000 objective function evaluations. Instead, all results have also been displayed for smaller numbers of objective function evaluations, in order to gain insights on a broad range of possible problem settings.

As a reference strategy three variants of the ES have been chosen:

- $(1 + 10)$-ES: A single-parent ES with small offspring population size
- $(5 + 20)$-ES A multi-membered ES with moderate population size and selection pressure
- (5+35)-ES A multi-membered ES with moderate population size and recommended selection pressure of seven. This setting was studied in [USZ03] within the context of metamodel-assisted evolution strategies.
- $(15 + 100)$-ES: A typical multi-membered ES with large population size

Note that plus strategies have been chosen for our studies, because they turned out to be more reliable on problems where the number of objective function evaluations is small. Since our test runs have been conducted for a number of maximal 1000 objective function evaluations, the (5+20)-ES is considered as a good choice that allows many iterations and also some robustness on multimodal functions. Similar strategies like the (5+35)-ES, that was used in [USZ03], were outperformed by this strategy. Furthermore the $(1 + 10)$-ES has been employed, which is regarded to be less robust but faster on local optimization problems than the multi-membered ES.

The algorithms have been implemented using the TEA C++ library, which is a library for evolutionary algorithms developed at the Collaborative Research Center "Computational Intelligence" at the University of Dortmund [EH01]. All implementations used for the empirical comparison on test functions can be obtained from the authors web-site.

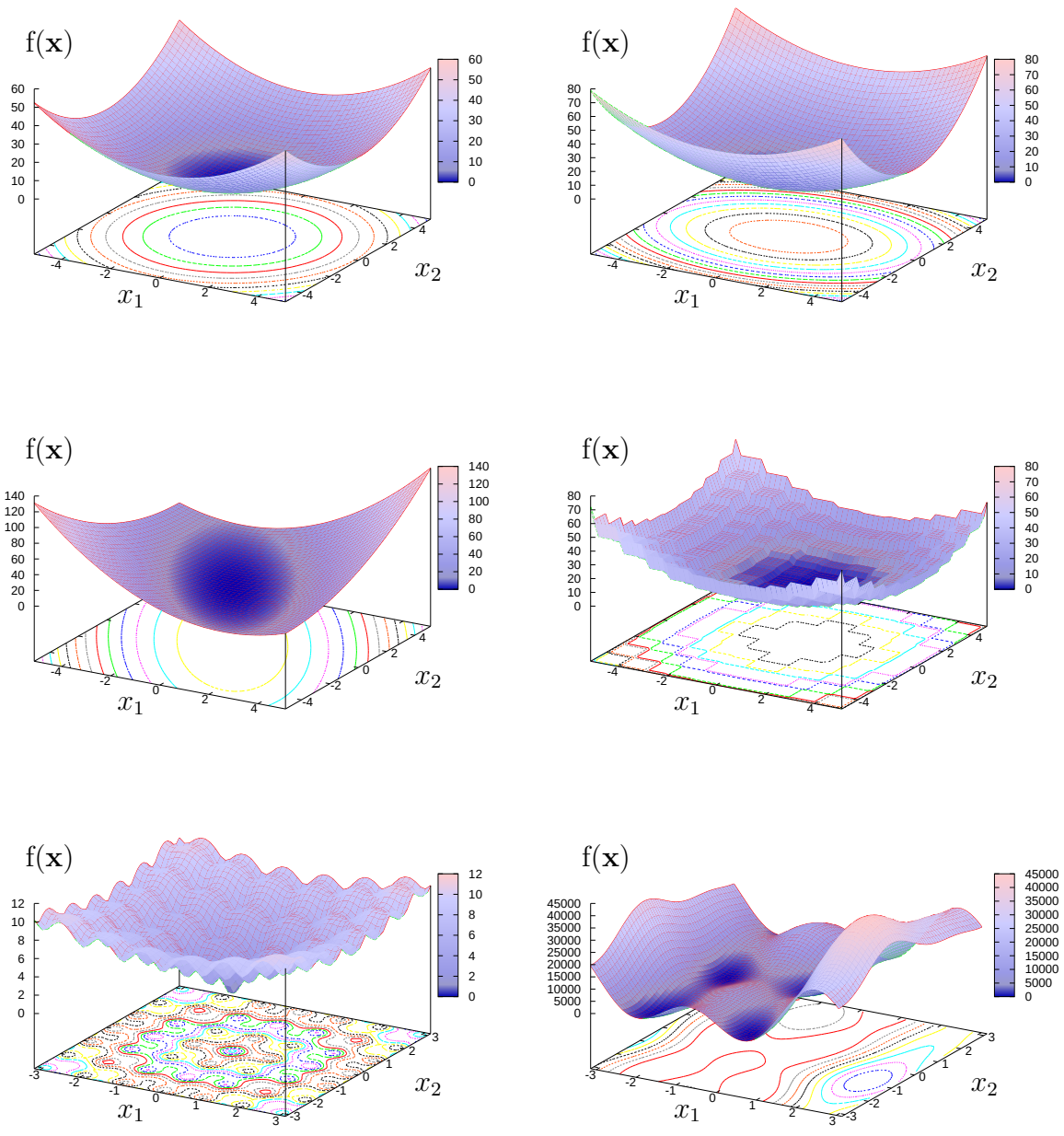


Figure 4.5.10: The pictures show two dimensional instantiations of the sphere problem (upper left), the ellipsoid problem (upper right), the double sum problem (middle left), the step problem (middle right), the Ackley problem (lower left) and the Fletcher Powell function (lower right).

Abbreviation	μ	λ	Type	ν	Filter	ω
`1p10`	1	10	ES	10	-	-
`5p20`	5	10	ES	20	-	-
`5p35`	5	35	ES	35	-	-
`15p100`	5	100	ES	100	-	-
`Mean`	5	100	MAES	20	mean value filter	0
`lb`	5	100	MAES	20	lb_ω filter	2
`PoI`	5	100	MAES	20	PoI-filter	-
`ExI`	5	100	MAES	20	ExI-filter	-
`MLI`	5	100	MAES	variable	MLI_ω-filter	0
`LBI`	5	100	MAES	variable	LBI_ω-filter	1
`rfilt`	5	100	MAES	variable	R_ω-filter	2
`pfilt`	5	100	MAES	variable	P_ω-filter	2

Table 4.5.1: Strategy variants tested in the algorithm comparison.

Table 4.5.1 displays an overview of all strategy variants that have been tested in the algorithm comparison for single-objective optimization. For the MAES filters with constant output size an output size of $\nu = 20$ has been chosen, i. e. $\nu = 20$ individuals have been pre-selected in each generation. Hence, we expect to get a strategy the behavior which is somewhere in between that of a $(5 + 20)$-ES and $(5 + 100)$-ES. The former would be the limit case, if the pre-selection criteria would result in a pure random sorting of the offspring population, and the $(5+100)$-ES behavior would be reproduced if the IPE-filter would work at a recall of one, i. e. if it detects all relevant individuals in each generation.

The motivation behind these settings for the MAES parameters is, to get a strategy that works with a sufficiently large number of generations and also allows for the step size adaptation. As a preliminary for a successful step-size adaptation, λ had to be much greater than μ. Moreover, we wanted to have a multi-membered ES. Though they may perform better on simple quadratic problems, single parent ES have the tendency to get stuck in local optima. It shall be noted, that we were not aiming at the maximization of the convergence speed of the ES on simple local optimization problems, as ES are usually not applied for simple problems. Usually, gradient-based methods - as discussed in section 3.2 - perform much better for this problem class. Rather, we desired a good compromise between convergence reliability and local convergence speed on more complex problems, including difficulties like discontinuities and/or multiple local optima.

In order to increase the number of generations a comparably small selection pressure of $\nu/\mu = 4$ has been chosen. The latter number measures the selection pressure in terms of precisely evaluated offspring individuals. If the metamodel provides good predictions, the selection pressure increases. Assuming that the recall of the employed filter is one, an ES with a selection pressure of $\lambda/\mu = 20$ would be emulated.

For the filters with variable output size it has been assured that at least one individual is pre-selected and precisely evaluated in each generation. By doing so, we intend to avoid stagnation of the optimization process and to maintain the necessary conditions for global convergence on regular functions as discussed in section 4.3.1.

Next, let us discuss the settings for the initialization and variation operators: The ES has been started from a random point in the interval provided with the problem definition. For each of the strategy variants the same set of starting solutions has been used. The initial step-size is 0.1 % of the interval range. The strategy was allowed to move beyond the bound constraints. Hence, the problems were treated as unconstrained optimization problems and the bounds have only been used for generating the starting points for the runs.

For this benchmark comparison an isotropic Gaussian mutations have been employed. The standard deviation has been adapted by means of mutative self-adaptation. The mutation step size a two point operator with a learning rate $\gamma = 1.3$ has been chosen ([Rec94], p. 47).

The choice of $\gamma = 1.3$ has been made due to a suggestion by Beyer ([Bey01], p. 325) for the starting phase (first 1000 generations) of an ES. At least for $\nu/\lambda \ll 1$, the adaptation of the shape of the mutation distribution should be a far less important issue for the metamodel-assisted ES than it is for the simple ES. If we would consider only the ν offspring individuals that are pre-selected, their distribution is significantly influenced by the IPE-filter that selects only the promising solutions from the generated sample. However, it is still very important to adjust the size of the sampling distribution in order to allow for an increase of the sampling radius if the population is far away from the optimum and a decrease of the sampling radius if the MAES approaches the optimum. The latter can well be achieved with a single step size.

However, more sophisticated step-size adaption methods like the individual step-size adaptation (algorithm 4) might also be considered to improve the long term performance of the MAES on local optimization problems with a high condition number.

Moreover, a discrete recombination of the object variables and an intermediate recombination of the step size variables were employed. A local recombination with two parents has been used for intermediate recombination of the step-size. Discrete recombination has been applied for the object variables. The variation operators are comparably simple and transparent and shall allow an easy reproduction of results.

As a random number generator the `rand()` of function from `Gnu C++ V2.95.4` has been used. This random number generator is based on a multiplicative congruential method and supports the new version of the`rand()` function with no problems on lower order bits. The generation of normal distributed random numbers has been done with a technique proposed in [Rin01]. $x = -6 + \sum_{i=1}^{12} U_i(0,1)$. Here, $U_i(0,1), i = 1, \ldots, 12$ denotes uniformly distributed random numbers. The resulting number x is approximately distributed like the standard normal distribution. A normal probability plot for a sample of 50000 random numbers is depicted in figure 4.5.11. The deviation from the true normal distribution is insignificant. Only in the outer regions which are sampled with a probability of less than 0.001 some slight deviations from the true normal distribution appear.

Though it was not used in this work, for future studies we consider the use of the Box-Muller transformation in its polar form as proposed by Carter [Car94]. This generator of random numbers has the advantage that it needs less random numbers and produces random numbers with infinite support. The latter is especially important for studies of

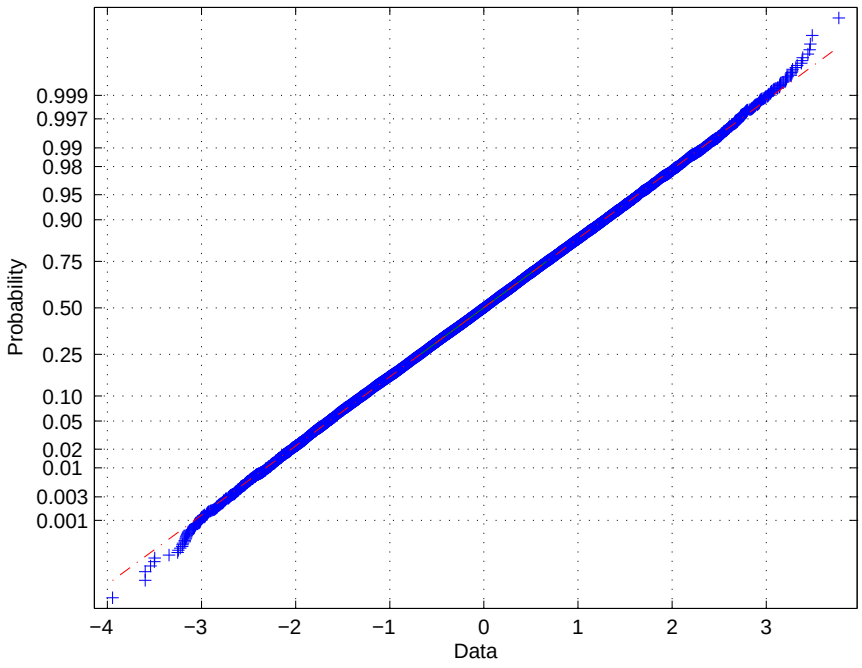


Figure 4.5.11: Normal probability plot of the 50000 samples from the pseudo random number generator used in the experiments for generating normally distributed pseudo random numbers.

the long-term behavior of the MAES, as the infinite support of the gaussian distribution guarantees the convergence for $t \to \infty$ on regular functions (see section 4.3).

The metamodel that has been chosen was an implementation of the gaussian random field model using cross-validation for obtaining the correlation parameter θ. A simple grid search method has been used for the calibration of this parameter. For the inversion of the correlation matrix, an implementation of the LU factorization by M. Dinolfo [Din98] has been used. If m is the number of samples, the execution time of this code is proportional to m^3. For each inverted matrix, by checking $\det(A \cdot A^{-1}) = 1$ for each inversion, it has been assured that the matrix inversion was successful. A minimal distance between neighboring solutions of 10^{-8} have been demanded in the selection of neighbors in order to avoid singularities. Furthermore, the metamodel accuracy has been restricted to theta values between 10^{-6} and 10, to avoid long running times for the calibration of the metamodel parameters.

Furthermore we used a local metamodeling technique. This means that for each predicted point a metamodel has been trained from its neighboring solutions. The number of neighboring solutions in the test runs was 30. In the transient phase, when the database was filled with less than 30 points, objective function evaluations were made instead of metamodel predictions. The calculation time for predictions was about 0.1 seconds (on an Intel Pentium 4 with 2 GHz) in the beginning of the search (database with 30 points) and about 0.3 seconds in the final stage (database with 1000 points). This time consumption can be neglected, if the time consumption of objective function evaluations lies in the range of several minutes, but it makes the implementation of statistical comparisons on

the suggested benchmark problems already very time consuming. Here we note that one run of the $(5, 20 < 100)$-MAES with 1000 objective function evaluations demands 5000 evaluations of the metamodel, which results in a total time consumption of about 20 minutes for a run on a single processor machine.

4.5.3 Discussion of results

Next, we will discuss the results obtained for the MAES variants listed in table 4.5.1. First, one by one, the results for filters with a constant output size will be studied. Later, we take a look at filters with an adaptive mechanism to control the output size.

Filters with constant output size

We start with a comparison of MAES versions that work with constant output size filters. As a consequence of this restriction, the differing performance of the strategies reported cannot be attributed to a different number of generations or different population size parameters. It is merely attributed to the choice of the criterion for selecting the subset. The mean value, lb_ω, PoI and ExI filter have been tested.

Later it will be demonstrated that by choosing values of the confidence factor ω and output size ν we can shift the search characteristics on the trade-off from a fast local convergence speed with low robustness to a more robust behavior with decreased local convergence speed. As a starting point for our comparison we chose a $(5 + 20 < 100)$-ES. For the lb_ω filter ω has been set to a value of 2.0. This choice of parameters is a good compromise solution, that leads to a high robustness on difficult test problems, accompanied by a convergence speed to local optima, that is still significantly higher than that of most conventional ES.

Note, that we will provide more details on the choice of ν, μ and ω in section 4.5.5.

Performance on the sphere problem in different search space dimensions Table 4.5.2 shows the results for the sphere problem in different dimensions of the search space. For each dimension we have plotted the median and the 80th percentile of the best found function value so far. In order to get statistically significant results, 20 runs have been conducted for each strategy.

The first row of table 4.5.2 displays results on the convergence dynamics measured on the five dimensional sphere function. For this test case the tested MAES variants clearly outperform tested conventional ES variants. This result is significant, since the 80th percentile of the MAES runs is still clearly better than the median of the conventional strategies. As a rule of thumb it can be stated that the MAES needs about half the time than the best strategy among the conventional ES to achieve the same precision.

An unexpected observation has been that the strategies that use the confidence information, namely the lower confidence bound, the PoI and the ExI MAES, are not slower than the mean value strategy. This can be attributed to the fact that the models are very precise for the sphere problem and thus there is no significant difference in the predictions

due to the mean value and the MAES versions that consider also the imprecision of the metamodel.

The acceleration of the ES continues to function in different phases of the optimum approximation that are characterized by different orders of magnitude for the precision of the optimum approximation. This is an indicator for the functioning of the step-size adaptation and also for the functioning of the online learning of the metamodel. In other words, the precision of the local metamodel adapts to the precision of the search. This result will be observed on many other test cases. A detailed study of this scale invariance will be provided in section 4.5.6, where the long term behavior of the MAES is discussed.

Note, that the operators of the MAES work with a maximal precision of 10^{-6}, in order to make sure that the matrix operations for computing the metamodel predictions work well. Working with higher precision arithmetics would clearly slow down the optimization process. For more complex functions than the five dimensional sphere problem this precision limit has almost no influence on the behavior of the strategies.

So far we discussed only results for the five dimensional case, where function approximation is easy. The second and the third row of the table depict results for 10 and 20 dimensions, respectively. In 10 dimensions the MAES still outperforms the classical ES versions. It can be observed that the convergence speed $(1 + 10)$-ES is now much closer to that of the MAES. However, the $(1 + 10)$-ES is not very robust, as the plot of the 80th percentile reveals. Still the MAES is about a factor two faster than the conventional $(5 + 20)$-ES - the strategy with the same population size parameters.

In addition, it can be observed that the MAES versions that use the confidence information of the metamodel converge slightly slower than the MAES with mean value filter. This effect gets more visible in 20 dimensions (third row of table 4.5.2). It can be attributed to the fact that the model quality gets worse in higher dimensions and thus the MAES concentrate more on exploration than on exploitation, the effect of which is a slowed down convergence.

Also in higher dimensions the acceleration achieved with the MAES is significant. One might intervene, that the $(1 + 10)$-ES converges with the same speed than the MAES in the 20 dimensional case. Without providing an extra table, it shall be noted here that MAES versions that work with an increased generation number converge much faster on the sphere model. However, these strategies as well as the $(1 + 10)$-ES show a lack of robustness on more difficult test functions. Empirical evidence for this will be given later in this section, when we discuss results on multimodal test functions.

Results on further unimodal test functions The sphere model is a very simple test function, even among unimodal functions. It is symmetrical, decomposable and differentiable. Further unimodal test problems have been investigated on, in order to gain confidence in the observations on convergence behavior of the MAES. Table 4.5.3 sums up these results.

The first row of the table 4.5.3 depicts the convergence behavior of different strategies on the ellipsoid model. Again, a clear superiority of the metamodel-assisted strategies can be observed. Still the MAES variants converge almost twice as fast as the equivalent standard ES, i.e. the $(5 + 20)$-ES. In addition, it converges faster than the $(1 + 10)$-ES.

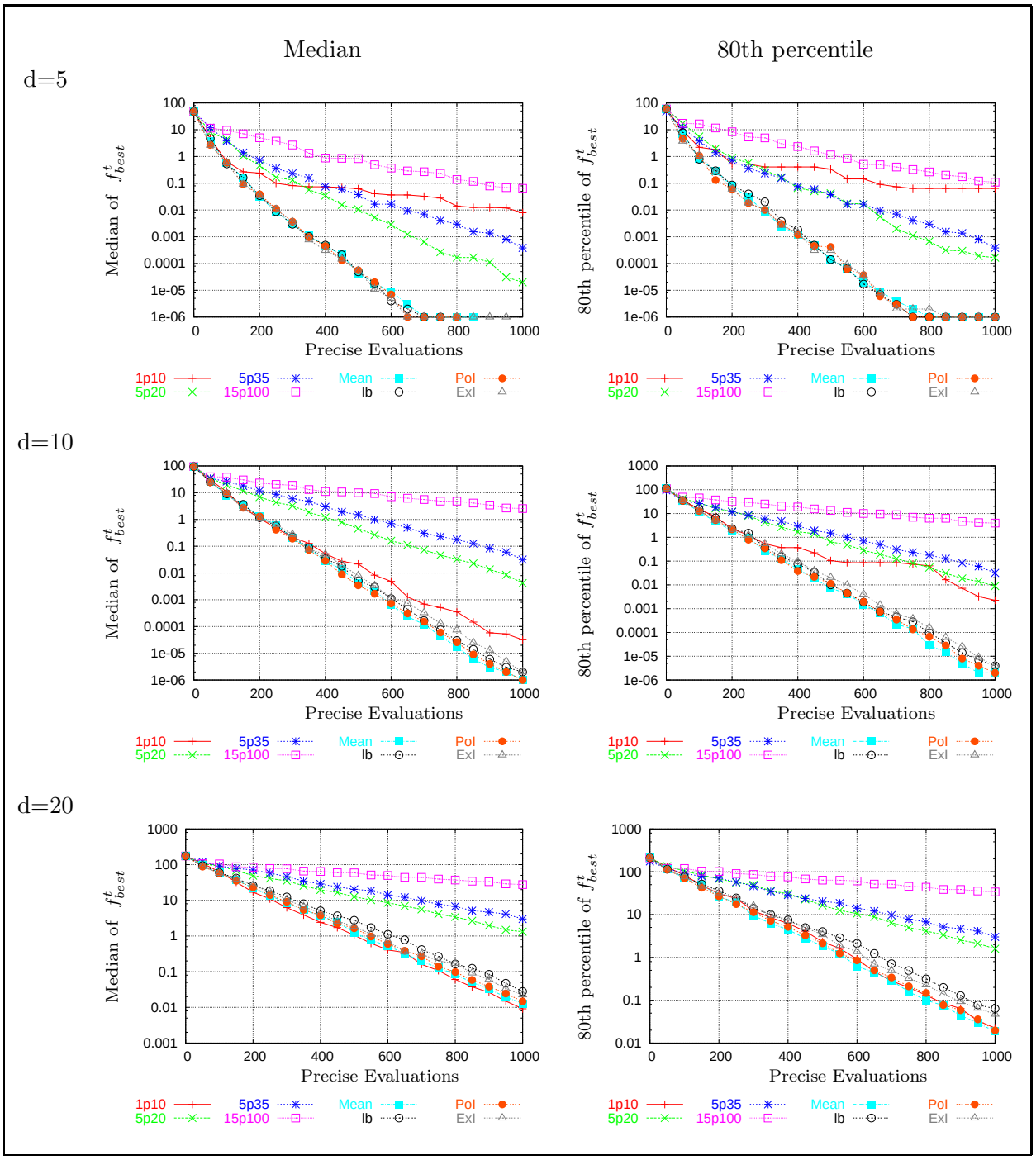


Table 4.5.2: Results for the sphere model in different dimensions. The descriptions of the studied algorithm variants can be found in table 4.5.1.

This indicates, that the MAES is capable of dealing with problems, where the variables have a different impact on the function value.

Another problem, that has been investigated on, was the double sum problem, the results for which are displayed in the second row of the table 4.5.3. For this test problem, variables do not only have different effects on the function value but they are also correlated. Note, that correlated variables are not reflected by the choice of the correlation function for the MAES. On this problem the mean value MAES and lb_ω MAES have

almost the same convergence speed than the equivalent $(5 + 20)$-ES. However, the PoI and ExI criterion perform better than their conventional counterpart.

Certainly, the increased robustness, due to the confidence information used, would be suspected as an explanation for the increased performance. But, then the performance of the lb_ω criterion should also have an increased performance in comparison to the mean value MAES, and this is definitely not the case. Hence, the use of confidence information alone cannot serve as an explanation for the superior behavior of the PoI and ExI MAES.

In order to explain this effect, let us recall a result stated earlier (section 4.2.2: For the PoI and for the ExI the relative position of the reference value f^t_{best} to the predicted value $\hat{y}$ is of crucial importance for determining, whether the standard deviation of the prediction $\hat{s}$ has an positive effect on the ranking of an solution or not. Unlike the PoI and ExI criterion the lb_ω criterion always rewards solutions with high value of $\hat{s}$. In particular, the PoI and ExI criterion reward solutions that have an decreased value for $\hat{s}$ if $\hat{y}$ is smaller than the best found solution so far, thus rewarding solutions that are improvements with a high certainty to solutions which are improvements with a lower certainty. This is not the case for the lower confidence bound criterion, which always rewards solutions with high certainty, provided their predicted value is equal.

The experimental data of the double sum problem confirms that the metamodel tends to underestimate the true function value. Now, due to their characteristics, the PoI and ExI are able to distinguish between underestimated values and values that are better than the currently best function value with a high certainty. This leads to a more focussed search and to the significantly better results than that obtained with the strategy using the LCB filter, which focusses on underestimated solutions, and the mean value criterion which does not care at all for the certainty of an prediction, thus having a convergence behavior that lies in between that for the PoI and ExI on the one hand and the lower confidence bound criterion on the other hand.

Last but not least, the behavior on a discontinuous function with plateaus should be discussed. The results on the 10-D step function, that reflects these characteristics, are depicted in the third row of table 4.5.3.

Due to their model assumptions, gaussian random field do not support the modeling of discontinuous functions. However, they can still provide valuable predictions, if there is some causal structure in the modeled landscape. In particular, this holds, if the 'jumps' in the function values are small compared to the range of function values in the sample. For the step function, this means, that the metamodel can predict with a small relative error if the sample is widely distributed in the search space and thus the relative change of function values is similar to that of the sphere function. As the search approaches the optimum, an increasingly poor quality of the metamodel is to be expected. However, the standard deviation $\hat{s}$ can still serve as an relative indicator for the unexploredness of an region.

The results obtained with the MAES are consistent with the aforementioned assumptions. Indeed, the MAES finds rapidly a near optimal region and then tends to stagnate. The stagnation time is lower for strategies that make use of confidence information and thus have the tendency to strive to the unexplored regions of the search space. The latter characteristic also helps to navigate on plateaus, where a random walk tends to re-sample

the same region again and again.

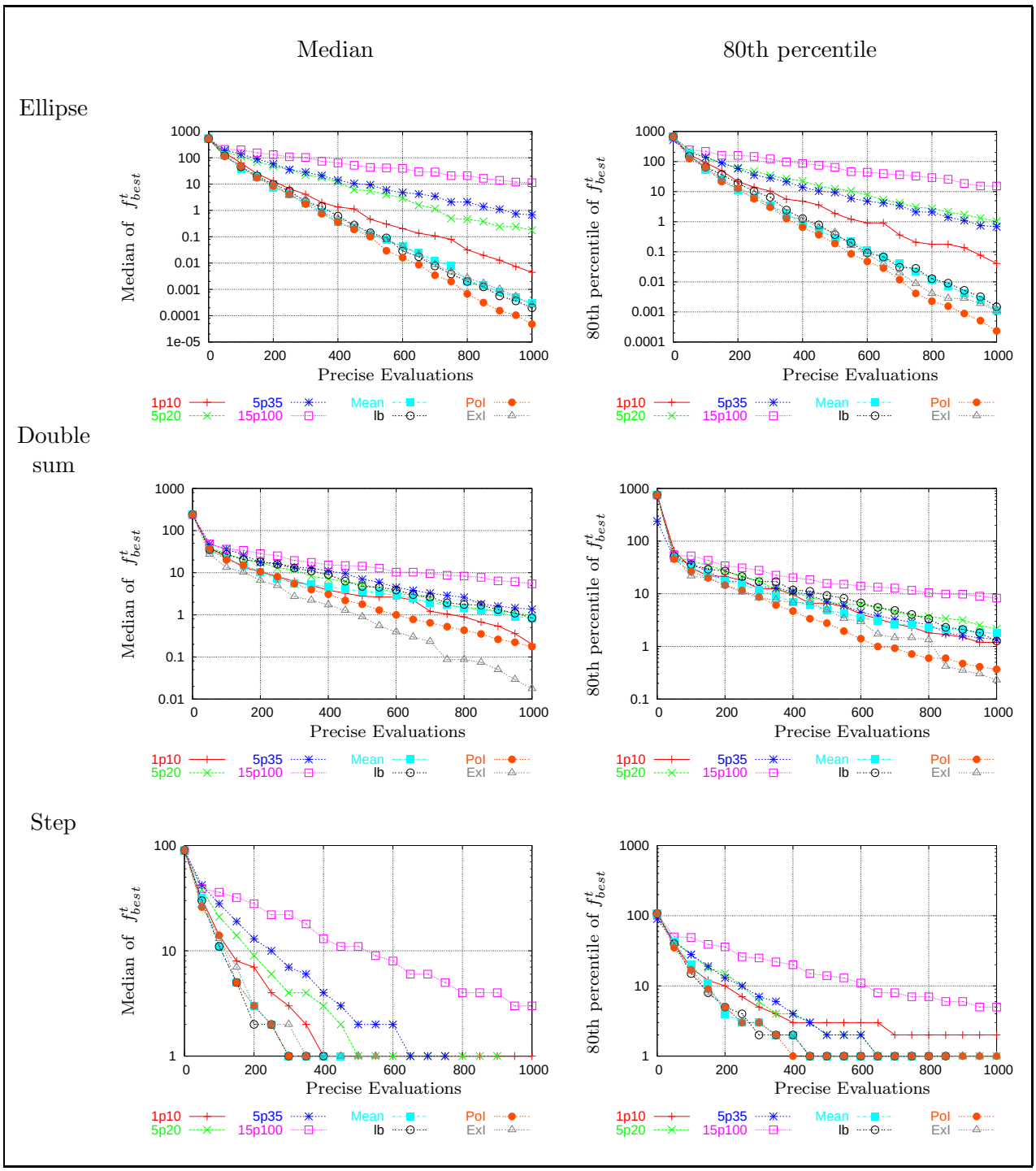


Table 4.5.3: Results for unimodal functions using filters with constant output size. The descriptions of the studied algorithm variants can be found in table 4.5.1.

Results on multimodal functions Evolutionary algorithms are often applied for multimodal optimization. Hence, we included some multimodal problems into our benchmark. These are the Ackley problem (10-D and 20-D), the Rastrigin problem (10-D), and the Fletcher Powell problem (10-D). Table 4.5.4 shows the results on these functions. For some parameterizations of the MAES, plots in table 4.5.4 are accompanied by detailed

plots of all test runs (table 4.5.5 for the Ackley problem and table 4.5.6 for the Fletcher Powell problem).

A first test problem for multimodal optimization has been the Ackley problem (section 4.5.1), the results on which can be obtained from the first ($d = 10$) and second row ($d = 20$) in table 4.5.4. In contrast to the Rastrigin problem, the global trend of the Ackley function is very flat in the regions far away from the optimum (cf. figure 4.5.9). Therefore, the ability of the strategy to 'jump across' local optima is of crucial importance in the beginning of the search.

Conventional ES with a high selection pressure seem to have more problems, than those with low selection pressure. In particular, the $(1 + 10)$-ES gets clearly outperformed by all other ES with $\mu > 1$. Again, the MAES versions perform significantly better then the conventional ES. Unlike the mean value MAES, all MAES versions using the confidence information, namely the lb_ω MAES, the PoI MAES and the ExI MAES, found the attractor region of the global optimum in more than 50% of the test runs. Only the MAES using the lb_ω-filter obtained the attractor region of the global optimum in more than 80% of the test runs. In 20 dimensions (third row of table 4.5.4), the advantage of strategies using confidence information gets even more visible.

A detailed comparison of twenty runs for the mean value MAES and the lb_ω MAES on the 10-D Ackley problem is given in table 4.5.5. Here, convergence to local sub-optima, which goes along with a reduction of the sampling radius, can be observed many times. It is notable, that the convergence to local sub-optima also happened for the lb_ω MAES, though it occurred only in two out of twenty runs. Once the sampling radius has been reduced, there is almost no more chance to escape from a local optimum. This effect cannot be attributed to the failure of the lb_ω criterion to detect unexplored regions in the search space. Rather, it points to the possible shortcoming of the MAES, that it might not sample points in unexplored regions if the step-size is too low. Thus these points cannot be preselected, even though they might have good values for the lb_ω. A possible measure to counteract this problem, would be to keep the sampling radius high. However, such strategies have not been tested in this work but their development bears potential for future improvements.

Further support for the hypothesis that the use of filters that consider $\hat{s}$ helps to improve results on multimodal problems can be obtained from the experimental study on the Rastrigin problem (cf. 4.5.1), the results of which are displayed in the third row of table 4.5.4. As discussed previously (section 4.5.1), from a distance the Rastrigin function looks like a quadratic unimodal function. The superposed sinus and cosine terms have a constant, high frequency and are relatively low in amplitude.

The results indicate, that it is difficult to find attractor basin of the global optimum within 1000 evaluations, even for the MAES. However, we can compare the relative performance of the different ES and MAES on this problem. Among the conventional strategies tested, the ES instantiations with $\mu > 1$ performed better. Furthermore, a low selection pressure seems to increase the average performance, as the comparison between the $(5 + 20)$-ES and $(5 + 35)$-ES reveals. It is remarkable, that all MAES versions perform better than the conventional ES versions on this problem. In particular, the $(1 + 10)$-ES is no longer competitive, as it was the case for one of the unimodal functions. Despite, it tends to converge quickly to local optima. Among the MAES versions, the strategies using the

lb_ω-filter and the PoI filter perform best, which can be attributed to the fact, that they reward solutions, that have relatively low predicted value $\hat{y}$ accompanied with a high standard deviation $\hat{s}$. This entails an increased tendency to escape from local optima.

Finally, a study that points to the limits of the applicability of the MAES, is discussed. The Fletcher Powell function is the most difficult problem among the multimodal test problems in the comparison. In contrast to the Rastrigin and Ackley problem, the optima are irregularly spaced and have attractor basins of different size. The fourth row of table 4.5.4 displays averaged test runs for this problem. The results are supported by table 4.5.6 that displays results in a more detailed resolution. With the exception of the (15+100)-ES, all strategies stagnated after about 500 evaluations. Thus, the initial phase of the optimization decided, to which optima the strategies converged. An advantage can be made out for the PoI and mean value strategy. However, it seems difficult to explain the superior behavior of these strategies, as from the detailed plots in table 4.5.6 we obtain that their behavior is highly unpredictable. It seems that for this problem class other settings of the MAES parameters are to be chosen, if we want to obtain a good result in the long run. Results for the LBI_ω MAES, which work with a much lower selection pressure, point in that direction (see lower right corner of table 4.5.6). Later, in the discussion of adaptive filters, we will come back to this point.

In summary, it was observed that the MAES outperforms standard ES on multimodal problems. Filters using the confidence information typically lead to an increased convergence reliability. This holds, in particular for relatively simple multimodal problems. For more complex multimodal problems, like the Fletcher Powell problem, the behavior of the MAES with constant output-size filters becomes highly unpredictable for the given number of 1000 evaluations, though for some MAES variants still an accelerated convergence was encountered.

Filters with variable output size

Among the filters with variable output size we distinguished between filters with a high recall – i.e. the LBI_ω-filter and the R_ω-filter – and filters aiming at a high precision – i.e. the MLI-filter and the P_ω-filter . The results for these types of filter strategies shall be discussed next.

Table 4.5.7 displays results obtained with MAES using filters with adaptive output size on unimodal functions, and in table 4.5.8 results for multimodal problems, respectively. In both tables, they are compared to the aforementioned $(5 + 20 < 100)$-MAES variants, all of which used filters with constant output size.

On quadratic problems MAES that used high precision filters performed by far better than all other MAES strategies tried so far. For example the MLI filter found the optimal solution on the 10-D sphere model in about 18 % of the time of the corresponding (5+20 < 100)-MAES. The P_ω-filter was even slightly faster.

Clearly, a lack of robustness is the price that has to be paid for the improved performance on smooth unimodal functions. This already comes to show, when looking at the results on the step function (fourth row of table 4.5.7), where both, the MLI-MAES and the P_ω-filter MAES failed to find the global optimum. Similar results were found for multi-

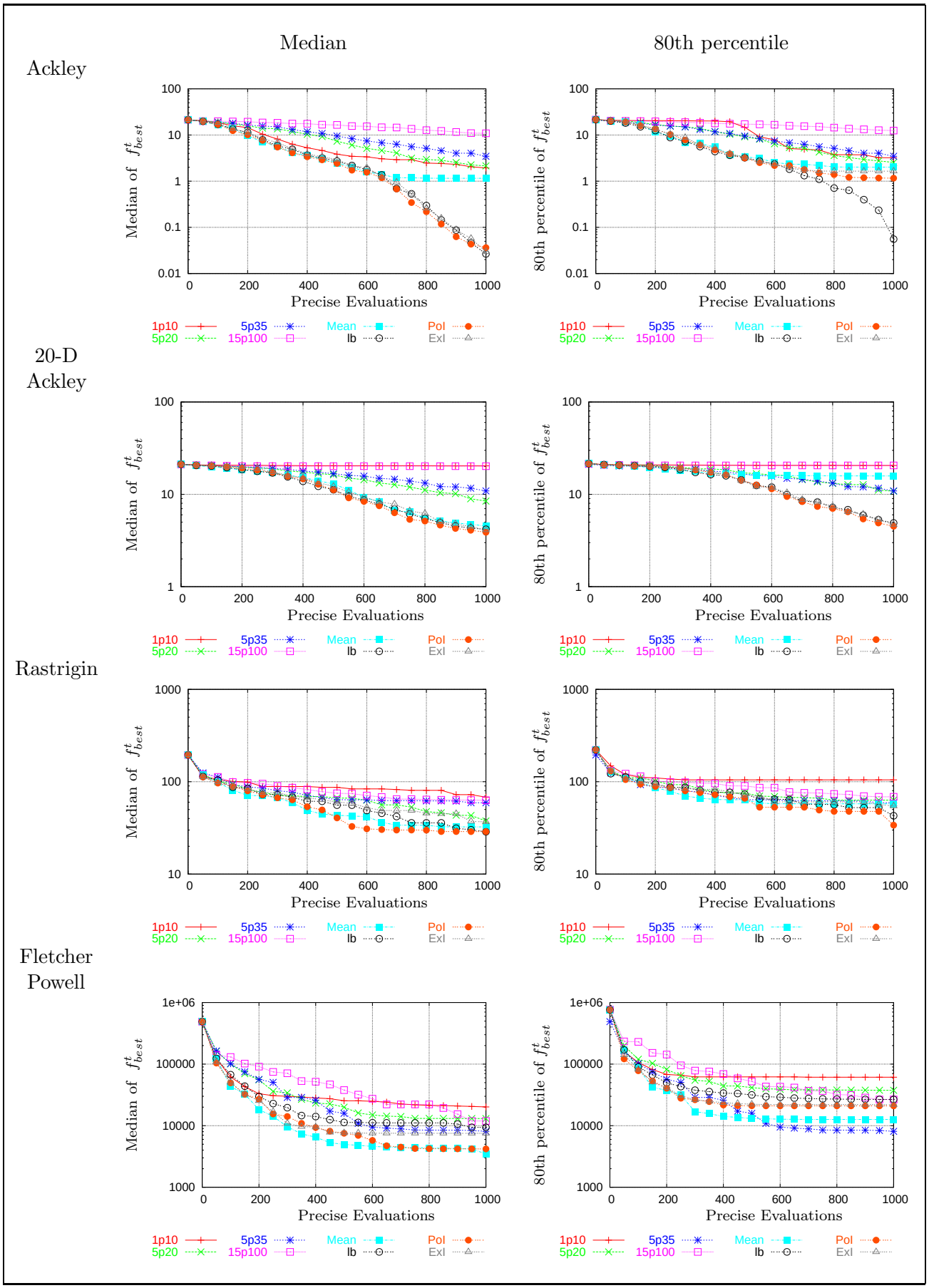
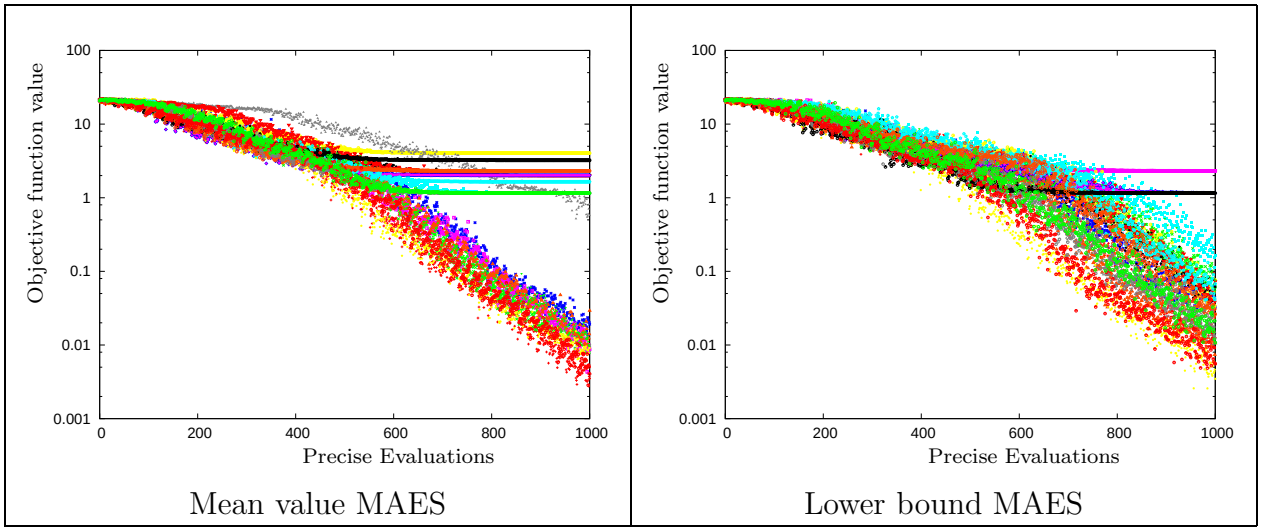
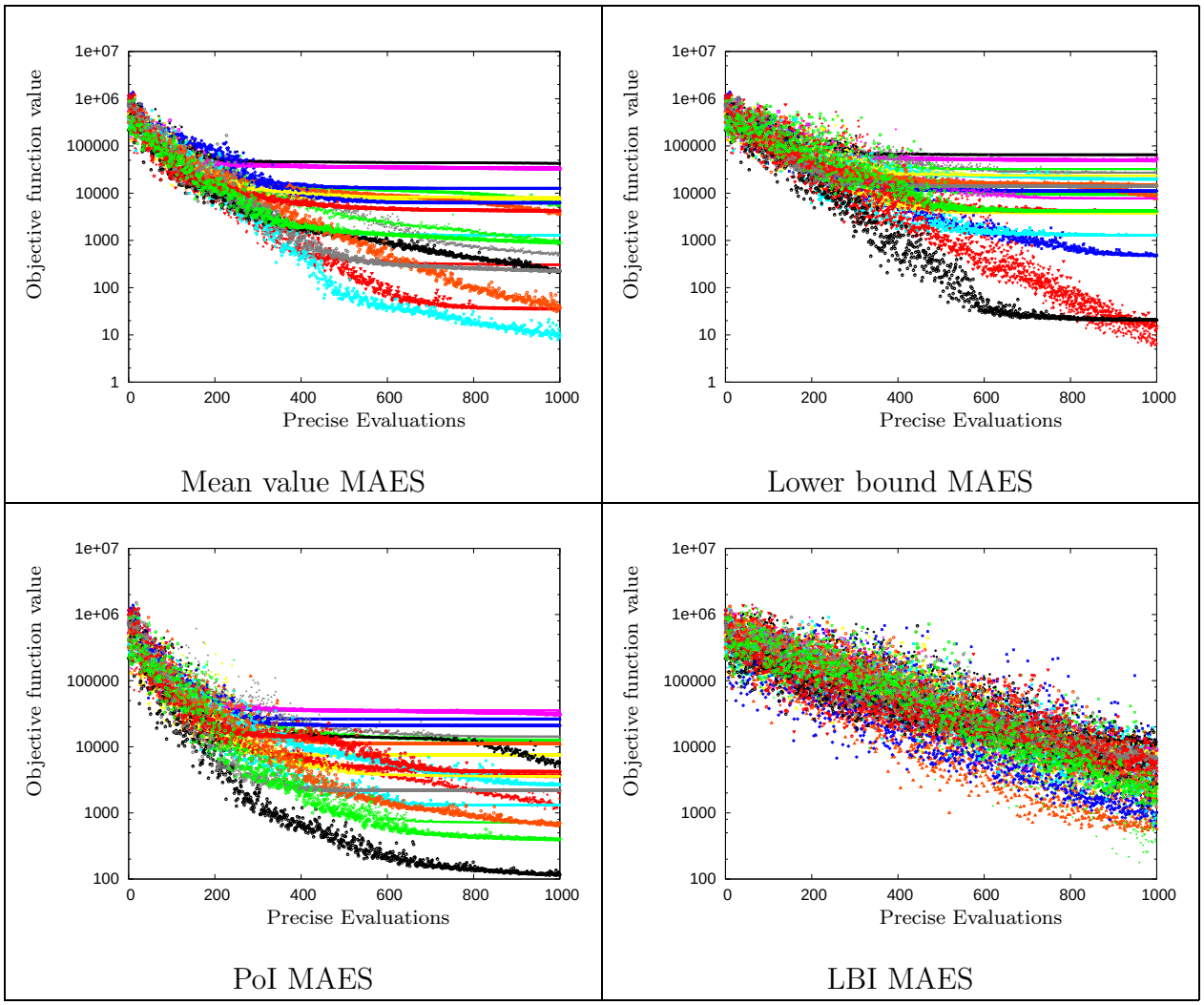


Table 4.5.4: Results of the $(5+20<100)$-MAES on different multimodal functions. The description of the studied algorithm variants can be found in table 4.5.1.



Mean value MAES | Lower bound MAES

Table 4.5.5: Detailed plot of the function values measured during 20 runs of different MAES versions on the Ackley problem.



Mean value MAES | Lower bound MAES

PoI MAES | LBI MAES

Table 4.5.6: Detailed plot of the function values measured during 20 runs of different MAES versions on the Fletcher Powell problem.

objective functions (table 4.5.8).

A contrary behavior was observed for MAES variants, using filters that aim at a high recall, namely the LBI_ω-MAES and the R_ω-filter -MAES. These MAES showed a relatively good performance on multimodal problems, but converged extremely slow on smooth local optimization problems. In particular, this holds for the LBI_ω-MAES. The MAES versions that worked with filters of constant output size behave in a way that lies between these extremes.

An explanation for the complementary behavior of the high precision and high recall filter on local optimization problems is probably that the high recall filters let a significantly larger number of individuals pass in each generation than the high precision filters. As a consequence, the number of generations for the high recall filters was much lower than that for the high precision filters. This explanation is supported by the Box-plot in figure 4.5.12, where the number of generations are compared for different MAES. The R_ω-filter works with a low number of only 20 generations, indicating that almost all individuals generated during the test runs have passed the filter, while the MLI and P_ω-filter achieve a high number of generation, though not each of the individuals was rejected by these filters. Note, that the $(5 + 20 < 100)$-MAES computed a total of 50 generations, which is similar to the average number of generations performed with the LBI_ω MAES.

The value of $\omega = 1$ has been chosen for the LBI_ω filter in contrast to $\omega = 2$ the R_ω-filter (see table 4.5.1). Due to lemma 5 in section 4.2.4 an decreased value of ω leads to a lower permeability of the R_ω-filter and LBI_ω filter. Hence, by choosing a lower value of ω, the number of generations can be increased. The empirical results indicate, that ω has to be chosen sufficiently much lower than the default value of $\omega = 2$, in order to avoid that (almost) all individuals pass the filter.

A remarkable result has been achieved on the multimodal Fletcher Powell function. Here the MAES using the LBI_ω filter performed superior (third row of table 4.5.8) to all other tested MAES variants. A detailed plot of the results on that function is displayed in table 4.5.6 (lower right). The LBI_ω MAES was the only strategy that did not have the tendency to converge to local sub-optima. The only drawback of the LBI_ω MAES seemed to be, that it did not allow for a fine tuning of the result in the end. Hence, for practical applications, it is recommended to further improve the final result obtained with the LBI_ω MAES by means of a local optimization strategy.

Summing up, we may learn two lessons from the results obtained with filters of variable output size. Firstly, it has been obtained that high precision filters converge very fast to local optima but fail to work on discontinuous and multimodal problems, while high recall filters are much more robust on multimodal and discontinuous problems but lack precision in the final optimum approximation. Secondly, we observed, that a possible danger when using MAES with high recall filters is, that they might let pass almost all individuals. Hence, the choice of ω can be crucial.

However, this entails that the user has to choose a parameter again and we loose one of the alleged advantages of the MAES with variable output size, since we claimed that we can get rid of one of the parameters. This also holds for threshold versions of the PoI filter, namely the PoI_τ-filter, since they are equivalent to the LBI_ω filter with $\tau = \Phi(-\omega)$. Still, the MAES with adaptive output size filters have the advantage that they can react

more flexible on the measured function topology during the run, by increasing the output size in case of rugged landscapes and decreasing it in case of smooth landscapes.

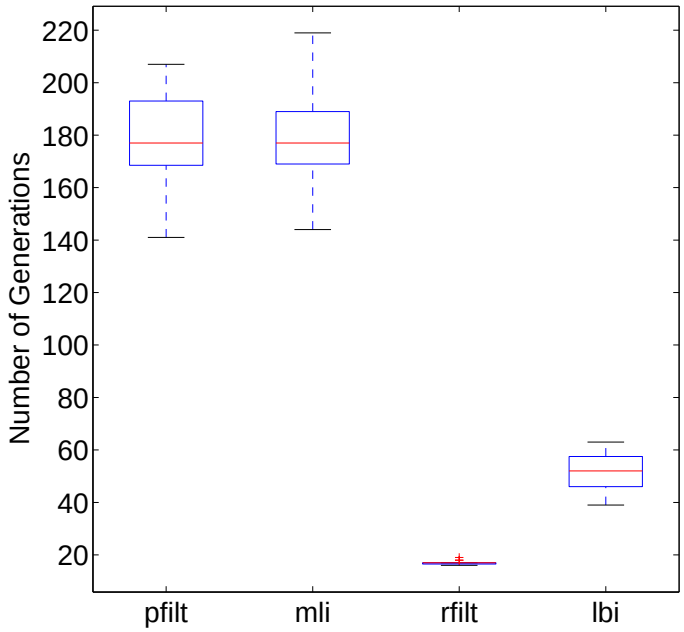


Figure 4.5.12: Box-plots for the number of generations performed by variable output filters on sphere model for a limited number of 1000 objective function evaluations (pfilt = P_ω-filter, mli = MLI-filter, rfilt = R_ω-filter, lbi = LBI_ω-filter).

4.5.4 Performance of the metamodel

It might be alleged, that the metamodel assistance does not support the search in the way we explained it, but some other obscure effect is responsible for the increased performance of the strategies. In order to meet this objection several indicator plots have been studied, that shall provide us with more details about what is going on during the run of a MAES. The aim was mainly to verify that the metamodels and the filters based on them behave in a way that is consistent with their theory. Furthermore, the results shall lead to a deeper understanding of the behavior of the different MAES.

First, we studied the numerical quality of the metamodel predictions for runs on the sphere model in different search space dimensions by means of $y \sim \hat{y}$-plots (figure 4.5.13 and 4.5.14). The results are displayed in a twice logarithmic $y \sim \hat{y}$-plot in order to screen the prediction quality in different orders of magnitude.

All points were near the intersection line, meaning that the correspondence between the metamodel predictions and the true objective function value was very high. The relative deviation from the true value is almost constant for different orders of magnitude of the function value. It has also been observed, that in higher dimensions the quality of the metamodel hardly decreases (4.5.14). The performance loss due to the dimension is less significant than one might have been expected on basis of the 'curse of dimension'. This

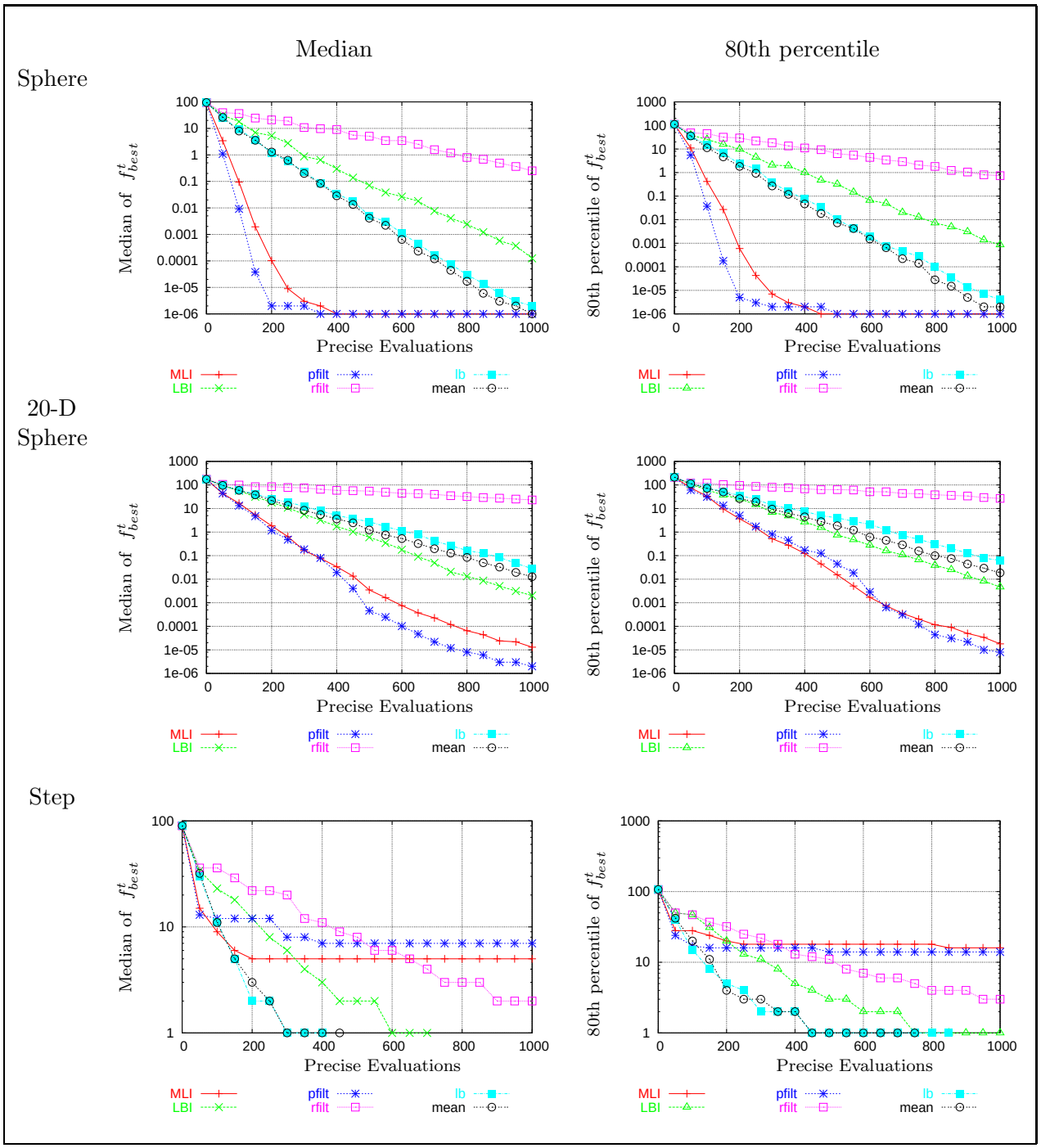


Table 4.5.7: Results of MAES using filters with variable output size on unimodal functions. The descriptions of the studied algorithm variants can be found in table 4.5.1.

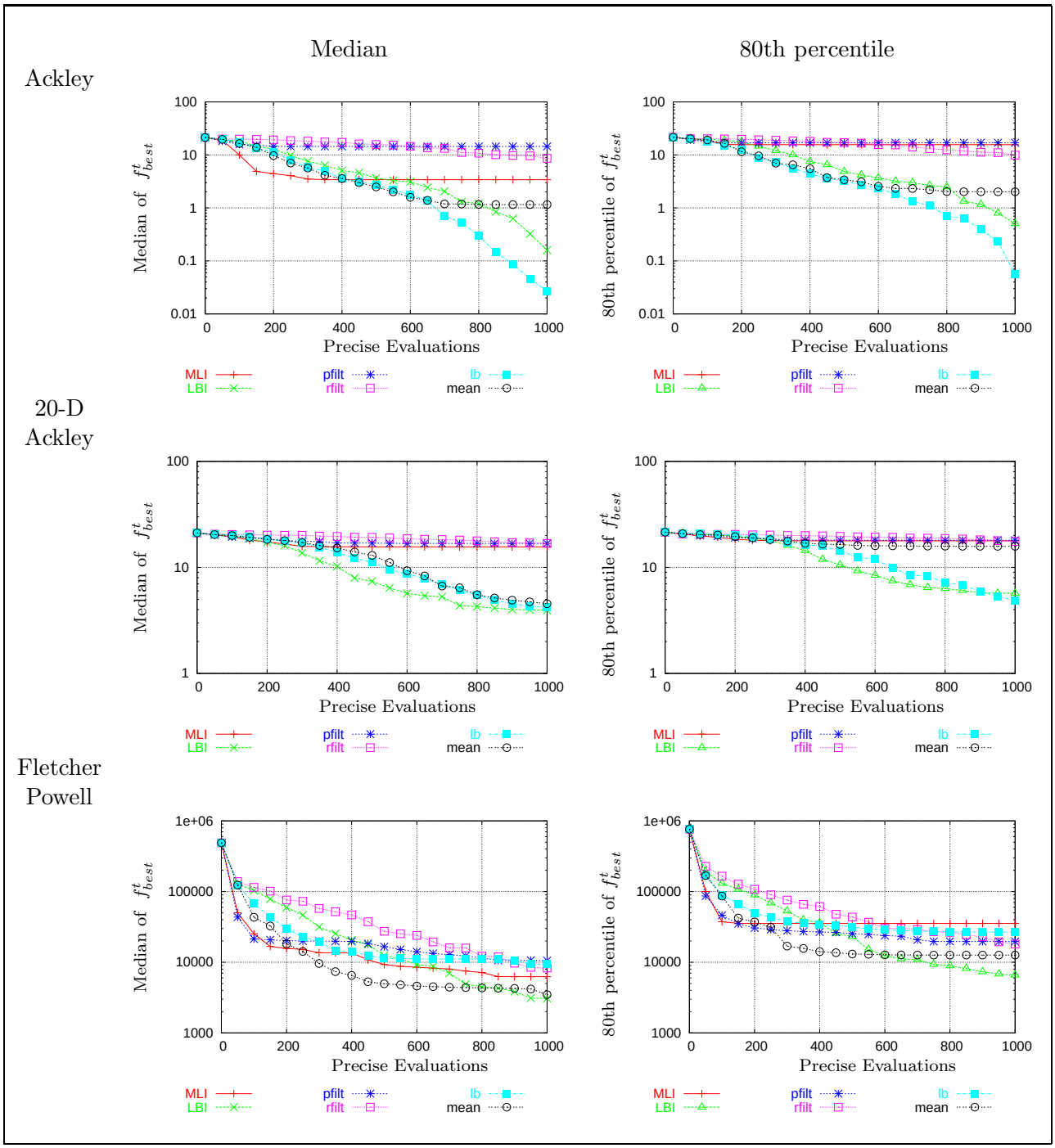


Table 4.5.8: Results for MAES using filters with variable output size on multimodal test problems. The descriptions of the studied algorithm variants can be found in table 4.5.1.

result can be explained by the decreased convergence speed, that entails an increased dwelling time of the MAES on the 20-D problem in different distance ranges to the optimum and thus an more extensive sampling.

Additionally, the accuracy of the predictions on the multi-modal 20-D Ackley problem was investigated (figure 4.5.15 and 4.5.16). Also on this problem we observed a strong correlation between the predicted and true function values, thought the relative error considerably higher on these problems. From the $y \sim y_{lb}$ plot we conclude that this is also reflected by an increase width of the estimated confidence interval.

On basis of the $y \sim \hat{y}$-plot exclusively it is difficult to judge whether the metamodel quality suffices to find the a precise sorting of the offspring solutions or not. Thus we also provide indicator plots that display the number of inversions (cf. section 4.4.3) for different stages of the optimization and two representative problems (figure 4.5.17).

For chosen test problems, namely the 20 dimensional sphere problem and the 10 dimensional Ackley problem, the number of inversions is significantly below the number of inversions that is expected for a random sorting of the 100 offspring individuals. The results on the other test problems look similar and have thus been omitted in this thesis. Recall, that the expected number of inversions $\mathrm{E}(\xi_{100})$ and its standard deviation $\sigma(\xi_{100})$ have been derived in the example of section 4.4.3. In the figures, dashed lines indicate this mean value and the lower confidence bound (= mean value minus standard deviation) for random sorting. The number of inversions fluctuates and decreases slightly towards the end, which can be explained by the enrichment of the database with sampled points. In higher dimensions the number of inversions grows on average, but stays below the critical number of $\mathrm{E}(\xi_{100}) - \sqrt{\mathrm{Var}(\xi_{100})}$.

For the other filters with constant output size, the results are worse than that for the mean value filter. This is consistent with the theory of these filters, because they do not sort the population by means of the predicted value but by a value that also depends on the standard deviation of the prediction. This more or less biases the ordering achieved for the offspring population. With exception of the sphere problem, a low number of inversions does not directly correspond to a better convergence behavior.

Due to these results, using the number of inversions as a performance measure, or even as a measure to adapt strategy parameters during the run, would be misleading. The results on the Ackley function provide a good example for the problem. Here, in the stage of convergence to local sub-optima, the number of inversions is very low, because after a while the sub-optimal region is well exploited. Hence, the number of inversion indicator would suggest that the strategy is doing well, not recognizing that it is trapped in a nicely modeled local sub-optimum.

It might be claimed, that the main purpose of including the variance into a pre-selection criterion, is to boost sampling in regions of poor model quality in order to obtain a better metamodel. The results presented in table 4.5.17 do not support this argument. Rather, the high performance of the lb_ω filter on the Ackley function, supports the thesis that the 'escape' to unknown regions (white spots) of the search space has to be supported, which helps to prevent stagnation of the search.

Additionally, *recall* and *precision*, as introduced in section 4.4, have been measured for all employed MAES variants over time. Tables 4.5.9 and 4.5.10 display the results for

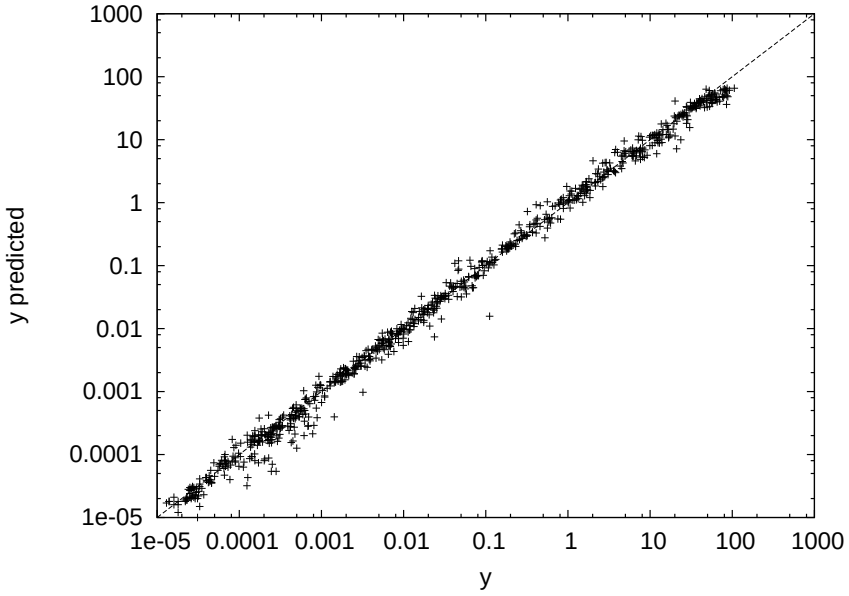


Figure 4.5.13: $y \sim \hat{y}$-plot for all predictions made during the run of the (5+20<100) mean value MAES on the 10-D sphere model.

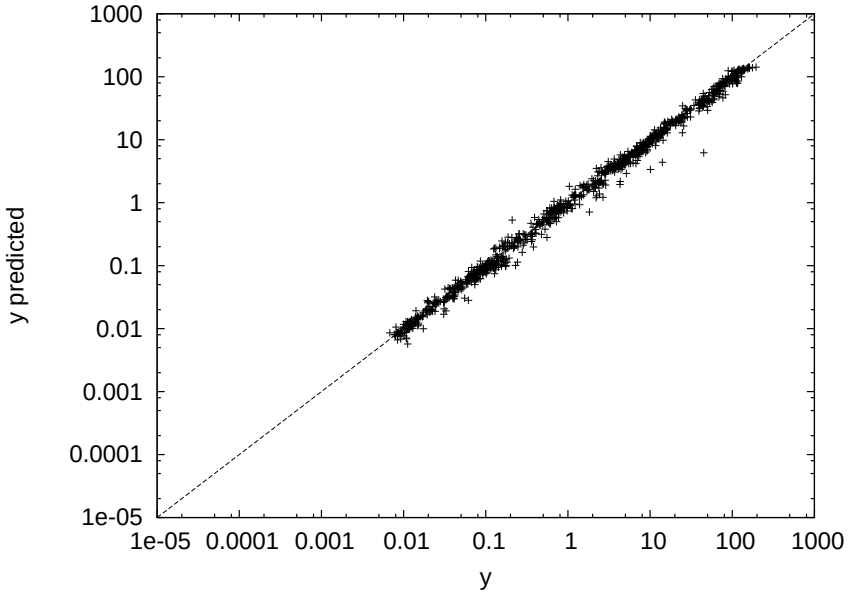


Figure 4.5.14: $y \sim \hat{y}$-plot for all predictions made during the run of the (5+20<100) mean value MAES on the 20-D sphere model.

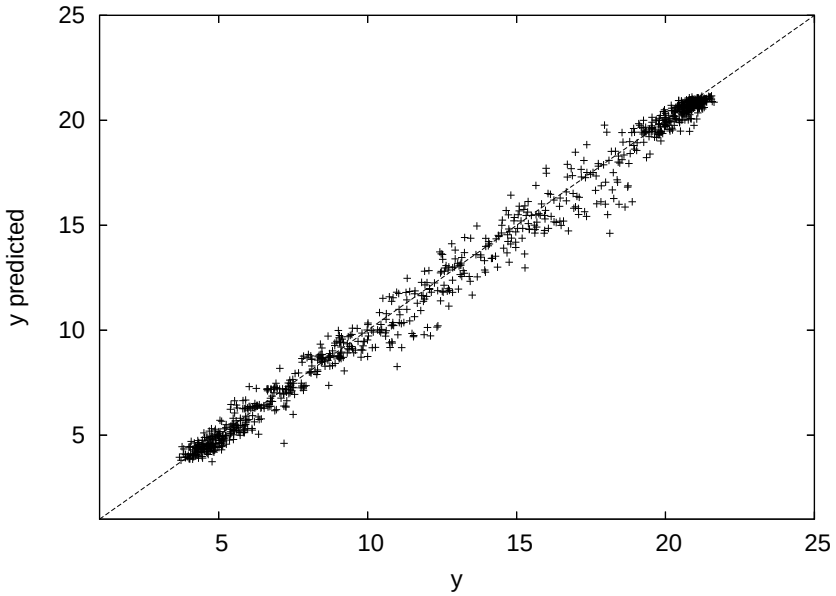


Figure 4.5.15: $y - y$ plot for a run of the MAES on the 20-dim. Ackley function.

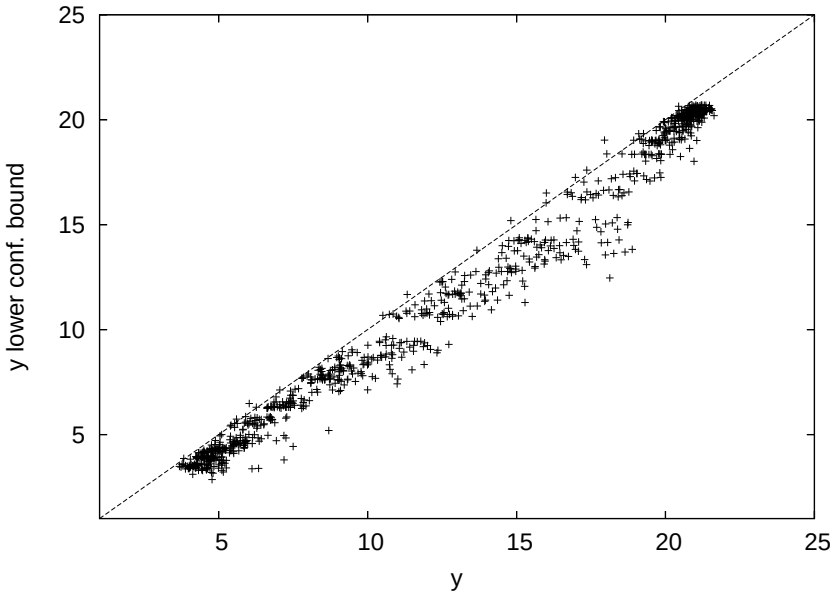


Figure 4.5.16: $y - y_{lb}$ plot for a run of the MAES on the 20-dim. Ackley function.

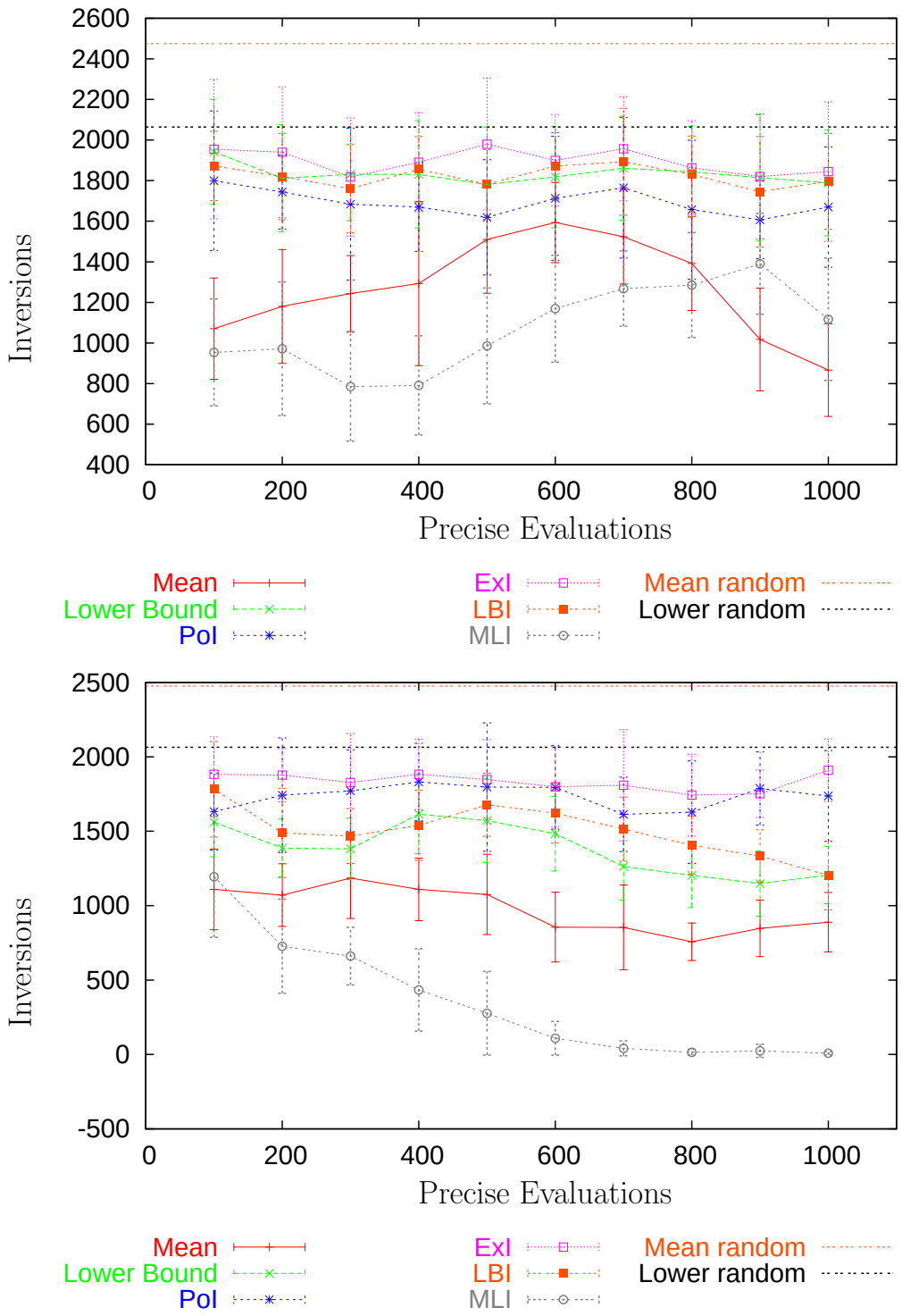


Figure 4.5.17: Average results for the number of inversions for two different test problems, the 20-D sphere problem (upper) and the Ackley problem (lower). For each combination of algorithm and test problem 20 runs were performed. The descriptions of the studied algorithm variants can be found in table 4.5.1.

filters with constant output size as well as for adaptive filters.

Clearly, the results reflect the main characteristics of the filters that were expected from theory. This is, that the MLI and P_ω-filter both work at a high precision and a low recall and that the LBI filter and R_ω-filter work at a low precision but high recall.

For the filters with constant output size (table 4.5.9) the precision stays below a value of 0.25, which is consistent with the fact that only $\mu = 5$ out of $\nu = 20$ pre-selected individuals are finally selected and thus it is impossible to obtain higher precision values than $5/20 = 0.25$. In addition the PoI and mean value filter have a slight tendency to work at a higher precision.

The highest recall for constant output size filters were achieved for the mean value criterion, where more than 80% of the relevant individuals are among the preselected 20 individuals. The only exception to this rule was the result obtained on the double sum function. As it has already been said, the high correlation between variables severely violates the model assumptions of decomposable correlation functions. However, it is interesting to see that the bad model quality for this function does not lead to a complete failure of the MAES. This can be attributed to the fact, that a wrong sorting of the offspring population does not completely deteriorate the search process. Rather, the strategy falls back to the standard $(5 + 20)$-ES, in that case.

The only case, when the metamodel-assistance could indeed be harmful would be, if it produces a deceptive ordering, which - in case of the $(5+5 < 100)$ -ES would be indicated by recall values below $20/100 = 0.2$. Strangely enough, this is exactly the case for the expected improvement criterion on the double sum, which - besides the PoI criterion - performed best on the double sum problem. A possible explanation for this behavior is, that this filter either selects solutions that are almost for sure among the 5 best solutions, or it selects solutions that are placed far away from good solutions, and the latter case appears more often, since regions far away from a local optimum are sampled more often. The results in the precision column for the double sum function in table 4.5.10 support this conjecture.

Another reason for the good results on local optimization problems is certainly the high number of generations (figure 4.5.12). It is surprising that the low selection pressure with regard to the ratio between the number of pre-selected individuals and parent individuals, that is often smaller than one, does not lead to a failure of the step-size adaptation. Clearly this indicates, that the number of generations is an important factor for scaling between more local and more global search. If we compare the results of the MLI-MAES with that of the lower confidence bound and mean value MAES, it can also be concluded, that the choice of filter criterion for a filter with constant output size is far less important for the behavior of the strategy than the choice of the number of pre-selected individuals that determines the number of generations performed.

The price that has to be paid for the extremely high acceleration on local optimization problems is a lack of robustness on discontinuous and multimodal problems. The results on the step problem as well as the results on the Ackley and Fletcher Powell function (table 4.5.8) clearly indicate this.

Again, the experimental results demonstrate the similarity between the behavior of the LBI and R_ω-filter -MAES on the one hand, and the MLI and P_ω-filter -MAES on the

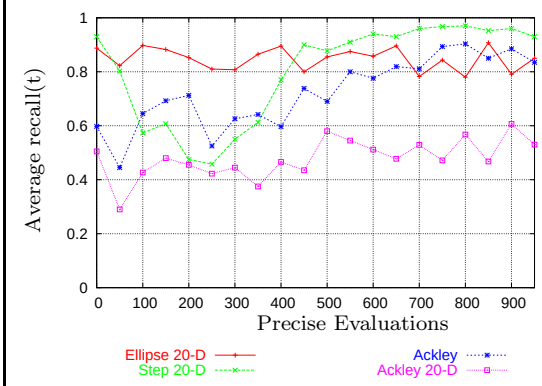
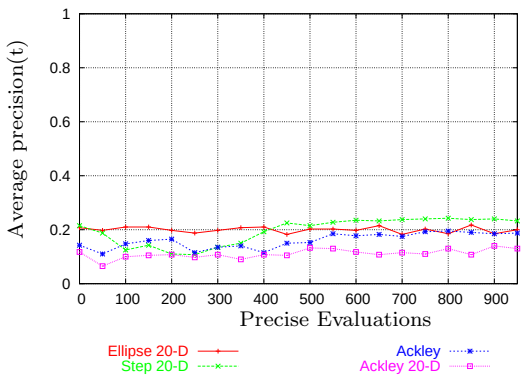
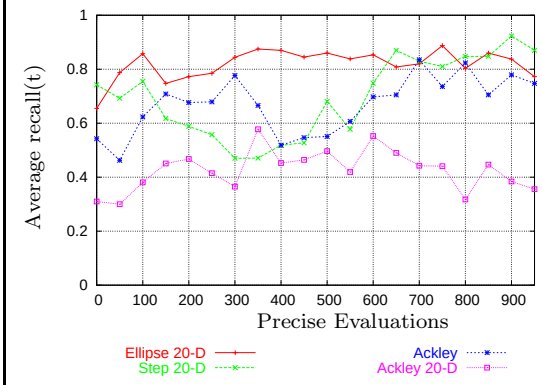
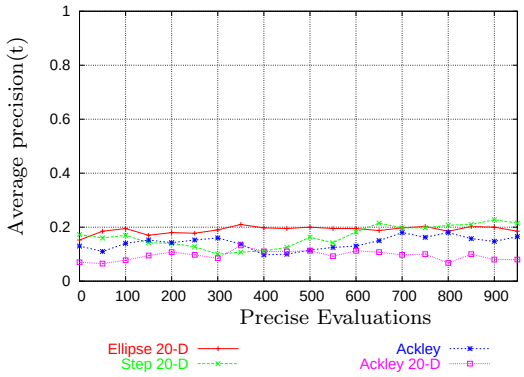
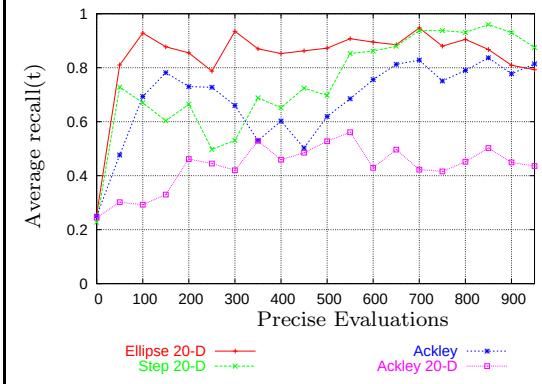
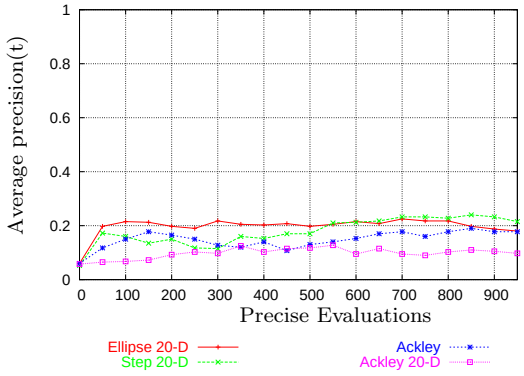
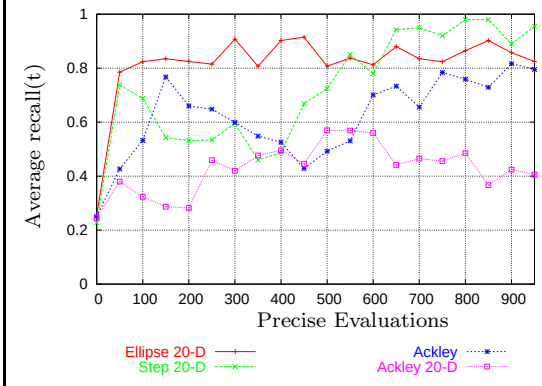
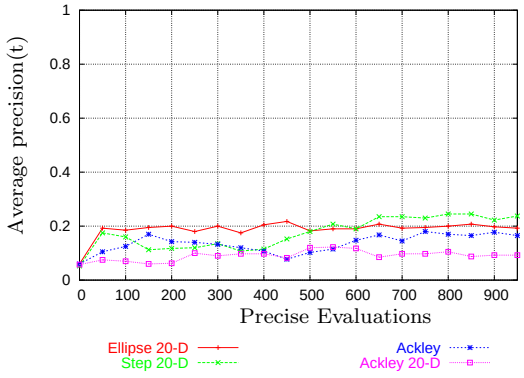
	Recall	Precision
Mean		
lb_{ω}		
PoI		
ExI		

Table 4.5.9: Averaged recall and precision of filters with constant output size.

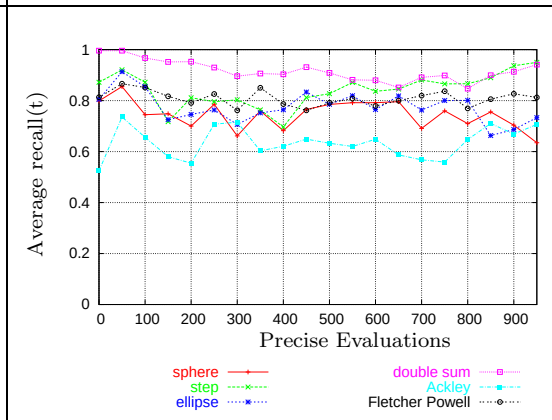
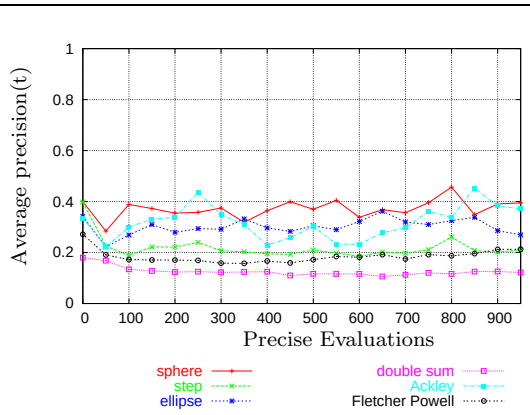
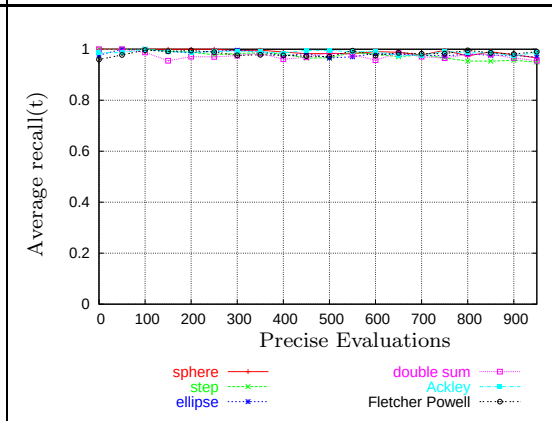
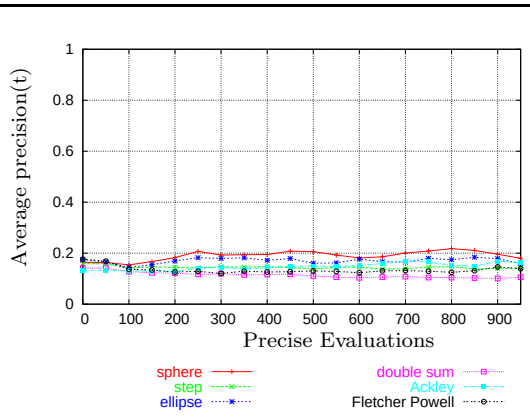
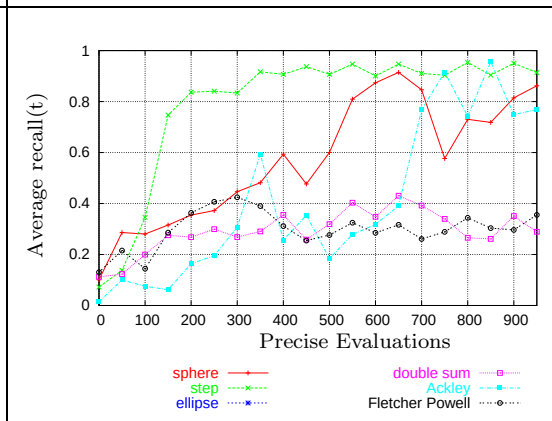
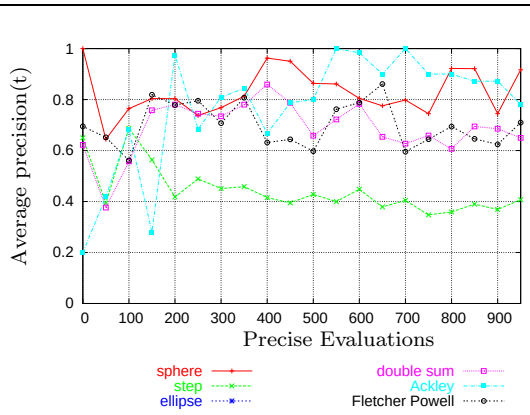
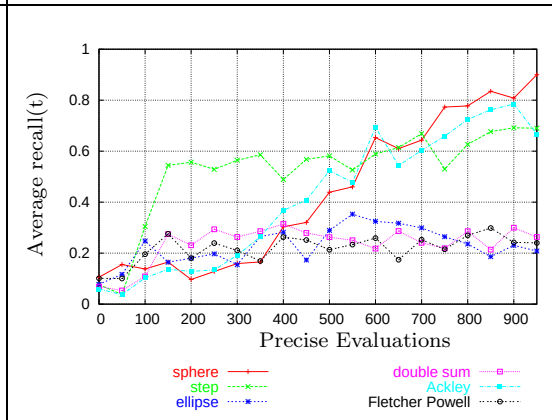
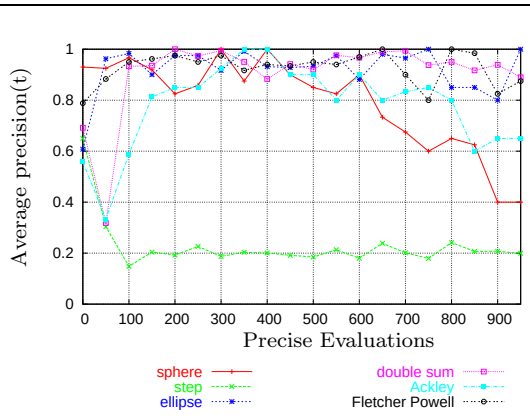
	Recall	Precision
lbi		
rfilt		
mli		
pfilt		

Table 4.5.10: Averaged recall and precision values of filters with variable output size.

other hand. If one can spend more than one run on a particular problem, it would be recommended to try both strategies.

4.5.5 Study of strategy parameters

Next, two studies reported originally by Emmerich et al. [EBN05] are reproduced. The first study aimed at studying the behavior of the MAES for different settings of the important strategy parameters μ (the number of individuals in the parent population) and ω (confidence factor). A second study addresses the long term behavior of the MAES for local optimization.

The influence of μ and ω on the convergence dynamics was studied on the 20-dimensional Ackley function (appendix 4.5.1). Figure 4.5.18 displays the results. It indicates that an optimal value for ω exists. For high ω values ($\omega = 3$), the intensive exploration entailed a low convergence speed. For low ω values ($\omega \leq 1$), the MAES likely converge to a local optimum, since the confidence information had less impact on the criterion for pre-selection.

Additionally, the effect of the population size μ on the convergence speed of MAES is illustrated in figure 4.5.18. Low values of μ implied a higher selection pressure. This led to significantly better results during the first generations but increased the risk of premature stagnation in local optima. For the discussion of the effect of the selection pressure on the extinction of sub-populations on multi-modal landscapes we refer to studies by [SEP04, PSE05], in which the author of this thesis was involved. Note, that stagnation occurred later than in case of runs with a too small ω. It is also noteworthy, that choices $\omega = 2$ and $\mu = 5$ led to a similar behavior to those of the ExI and PoI strategies with $\mu = 5$. From the experience we gathered so far, $\mu = 5$ and $\omega = 2$ is also a good default setting for the MAES that allows a fast convergence to local optima as well as an increased robustness as compared to the mean value criterion.

Also, the setting for ν has been checked for constant output-size filters. Without displaying results, we note that a higher value of ν decreases the local convergence rate significantly, while a two low setting of ν increases the risk of pre-mature convergence. Values of ν between 10 and 20 turned out to yield the best results. However, we choose the setting of $\nu = 20$ in order to guarantee a sufficient selection pressure in case that the metamodel generates a more or less random sorting on the offspring population. If the prescreening produces a random sorting on the offspring population, the convergence behavior of the $(\mu + \nu < \lambda)$-MAES would correspond to the a simple $(\mu + \lambda)$-ES.

However, there are still some *open questions* concerning the control parameters of the MAES. For example we have not tested if there is interaction between the parameters ν, μ and ω. Moreover, different step-size adaptation mechanisms and/or recombination operators could be tested. An interesting approach for finding optimal parameter settings for evolutionary algorithms has recently been proposed by Bartz-Beielstein [BB05]. It could be an interesting direction for future research to apply this approach to obtain near-optimal parameter settings for the MAES on different problem classes.

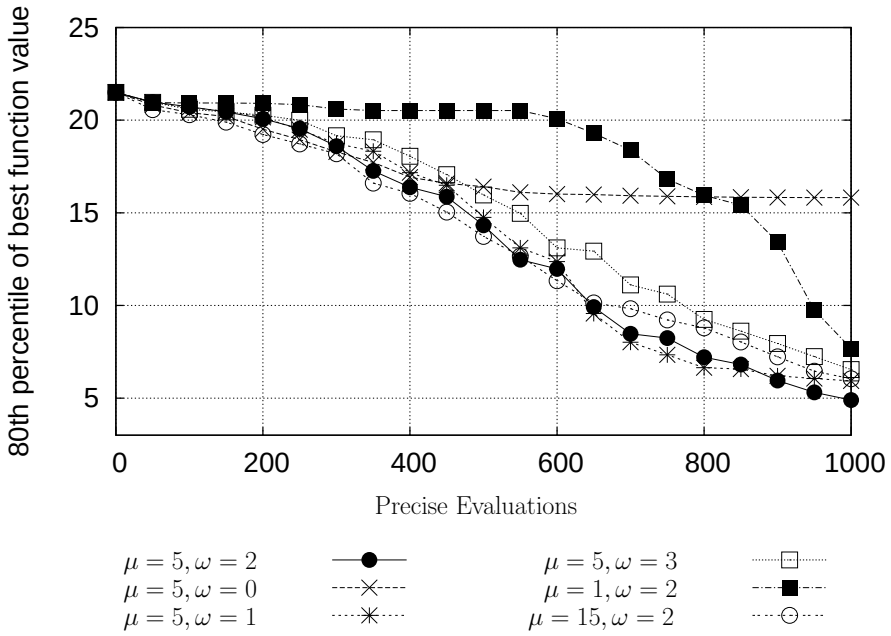


Figure 4.5.18: Development of the 80%-quantiles for best found function values for differently parameterized versions of the MAES. The runs have been conducted on the 20-dim. Ackley function. Different settings for the population size μ and the confidence factor ω in the $(\mu + 5 < \lambda)$-MAES using the criterion described in equation 3.4.17 are displayed.

4.5.6 Long term behavior

Finally, the long term behavior of the MAES was studied. Figure 4.5.19 displays results for a long run with 2000 precise evaluations on the 20-dim. ellipsoid problem. The results indicate that the MAES is capable to approximate a local optimum with a high precision and does not converge to a false optimum, as it might be suspected (cf. [Jin05]). Another important observation is that the absolute error of the prediction shrinks proportionally to the distance from the local optimum.

Summary of experimental results

The experiments already give a good impression of the behavior of the MAES variants on problems with moderate dimensions. On the basis of the performance studies it is possible to select the adequate type of filter, if we have an assumption about the topology of the landscape on which to optimize. First attempts towards an explanation of the observed behavior have been made and supported by experimental data.

Summing up, let us highlight the lessons we have learned from the study. Due to their empirical nature the following results should be understood as rules of thumb for the practitioner who wants to apply the MAES and not as precise results:

- The MAES working with constant output size filters accelerate the standard ES by a constant factor ranging from two to eight for the tested problems.

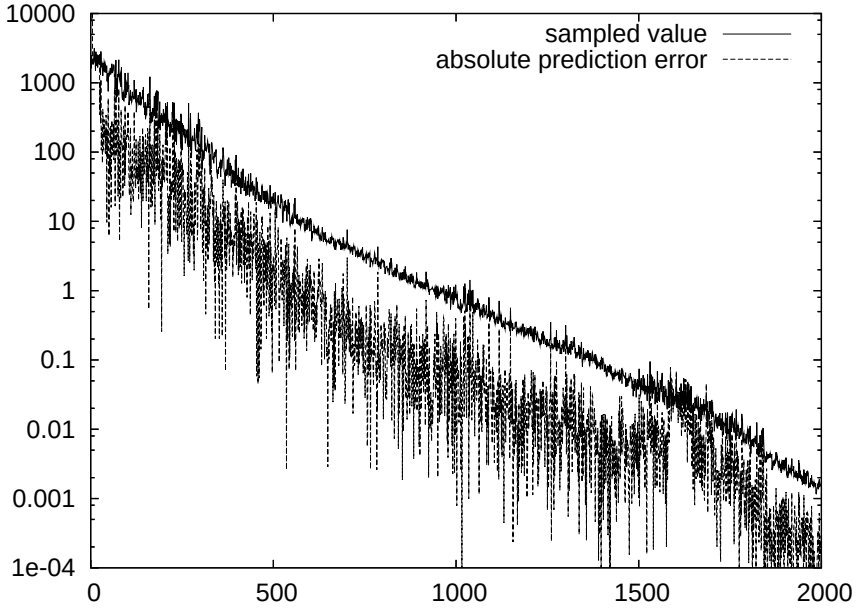


Figure 4.5.19: A run of the $(20+5 < 100)$-MAES with PoI pre-screening on the ellipsoid problem with 2000 evaluations. It demonstrates that the MAES is capable to converge to a high precision. Also, it can be obtained that the error of the predictions shrinks proportionally with the distance to the optimum.

- The linear convergence order of the ES on simple problems gets preserved
- Using the PoI, ExI or lower confidence bound criterion might slightly decrease the local convergence speed but significantly improves the results on some multimodal functions
- The lower confidence bound strategy performed better on discontinuous problems with plateaus, while the PoI showed better results for local optimization problems and for the Fletcher Powell Problem. The ExI criterion only outperformed these strategies in case of correlated variables.
- The quality of the metamodel trained merely from the pre-selected individuals remains sufficiently well to establish an approximate sorting on the offspring population. This sorting is significantly better than a random sorting of the population.
- The choice of ν determines significantly the convergence speed and robustness of the strategy. The $(5+20 < 100)$-ES provides a robust choice for population parameters.
- The theoretical assumptions about the precision or recall of the variable output filters prove to be valid on the basis of the experimental data. The R_ω-filter MAES best emulates the 'true' behavior of the ES.
- Variable ratio filters easily tend to result in extreme values of ν, thus either deteriorating convergence speed or robustness. Moderate values of ω are recommended to achieve a more balanced behavior.

Further experimental research will have to deepen the understanding of the MAES. Studies of the distribution of the sampled and/or pre-selected populations hopefully can provide further insights on their dynamical behavior. Probably the most important question will be to derive a robust control of ν. It will not suffice to base the decision whether to spend more objective function evaluations merely to enhance the quality of the metamodel and to focus on emulating the 'true' ES, by maximizing recall and/or precision values. It is far more important to avoid premature stagnation of the strategy by detecting new unexplored but promising areas of the search space. The lower confidence bound and the LBI filter are first attempts in this direction that needs to be further explored.

4.6 Conclusions

In this chapter the MAES has been introduced. The metamodel-assistance can easily be integrated into existing versions of the ES. It requires the installation of a IPE-filter in the main loop of the ES and the maintenance of a database of conducted objective function evaluations, that is used for approximating the function values of offspring individuals by means of GRFM.

Various types of filters have been introduced and compared, both on a theoretical and on a practical level thereby making extensive use of the variance information provided by the GRFM. A theory of filters for the MAES has been outlined. First, fixed cardinality filters have been used that sort the population by scalar criteria. Differences and similarities between these criteria have been pointed out. Then filters with a variable number of pre-selected solutions have been proposed and related to each other. It has been found that the concepts of precision, recall and permeability are important for evaluating and adjusting the characteristics of these filters. Based on the theory of interval orders, the P_ω-filter and R_ω-filter have been derived.

Next, the convergence properties of the MAES have been studied. It has been pointed out that under the condition that at least one individual passes the filter each time, the global convergence property of the ES on regular functions is inherited by the MAES. With regard to the performance the speed-up for a perfect filter, i. e. a filter that selects the same individuals than the replacement, has been measured, thereby considering different times for the approximation and for the objective function evaluation. An expression for a break even point has been found, for which the speed of the ES and MAES are equivalent in the best case scenario of optimal precision and recall. It has also been motivated that for the analysis of the MAES experimental results are needed, since the loss of the Markov property forbids the analysis of the MAES with the standard theory.

Measures for experimental studies of the MAES have been suggested. Three kind of measures have been proposed. For the performance analysis history plots standard techniques like plots of the median and 80% quantiles MAES seem to be adequate to measure average behavior and robustness. It has been pointed out that the number of objective function evaluations determines the time of the strategies and is usually limited.

New measures for the analysis of filter characteristics have been proposed, that quantify their precision and recall based on the number of selected and non-selected relevant offspring individuals. For accuracy measuring of the metamodel the double-logarithmic

$y \sim \hat{y}$ and $y \sim \mathrm{lb}_\omega$ plot based on cross-validations has been adopted.

In order to measure the capability of a criterion-based filter to establish a proper ranking on the offspring population, the number of ordered pairs provide a useful measure. By employing the theorem of Sachkov, it has been demonstrated, how this measure can be used to disapprove the hypothesis that sorting by means of function approximations is no better than pure random sorting. Both, the numerical accuracy measure and the number of sorted pairs measure can be used for an online screening of the metamodel quality.

Finally, the results of various test runs of the MAES on selected artificial test problems have been displayed and interpreted. It has been obtained in the test runs that the pre-screening with MAES speeds-up the ES significantly on a set of problems that include some of the most common difficulties for optimization strategies. Exclusively in the presence of discontinuities and plateaus the MAES failed to perform better than the corresponding ES. The test runs also indicate that the MAES is quite robust against little errors in the model assumptions. Furthermore, the runs adds further evidence to the hypothesis that the incorporation of the variance increases the robustness of the strategies, in particular for multimodal functions. The reason for the increased robustness has not been due to a better sorting or a better recall of the strategies employing the lb_ω filter. More likely the increased robustness is caused by the mechanism that the variance term drives the MAES into new unexplored regions of the search space.

5 Metamodel-assisted constrained optimization

In throwing stone at a mouse, beware of breaking a precious vase.

Chinese proverb

In this chapter we will deal with the handling of implicit (inequality) constraint functions in the MAES, i. e. constraint functions that are evaluated by means of the time consuming evaluator. Accordingly, we discuss an instantiation of the problem definition in section 5.2, where the evaluation tool describes a mapping $\mathbb{R}^d \to \mathbb{R}^{n_f+n_g}$ with $n_g > 0$ and $n_f = 1$.

For constraint handling we adopt the metric penalty approach as described by Hoffmeister and Sprave [HS96]. We propose selection procedures for candidate solutions, the objective and constraint functions of which have been evaluated approximatively.

The different topics of constrained metamodel-assisted optimization will be discussed in the following order: In section 5.1 we start with a brief survey of constraint handling methods and discuss their applicability for the MAES. Section 5.2 introduces the treatment constraints within the ES and MAES based on a metric penalty approach. Section 5.3 generalizes the IPE-filters of the MAES (section 4.2) to the constrained case. The remainder of the chapter (section 5.5) provides a study on benchmark problems.

5.1 Constraint handling methods

In the past, several techniques have been proposed for the treatment of constraints in evolutionary algorithms. A comprehensive summary of constraint handling methods in EA is given by Coello Coello [Coe99]. Classical constraint handling methods are also discussed in [BFM97]. Next, we pick out some of the most important approaches to deal with constraints and discuss their applicability within the MAES.

A common way to handle simple constraints is to transform the search space in a way that it contains no more infeasible solutions. This can be done by so-called decoders (cf. MF00) or just by choosing an appropriate representation. This approach can be generalized in a straightforward manner, if the metamodel is learned for the transformed search space. However, this approach demands for analytical expression of the constraint functions. Hence, it cannot be applied for the treatment of implicit constraints. A related approach for handling constraints is the use of repair heuristics. Starting from an infeasible point, a nearby feasible point is searched for, e. g. by means of an local

minimization of a penalty term. Then the objective function value of the repaired solution is assigned to the individual as fitness value [Coe99]. Another idea would be to reject and re-generate offspring solutions until a sufficient number of offspring solutions has been obtained. All these approaches typically demand for a large number of constraint function evaluations and are thus inapplicable if the evaluation of the constraint function is time consuming.

There is a variety of methods that use non-standard population models and selection operators in order to incorporate constraints. For example Paredis [Par95] used a predator prey approach with two separate populations and Schoenauer et al. [SX93] used an scheme where attention is paid to different constraints at different times. Recently, Kramer [Kra03] proposed meta-evolutionary approaches and an co-evolutionary methods for handling constraints in evolution strategies. We will not consider co-evolutionary methods in this thesis, though it may be an interesting direction for future research.

Penalty function methods are probably the most common approach to handle implicit constraints. The general principle of these methods is build a penalty function that integrates the function values of violated constraints and is added to the objective function value. Thereby the constrained optimization problem is re-casted as a single-objective optimization problem. The literature distinguishes between dynamic and static penalty functions. The former class of penalty methods works with dynamical schedules to control the impact of the penalty term during optimization. A classical approach for this is the sequential unconstrained minimization technique (SUMT) by Fiacco and McCormick [FM90]. For a more recent discussion of the SUMT we refer to [Nas98]. However, the metamodels proposed in this thesis cannot learn dynamically changing landscapes. Moreover, the adoption of a scheme like it is used in the SUMT methods involves the choice of further parameters by the user. These are the reasons why we cannot use these methods in this thesis.

Static penalty functions usually introduce a penalty term that assures that infeasible solutions are always inferior to any feasible solution. As a consequence they introduce a discontinuity at the constraint boundary and thus it is difficult to model the resulting fitness function by means of interpolating metamodels like GRFM.

In this work we base the constraint handling methods on the *metric penalty approach*, as suggested by Hoffmeister and Sprave [HS96]. Instead of building a surrogate function from the constraint and objective function values, Hoffmeister and Sprave proposed to establish a quasi-order on the set of solutions that is based on the values of the constraint functions and the objective function. Like for the static penalty function approach, feasible solutions are always ranked better than infeasible ones. Feasible solutions are compared by their objective function value and infeasible solutions are compared by the values of the violated constraint functions. Since ES selection schemes usually do not demand for absolute fitness values but only for a ranking, this approach provides an elegant way of constraint handling. In the next chapter we will be more precise on how this approach and how it has been implemented in the MAES.

5.2 Constrained optimization with evolution strategies

Constrained optimization problems, with one objective function f and n_g ($n_g > 0$) constraint functions $(g_1, \ldots, g_{n_g})$ can be formally stated as:

$$f(\mathbf{x}) \rightarrow \min \tag{5.2.1}$$
$$g_1(\mathbf{x}) \leq 0, \ldots, g_{n_g}(\mathbf{x}) \leq 0 \tag{5.2.2}$$
$$\mathbf{x} \in \mathbb{S} \subset \mathbb{R}^d \tag{5.2.3}$$

For notational convenience, we shall denote vectors of output values by $\mathbf{y} = (y_1, \ldots, y_{n_g+1})$, whenever this seems suitable. In this notation the first position of the vector y_1 denotes the objective function value and the remaining positions y_2, ..., y_{n_g+1} denote the constraint function values.

As mentioned in the introduction, a common approach is the static penalty approach. The basic idea is to extend the definition of the objective function as follows:

$$f(\mathbf{x}) + \begin{cases} 0 & \text{if } \mathbf{g}(\mathbf{x}) < 0 \\ f^{\max} + \delta(\mathbf{g}(\mathbf{x})) & \text{otherwise} \end{cases} \tag{5.2.4}$$

with

$$\delta(\mathbf{g}(\mathbf{x})) = \sum_{i=0}^{n_g} (\max\{0, g_i(\mathbf{x})\})^{\gamma}.$$

Here we define that $\mathbf{g}(\mathbf{x}) < 0$ is true, if and only if all vector positions are lower than or equal to zero and at least one vector position is strictly lower than zero. In a similar manner, we define to be true $\mathbf{g}(\mathbf{x}) \leq 0$, if and only if all constraint values are lower or equal than zero. By δ we denote a *penalty function* that takes the value of zero, whenever no constraint is violated, and a positive value that reflects the severity of the constraint violation, otherwise. The penalty function establishes a metric in the infeasible region that can be used as an orientation for the optimization algorithm to find the feasible region. It is a common choice to use a quadratic penalty function $\gamma = 1$ or $\gamma = 2$. The latter choice is typically made when dealing with equality constraints. However, in this thesis we focus on inequality constraints why $\gamma = 1$ can be an adequate choice, as well.

The value of $f^{\max}$ needs to be chosen such that feasible solutions always dominate infeasible solutions. As indicated above, Hoffmeister and Sprave [HS96] pointed out that for rank-based optimization algorithms there is no need to define a parameter $f^{\max}$. Instead of computing a scalar value from the constraint values and objective function value, it suffices to establish a preference relation on the set of output vectors. This can be done by defining the following preference relation between output vectors containing both, the objective function and the constraint function values.

Let $\mathbf{y} = (y_1, \ldots, y_{n_f+1})^T =: (y_f) \circ \mathbf{y}_g$ and $\mathbf{y}' = (y'_1, \ldots, y'_{n_f+1})^T =: (y'_f) \circ \mathbf{y}'_g$ denote two result vectors. Then

$$\begin{aligned} \mathbf{y} \prec_c \mathbf{y}' (\text{ say } \mathbf{y} \text{ dominates } \mathbf{y}') :\Leftrightarrow & \\ \mathbf{y}_g \leq 0 \ \wedge \ \mathbf{y}'_g \leq 0 \ \wedge \ y_f < y'_f \quad & \vee \\ \mathbf{y}_g \leq 0 \ \wedge \ \mathbf{y}'_g > 0 \quad & \vee \\ \mathbf{y}_g > 0 \ \wedge \ \mathbf{y}'_g > 0 \ \wedge \ \delta(\mathbf{y}_g) < \delta(\mathbf{y}'_g). & \end{aligned} \tag{5.2.5}$$

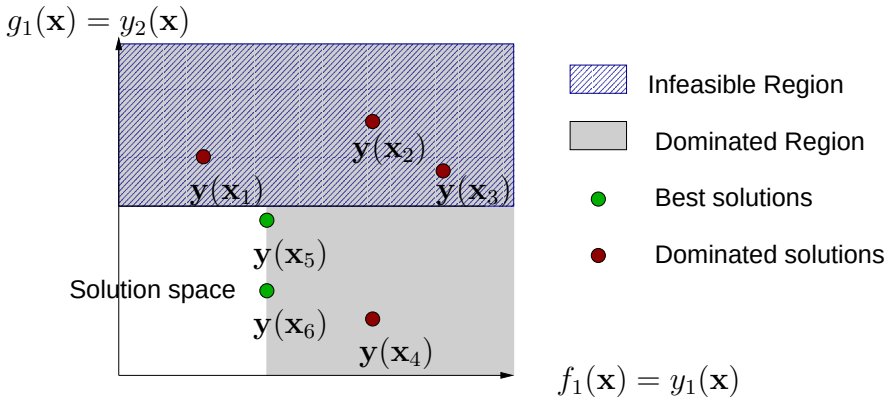


Figure 5.2.1: The rank ordering of solutions due to the constrained quasi-order $\prec_c$.

It will turn out in the subsequent discussion that the idea to define such a preference relation among solutions is a very suitable way to handle constraints that will allow for a coherent design of filtering strategies. The preference relation $\prec_c$ is used in the replacement of the ES in order to establish a quasi-order[1] among the solution candidates $\mathbf{x} \in \mathbb{S}$. Hence, the ES searches for minimal elements of this quasi-order.

Example: For a finite set the quasi-order of solutions is visualized in figure 5.2.1 for a space with $n_f = 1$ and $n_g = 1$. The solutions $\mathbf{x}_5$ and $\mathbf{x}_6$ share the same first rank. The solution $\mathbf{x}_4$ is assigned to the second rank. The solutions $\mathbf{x}_3$, $\mathbf{x}_1$ and $\mathbf{x}_2$ are assigned to the last three ranks in the given sequence. The sequence of the infeasible solutions $\mathbf{x}_1, \mathbf{x}_2$ and $\mathbf{x}_3$ is determined by the distance to the constrained boundary that is measured by the penalty function.

The introduction of the extended preference relation makes it possible to handle constraints within the standard ES. However, further adaptations to the ES have to be made in order to accelerate convergence in the presence of constraints. For example, Kramer [Kra03] found that correlated mutations and/or the introduction of a minimal step-size can be important measures in order to prevent the standard ES from stagnating in suboptimal regions of the search space.

5.3 Generalization of the IPE-filters

In order to extend the MAES for constrained problems further adaptations have to be made. In particular, IPE-filters have to be adapted for metamodels with multiple outputs.

Moreover, whenever dealing with a single-objective function and one or more constraints ($n_f = 1, n_g > 0$), the IPE-filters introduced in section 4.2 should be based on the preference relation in equation 5.2.5.

[1] an order where the same rank can be shared by different solutions

5.4 Metamodels with multiple outputs

Our goal is to utilize information about the multivariate probability distribution of an output vector, instead of a one-dimensional distribution for a scalar output in the IPE-filters. A common simplification is to model outputs of different constraint and objective functions separately [SWJ98]. Note that after the training of the GRFM the metamodels can also can also have a different parametrization of their correlation function. This can be advantageous, since some of the output functions may be more sensitive than others. The usage of independent metamodels still allows to get arbitrarily precise metamodels, whenever the local density of samples is sufficiently high.

For more accurately modeling functions with directly correlated outputs, specialized multivariate GRFM (Co-Kriging models) could be considered [Mye92]. Instead of independent gaussian distributions for the single outputs, a full multivariate gaussian distribution describes the likelihood of different realizations of the output vectors within this approach. However, the co-kriging approach can be numerically demanding and add instabilities to the prediction procedure, why it should be treated with caution[SWJ98].

In this thesis, will refer to the full multivariate distributions only in the conceptual design of the filters, by first stating general expressions for the pre-screening criteria, if possible, and then instantiating them for the special case of independent gaussian distributions describing the outputs.

For the independent metamodels, independent gaussian distribution with mean value $\hat{\mathbf{y}} \in \mathbb{R}^{n_y}$, $n_y = n_g + 1$ and standard deviations $\hat{\mathbf{s}} \in \mathbb{R}_+^{n_y}$ are considered as predictors, adopting a notation by Schonlau et al. [SWJ98] for constrained bayesian optimization. Here the values $\hat{y}_1$ and $\hat{s}_1$ correspond to the predicted objective function f, whereas the values $\hat{y}_2, \ldots, \hat{y}_{n_g+1}$ and $\hat{s}_2, \ldots, \hat{s}_{n_g+1}$ correspond to the predictions of the constraint functions $g_1, \ldots, g_{n_g}$, respectively.

5.4.1 Mean value and lower confidence bound filter

Whenever ranking with the $\prec_c$ relation, it seems quite natural to use the following generalizations of the mean value ($\hat{y}$) and lower confidence bound (lb_ω) filter:

For *mean value filter* calculate $\hat{y}_i(\mathbf{x})$, for $i = 1, \ldots, n_g + 1$ and rank all solutions by means of these mean value vectors to the $\prec_c$ quasi-order. Again, this criterion does not make use of the uncertainty measure $\hat{\mathbf{s}}(\mathbf{x})$, and thus it can be used with metamodels (e. g. RBFN) that do not provide such a measure.

Accordingly, for the generalized lb_ω criterion the algorithm ranks solutions by means of lower confidence bound vectors

$$\mathrm{lb}_{\omega,i} = \hat{y}_i(\mathbf{x}) - \omega \cdot \hat{s}_i(\mathbf{x}), i = 1, \ldots, n_g + 1. \tag{5.4.6}$$

Here, the computed lower confidence bound vectors can be understood as a lower confidence bound for the vector valued result. An illustrative example for a problem with one constraint function is given in figure 5.4.2.

As in the single-objective case the mean value filter and the lb_ω filter select a subset of

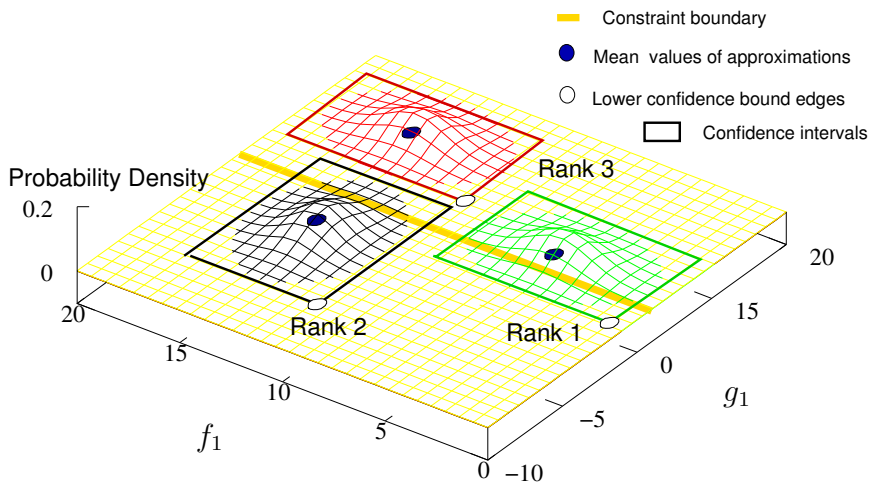


Figure 5.4.2: Example for establishing a rank order among approximations for a problem with one objective function and one constraint function. The rank numbers in the drawing result if the lower confidence bound criterion gets applied. If we would rank with the mean value criterion, the first and second ranked solutions would exchange their position.

fixed cardinality with the user-specified output size ν. The best ν solutions with regard to the applied criterion ($\hat{\mathbf{y}}$ or lb_ω) and the preference relation $\prec_c$ shall pass the filter.

5.4.2 Improvement-based filters

For the improvement-based filters we again need a reference point, which can be chosen as the worst function value in the parent population P_t, if an improvement of the population is envisaged. The reference point will be denoted with $\mathbf{y}_{ref} = (f_{ref}, g_{1,ref}, \ldots, g_{n_g,ref})^T$. Assuming that the reference solution $\mathbf{y}_{ref}$ is feasible, the improvement criteria can be generalized in a straightforward manner:

Since $\hat{\mathbf{y}}(\mathbf{x})$ is the most likely outcome of the computer experiment for any $\mathbf{x} \in \mathbb{S}$, the most likely improvement criterion can be redefined as:

$$\mathrm{MLI}(\mathbf{x}) = \begin{cases} \mathrm{I}(\hat{y}_1(\mathbf{x})) & \text{if } \hat{y}_i(\mathbf{x}) \leq 0, i = 2, \ldots, 1 + n_g \\ 0 & \text{otherwise} \end{cases} . \tag{5.4.7}$$

Note that this criterion is also valid for general multivariate gaussian distributions with correlated output variables. As in the single-objective case, the MLI filter shall let only those solutions pass that have a positive MLI.

Accordingly, the LBI criterion can be redefined for constrained optimization as:

$$\mathrm{LBI}_\omega(\mathbf{x}) = \begin{cases} \mathrm{I}(\hat{y}_1(\mathbf{x}) - \omega \cdot \hat{s}_1(\mathbf{x})) & \text{if } \hat{y}_i(\mathbf{x}) \leq 0, i = 2, \ldots, 1 + n_g \\ 0 & \text{otherwise} \end{cases} . \tag{5.4.8}$$

This definition is only meaningful for independent output variables (cf. section 5.4). In this case we can calculate lower confidence bound vectors, which, under the GRFM

assumptions, are valid with the probability

$$p_\alpha = \prod_{i=1}^{1+n_g} \Phi(\frac{\hat{y}_i(\mathbf{x}) - \omega \cdot \hat{s}_i(\mathbf{x}) - \hat{y}_i(\mathbf{x})}{\hat{s}_i(\mathbf{x})}) = (\Phi(-\omega))^{n_g+1} \tag{5.4.9}$$

Note that for a correlated distribution for the estimated output variables there is no straightforward generalization of the LBI criterion. In that case the ExI and PoI measures provide clearer concepts of how to integrate the uncertainty information.

Let us now turn to these integral criteria that have already been described for the single-objective case in section 4.2. The PoI criterion can be written as

$$\mathrm{PoI}(\mathbf{x}) = \int_{\mathbf{y} \in \mathcal{H}_f} \mathrm{PDF}_{\mathbf{x}}(\mathbf{y}) d\mathbf{y} \tag{5.4.10}$$

where $\mathrm{PDF}_{\mathbf{x}}$ denotes the estimated probability density function for response vectors for input $\mathbf{x}$ and $\mathcal{H}_f$ denotes the part of the $n_g + 1$ dimensional solution space dominated by $\mathbf{x}_{ref}$ (cf. Fig. 5.2.1) that reads:

$$\mathcal{H}_f := [(\underbrace{-\infty, \ldots, -\infty}_{n_g+1 \text{ times}}), (f_{ref}, \underbrace{0, \ldots, 0}_{n_g \text{ times}})]. \tag{5.4.11}$$

Let $\varphi : \mathbb{R} \to [0, 1]$ denote the PDF of the standard gaussian distribution. Then for independent response vector distributions (cf. section 5.4) the PoI criterion can be calculated as

$$\mathrm{PoI}(\mathbf{x}) = \int_{\mathbf{y} \in \mathcal{H}_f} \prod_{i=1}^{n_g+1} \varphi(\frac{y_i - \hat{y}_i(\mathbf{x})}{\hat{s}_i(\mathbf{x})}) d\mathbf{y} = \Phi(\frac{\hat{y}_1(\mathbf{x}) - f_{ref}}{\hat{s}_1(\mathbf{x})}) \cdot \prod_{i=2}^{n_g+1} \Phi(\frac{-\hat{y}_i(\mathbf{x})}{\hat{s}_i(\mathbf{x})}) \tag{5.4.12}$$

provided that the best solution is feasible.

For the ExI criterion a general formula for the expected improvement reads:

$$\int_{\mathbf{y} \in \mathcal{H}_f} (f_{ref} - y_1) \cdot \mathrm{PDF}_{\mathbf{x}}(\mathbf{y}) d\mathbf{y} \tag{5.4.13}$$

and for the case of an independent distributions of the output variables (cf. section 5.4) Jones et al. [JSW98] derived the direct formula for applications in bayesian global optimization:

$$\mathrm{ExI}(\mathbf{x}) = [(f_{ref} - \hat{y}_1(\mathbf{x}) \cdot \Phi(f_{ref}) + \hat{s}_1(\mathbf{x}) \cdot \varphi(f_{ref} - \hat{y}_1(\mathbf{x}))] \cdot \prod_{i=2}^{n_g+1} \Phi(\frac{-\hat{y}_i(\mathbf{x})}{\hat{s}_i(\mathbf{x})}). \tag{5.4.14}$$

Having defined these criteria, the ExI-filters and PoI-filters can be used in the constrained case as criteria in order to preselect the subset of the ν most promising solutions.

Also filters with variable output size could be considered. Again, for the PoI and ExI-filters threshold values could be provided that determine the minimal value τ that a solution needs to have in order to pass the filter.

5.4.3 Comparison of the PoI- and MLI-filter

For the threshold version of the PoI filter, i. e. the PoI_τ-filter, there is no longer an equivalence between the MLI-filter for a threshold probability $\tau = 0.5$ and between t the LBI_ω-filter, and the PoI_τ-filter with $\tau = \Phi(-\omega)$. The reason for this is, that, unlike for the MLI- and LBI_ω-filters, the PoI_τ-filter considers the distance to the constraint boundary for feasible solutions. The larger this distance gets, the more likely the individual is an improvement. Only for values located exactly at the constraint boundary and those with a mean value located in the infeasible region both filters decide the same way whether or not to reject a solution.

To put things in more concrete terms, the following lemma shall be stated for the permeability of the LBI_ω-filter.

Lemma 6. For $n_f = 1$ and $n_g > 0$ the LBI_ω-filter is less permeable than the PoI_τ-filter with $\tau = \Phi(-\omega)^{n_g+1}$ and the MLI-filter is less permeable than the PoI_τ-filter with $\tau = 0.5^{n_g+1}$.

Proof. We will prove the result for the LBI_ω criterion. The result for the MLI criterion will then follow immediately, from the equivalence between the MLI-filter and the LBI_ω-filter for $\omega = 0$. Let us assume the reference solution has an objective function value of f_{ref}. Furthermore, as always, we will assume that it is feasible. Let us assume a candidate solution for which all approximated constraint function values are active, that is $\hat{y}_i(\mathbf{x}) \geq 0$ for $i = 2, \ldots, n_g + 1$. Then $\text{PoI}_{\Phi(-\omega)^{n_g}}$ accepts this solution exactly if the LBI_ω does. This is the case, if the solution is placed on the boundary edge $\mathbf{y}_0$ with

$$\mathbf{y}_0 = (f_{ref}, \underbrace{0, \ldots, 0}_{n_g \text{ times}}). \tag{5.4.15}$$

If the coordinates of the constrained values of the solution move further below zero, the PoI_τ value further increases and $\text{PoI}_\tau(\mathbf{x})$ might exceed the threshold probability $\Phi(-\omega)^{n_g}$ for further solutions. On the other hand, the permeability of the LBI_ω-filter remains unchanged. □

Summing up, it has been shown, that all expressions for the improvement-based filters can be generalized to the constraint case. Moreover, it turns out that these generalized procedures can be easily computed, if we assume independent output variables. In particular, for the integral criteria PoI and ExI explicit expressions can be derived.

5.4.4 Interval filters

Last but not least, let us give a sketch of the generalization for interval based criteria that have been introduced in section 4.2.3.

We recall that one-dimensional confidence intervals have been used for filtering solutions. The P_ω-filter aimed at avoiding, with a high probability, the selection of solutions that are not competitive with the solutions in P_t (mistake B in section 4.2.3), and the R_ω-filter aimed at avoiding, with a high probability, the rejection of solutions that are competitive with solutions in P_t (mistake A in section 4.2.3).

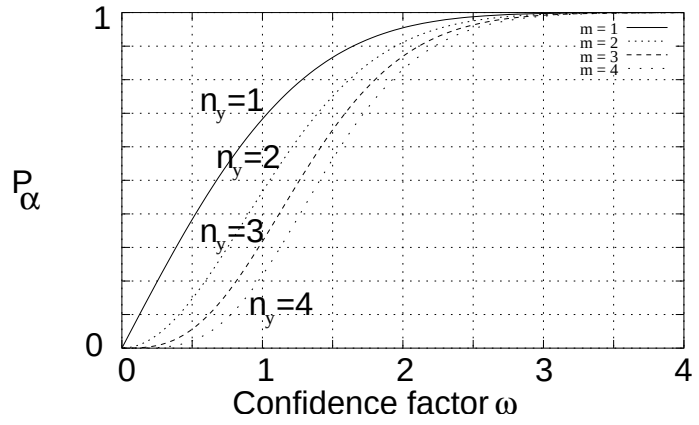


Figure 5.4.3: Computation of the confidence factor ω for a desired p_α value and a given number of independent response functions n_y (equation 5.4.16).

This idea can also be generalized for optimization with multiple outputs. The basic idea for the generalization is to work with confidence interval boxes bounding the precise function value instead of one-dimensional intervals. Within the pre-selection with multiple outputs, lower and upper confidence bounds of *confidence interval boxes* will be computed. Here, an *interval box* is a hyper-rectangle defined by an upper confidence bound vector $\mathbf{u} \in \mathbb{R}^{n_y}$ and a lower confidence bound vector $\mathbf{l} \in \mathbb{R}^{n_y}$ with $l_i \leq u_i, i = 1, \ldots, n_y$. As a confidence interval box for a random variable we denote a hyper-boxes in which the realization of a random variable is located with a certain confidence probability p_α. Confidence interval boxes are symmetrically placed around the mean prediction value. In case of independent distributions such confidence interval boxes can be calculated by $B(\mathbf{x}) = \hat{\mathbf{y}}(\mathbf{x}) \pm \omega \hat{\mathbf{s}}(\mathbf{x})$ (cf. Figure 5.4.2). The value for ω can be related to a user specified confidence probability p_α – the probability for the true result to lie inside an interval box $(= \Pr(\mathbf{y} \in B(\mathbf{x})))$ – by means of

$$p_\alpha(\omega) = \Pr(\mathbf{y} \in B(\mathbf{x})) = (1 - 2\Phi(-\omega))^{n_y}. \tag{5.4.16}$$

Here $\Phi(y) := \frac{1}{2}(1 + \mathrm{erf}(\frac{y}{\sqrt{2}}))$ is the cumulative gaussian distribution for the desired confidence level p_α. It is impossible to invert the expression for p_α analytically. Thus, values for ω have to be obtained by means of approximations. Practically, the ω value for p_α can be obtained from a graph of this function. Figure 5.4.3 depicts this graph for different n_y. For example for a desired confidence probability of $p_\alpha = 90\%$ and two outputs, the user has to set the confidence factor to $\omega \approx 2.0$.

Once having established the confidence interval boxes a generalization of the filters proposed in section 4.2.3 is straightforward, if we apply the rank ordering established by $\prec_c$ on the lower and upper confidence bound edges.

For the example in figure 5.4.2, an R_ω-filter with $\mu = 1$ would select the two solutions located in the lower part, whereas a P_ω-filter with $\mu = 1$ would select no solution, since it is not clear, which one of the two solutions located in the lower part is the best solution.

From this simple example we can already get the impression that the interval based filter tends either to extreme small or extreme large sizes of the output size. Also practical experiments we conducted with $\omega = 2$ on the test problem described in the next section, approved this, why we omitted a further study of this approach.

5.5 Study on an artificial test problem

As an example for an artificial test problem Keane´s bump function has been chosen.

$$\min -\frac{|\sum_{i=1}^{n}\cos^4(x_i) - 2\prod_{i=1}^{n}(\cos^2(x_i))|}{\sqrt{\sum_{i=1}^{n} i * x_i^2}}, \tag{5.5.17}$$

$$\prod_{i=1}^{n} x_i > 0.75, \ \sum_{i=1}^{n} x_i < \frac{15n}{2} \tag{5.5.18}$$

$$x_i \in]0, 10.0]$$

The Keane bump function is some kind of standard benchmark for nonlinear constrained optimization. It is highly multimodal and its optimum is located at the nonlinear constrained boundary. The true minimum of this function is unknown.

In order to evaluate the performance of MAES coupled with different pre-screening criteria in the constrained optimization problems, test runs on the 10-D Keane problem (equation 5.5.17) have been conducted. Results have been summarized in Fig. 5.6.4 (median) and 5.6.5 (reliability measured by 80% quantile). Two important observations could be made. First of all, any metamodel-assisted strategy performs significantly better than the corresponding EA without metamodel-assistance. Second, strategies using the confidence information perform much better than the ones utilizing only the predicted function value. However, the differences between the three pre-screening criteria that use confidence information (lower confidence bound, PoI and ExI) are less significant for this problem.

A possible explanation of the good performance of the strategies that take into account the uncertainty information is, that they more likely place points near the constraint boundary. Besides, also the advantage of these strategies on multimodal problems, that has already been pointed out in chapter 4 is a possible explanation for the superior performance of these strategies.

Further examples for the successful application of the MAES will be given in the chapter about applications (chapter 8).

5.6 Conclusions

In this chapter, extensions of the MAES for the optimization with time consuming constraint function evaluations have been derived. The basic principle of the generalization was to train separate metamodels for all constraint functions, instead of training the metamodel from the values of a penalized objective function. A vector valued comparison has been employed in order to compare solutions. For the lower confidence bound criterion, mean value criterion and their corresponding improvement based criteria LBI and MLI straightforward extensions can be defined, assuming a rank ordering that works with an penalty function criterion that always ranks feasible solution higher than infeasible solutions. For the integral criteria PoI and ExI expressions have been taken from literature, and for the first time employed within the context of evolutionary optimiza-

tion. Last but not least, also the interval based pre-selection criteria have been extended in a straightforward manner.

The test runs show, that the metamodel-assisted strategies with lb_ω filter seems to perform best in the long run, while in the first iterations it seems to better to work with the mean value criterion. In the test runs we conducted the MAES with the PoI criterion performed slightly better than the strategies with other filters that take into account the confidence measure. However, there seems to be no significant difference between the criteria ExI, PoI and the lower confidence bound criterion. Since the ExI and PoI criteria do not ask for a user specified parameter ω, these pre-selection methods have a slight advantage.

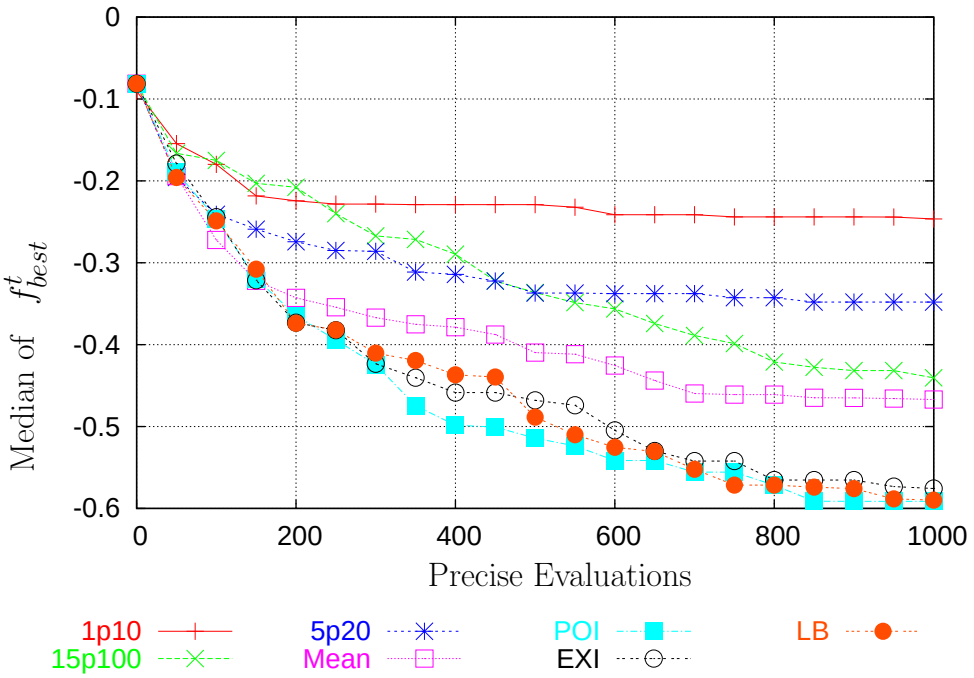


Figure 5.6.4: Median for best found feasible function values for different strategies on the multimodal and constrained 10-D Keane bump problem (20 runs, (15+100)-MAES)).

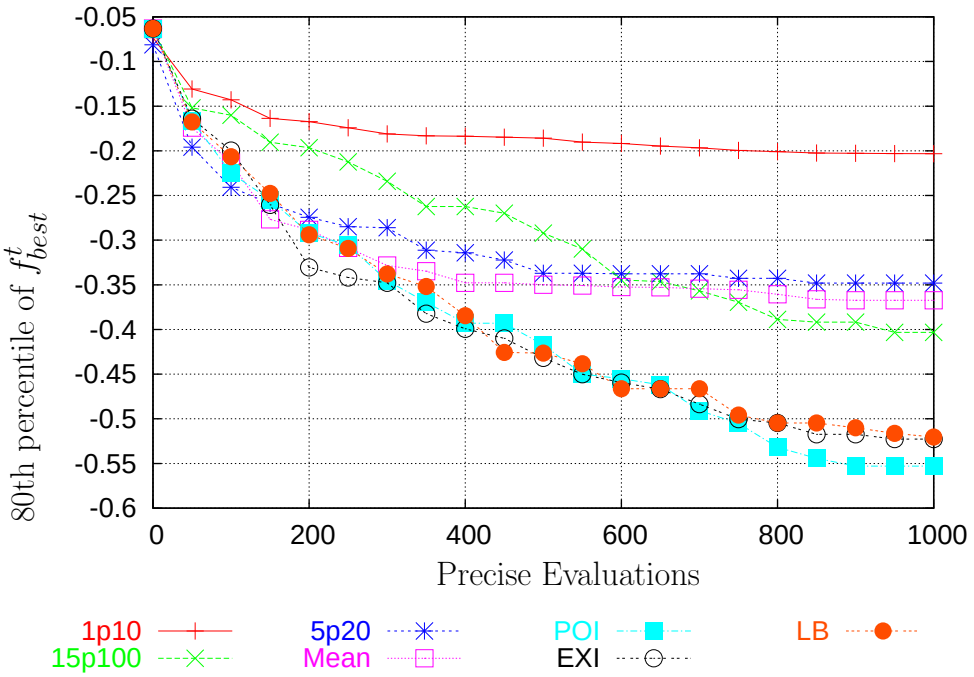


Figure 5.6.5: 80%-quantiles for best found feasible function values for different strategies on the multimodal and constrained 10-D Keane bump problem (20 runs, $(15 + 100)$-MAES).

6 Multi-objective optimization

People talk about the middle of the road as though it were unacceptable. Actually, all human problems, excepting morals, come into the gray areas. Things are not all black and white. There have to be compromises. The middle of the road is all of the usable surface.

D. Eisenhower

As in life, also in design optimization, we often have to deal with conflicting objectives. Optimization tools may not enable the decision maker(s) to dissolve conflicts, and thus save us from the struggle for a good compromise solution. However, so-called Pareto optimization tools can help to avoid lose-lose decisions, and to gain insights into the structure of the trade-off curve or surface comprising the interesting alternatives. Thus Pareto optimization can be a valuable tool for the decision maker.

Evolutionary multi-objective optimization algorithms (EMOA) will be considered as solution methods for this problem domain. We will focus on two versions of EMOA in order to discuss the integration of metamodeling techniques. These are the NSGA-II algorithm and the $\mathcal{S}$ metric selection EMOA (SMS-EMOA). With the latter, a new approach will be suggested that is well-suited for design optimization and for the generalization of metamodeling techniques.

After a brief introduction to the problem of Pareto optimization (section 6.1), we introduce two selected EMOA. The first one (section 6.2) is the well-established NSGA-II algorithm and the second one (section 6.3) is a recently proposed algorithm, the SMS-EMOA. The SMS-EMOA has been chosen, because it allows for an elegant integration of metamodel-assistance. Moreover, it yielded in superior results on standard benchmarks. Since the SMS-EMOA is a new algorithm, it is discussed in more detail and we compare it to established EMOA. Later, in chapter 7, metamodel-assisted versions of these two algorithms will be proposed.

6.1 Introduction into Pareto optimization

Pareto optimization [CVL02, Deb01, Zit99] has become an established technique for detecting interesting solution candidates for multi-objective optimization problems. It enables the decision maker to extract efficient solutions from the set of all possible solutions and to discover trade-offs between opposing objectives among these solutions.

In Pareto optimization vectors of objective function values are compared by a preference

relation that defines a partial order on the solution space. Given a problem with multiple objectives (to be minimized[1]) and no constraints ($n_f > 1, n_g = 0$), the preference relation can be defined for arbitrary solution vectors $\mathbf{y} \in \mathbb{R}^{n_f}$ and $\mathbf{y}' \in \mathbb{R}^{n_f}$:

$$\begin{aligned} \mathbf{y} \prec_p \mathbf{y}' (\text{say: } \mathbf{y} \text{ (pareto)-dominates } \mathbf{y}') :\Leftrightarrow \\ \forall i \in \{1, \dots n_f\} : \; y_i \leq y'_i \quad \wedge \\ \exists i \in \{1, \dots n_f\} : \; y_i < y'_i. \end{aligned} \tag{6.1.1}$$

According to expression 5.2.5, we can easily extend this definition to the constrained case ($n_g > 0$). Let $\mathbf{y} \in \mathbb{R}^{n_f+n_g}$ and $\mathbf{y}' \in \mathbb{R}^{n_f+n_g}$ denote two arbitrary solution vectors. Their first n_f positions denote objective function values and the last n_g positions the values of the constraint functions. Moreover, let $\mathbf{y}_f := (y_1, \dots, y_{n_f})^T$, $\mathbf{y}_g := (y_{n_f+1}, \dots, y_{n_f+n_g})^T$, $\mathbf{y}'_f := (y'_1, \dots, y'_{n_f})^T$ and $\mathbf{y}'_g := (y'_{n_f+1}, \dots, y'_{n_f+n_g})^T$. Then

$$\begin{aligned} \mathbf{y} \prec \mathbf{y} :\Leftrightarrow \\ \mathbf{y}_g \leq 0 \; \wedge \; \mathbf{y}'_g \leq 0 \; \wedge \; \mathbf{y}_f \prec_p \mathbf{y}'_f \; \vee \\ \mathbf{y}_g \leq 0 \; \wedge \; \mathbf{y}'_g > 0 \quad \vee \\ \mathbf{y}_g > 0 \; \wedge \; \mathbf{y}'_g > 0 \; \wedge \; \delta(\mathbf{y}_g) < \delta(\mathbf{y}'_g). \end{aligned} \tag{6.1.2}$$

Here, $\delta : \mathbb{R}^{n_g} \rightarrow \mathbb{R}_0^+$ denotes a metric penalty function (cf. expression 5.2).

For notational convenience, we also define a preference relation on the search space $\mathbb{S}$. Let $\mathbf{x}$ and $\mathbf{x}'$ denote two solutions in $\mathbb{S}$. Then

$$\mathbf{x} \prec \mathbf{x}' :\Leftrightarrow \mathbf{y}(\mathbf{x}) \prec \mathbf{y}(\mathbf{x}') \tag{6.1.3}$$

Given a set R of search points, the non-dominated subset $\text{nd}(R)$ of R is defined as[2]:

$$\text{nd}(R) = \{\mathbf{x} \in R | \; \nexists \mathbf{x}' \in R : \mathbf{x}' \prec \mathbf{x}\} \tag{6.1.4}$$

The aim in Pareto optimization is to detect the *pareto-optimal set*, defined as $\text{nd}(\mathbb{S})$, for a search space $\mathbb{S}$, or at least a good approximation to this set. In contrast to the pareto-optimal set, we define the *Pareto front* as the set of all solutions that correspond to points in the pareto-optimal set.

According to Deb [Deb01], a good approximation to the Pareto front is an approximation that covers a large portion of the Pareto front. In practice, the decision maker often wishes to evaluate only a limited number of pareto-optimal solutions. Typically these solutions include extremal solutions as well as solutions that are located in parts of the solution space, where good compromise solutions can be found.

[1] By simply inverting the sign of objective functions, maximization problems can be transformed into minimization problems.

[2] Note, that this order is not a partial order, because the antisymmetry axiom does not hold.

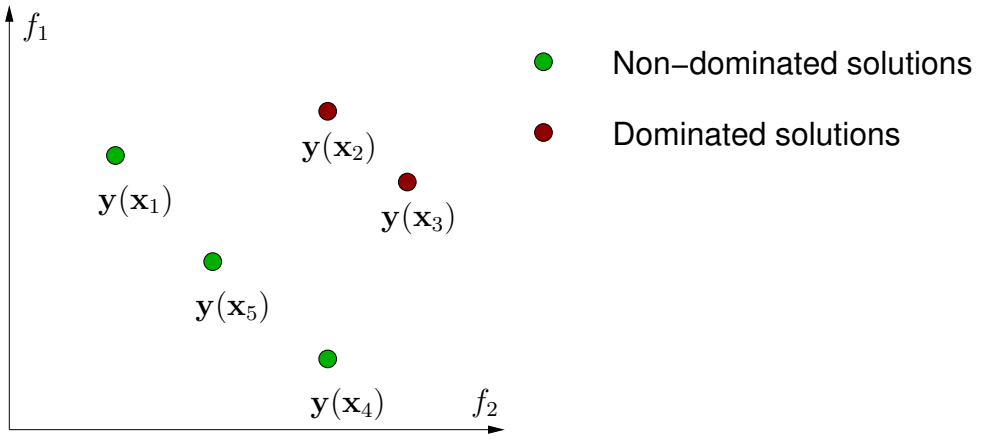


Figure 6.1.1: Dominance relations in multi-objective minimization: The solution vectors for some solution set $R = \{\mathbf{x}_1, \ldots, \mathbf{x}_5\}$ are depicted in a two-objective solution space spanned by f_1 and f_2, that denote the objective functions to be minimized. The set $\{\mathbf{x}_1, \mathbf{x}_4, \mathbf{x}_5\}$ is the non-dominated subset $\mathrm{nd}(R)$ of R.

In the context of Pareto optimization, the following definitions are also useful: A solution is said to be non-dominated by a set of solution, iff no solution in this set dominates the solution:

$$\mathbf{x} \preceq P :\Leftrightarrow \nexists \mathbf{x}' \in P : \mathbf{x}' \prec \mathbf{x}. \tag{6.1.5}$$

Two solutions $\mathbf{x}$ and $\mathbf{x}'$ are said to be *incomparable*, iff neither $\mathbf{x} \prec \mathbf{x}'$ nor $\mathbf{x}' \prec \mathbf{x}$. Furthermore, we will define the *solution set* Y as the co-domain of $\mathbf{y}$, i. e.

$$Y = \{\mathbf{y}(\mathbf{x}) | \mathbf{x} \in \mathbb{S}\}. \tag{6.1.6}$$

Example: As an example, Figure 6.1.1 depicts the ranking of a small set $R = \{\mathbf{x}_1, \ldots, \mathbf{x}_5\}$ of solutions by means of the Pareto preference relation. Solutions $\mathbf{x}_1$, $\mathbf{x}_4$, and $\mathbf{x}_5$ are non-dominated. Solution $\mathbf{x}_2$ is dominated by $\mathbf{x}_1$, $\mathbf{x}_4$, and $\mathbf{x}_5$, while $\mathbf{x}_3$ is only dominated by $\mathbf{x}_4$ and $\mathbf{x}_5$. The non-dominated solutions $\mathbf{x}_1$, $\mathbf{x}_4$, and $\mathbf{x}_5$ are mutually incomparable, as well as solutions $\mathbf{x}_2$ and $\mathbf{x}_3$.

The characteristics of the Pareto preference relation and the possible geometrical structure of the Pareto front for different numbers of objectives have been the subjects of theory for long. For a comprehensive introduction, the interested reader is referred to [Mie99]. A more practical introduction, focussing on aspects that are important in the context of EMOA, can be found in [Deb01] and in [CVL02].

6.2 Evolutionary multi-objective optimization

Several algorithms have been suggested for the approximation of Pareto fronts. Among them, evolutionary multi-objective optimization algorithms (EMOA) became increasingly popular, because they are considered to be robust and their design is flexible, meaning that they can be applied for different representations and adapted to different (parallel) computing environments. The elaboration of EMOA is a subject of ongoing

research. However, some state-of-the-art algorithmic variants have established in recent years. Among them are the strength Pareto evolutionary algorithm (SPEA) [ZLT01], Pareto evolution strategy algorithm (PESA) [KCF03], and the *non-dominated sorting genetic algorithm* (NSGA) [DAPM00a, DAPM00b, DPAM02]. The more recent variant of the latter approach, termed NSGA-II [DPAM02], has been frequently used for design optimization and features the $(\mu + \lambda)$-selection that is also used in evolutionary strategies. Thus, the NSGA-II algorithm was chosen as the starting point of our discussion on metamodel-assisted EMOA.

Like other EMOA, the NSGA-II aims at detecting a well-distributed set of solutions close to the Pareto front. This is achieved by using a special selection procedure within a $(\mu + \lambda)$-EA. In order to achieve a good convergence to the Pareto front, non-dominated solutions are always ranked higher than dominated solutions. Moreover, by means of the so-called *non-dominated sorting* procedure, a rank is assigned to each of the solutions in a population, expressing the degree of non-dominance of these solutions. The non-dominated sorting procedure works as follows: First the set of non-dominated solutions R_1 among the solutions in a population is determined. All members of this subset are assigned to the first rank. From the remaining set $R \setminus R_1$, the set of non-dominated solutions R_2 is detected and its members are assigned to the second rank. This procedure is repeated, until the whole population is subdivided into partitions $R_1, \ldots, R_\ell$, each of which members are assigned to ranks $1, \ldots, \ell$.

Obviously, there can be more than one element in one of the partitions. In order to establish a total ranking among the elements of a particular partition, *crowding distance* sorting is used. This sorting procedure assigns higher ranks to elements that contribute more to the diversity of the given set. In order o determine the crowding distance of an individual, first the distance to the nearest neighboring solutions is determined in each positive and negative coordinate direction. Here, the coordinates are always sought as the coordinates of the solution vectors. In order to calculate a scalar value, all these distances are summed up and the resulting value is the crowding distance. Extremal solutions, that have no neighboring solutions in at least one of the coordinate directions, are always preferred to non-extremal solutions.

Example: In figure 6.2.2 and figure 6.2.3 the sorting procedure of NSGA-II is illustrated for a two-objective minimization problem. Figure 6.2.2 displays the three partitions detected by the non-dominated sorting procedure, while figure 6.2.3 displays the ranking of the first partition due to crowding distance sorting.

Once unique ranks for the elements in a population have been determined, the $(\mu + \lambda)$ selection can be employed in the usual way (cf. subsection 3.5.1). The resulting NSGA-II algorithm will be termed $(\mu+\lambda)$-NSGA-II. Typically, as proposed in [Deb01], the $(\mu+\mu)$ selection is used in the NSGA-II.

6.3 The $\mathcal{S}$-metric selection

Though the NSGA-II selection method provides a unique ranking for a set of solution, it does not provide an intuitive, scalar measure of improvement. Such a measure would be desirable, for example, if we want to achieve a straightforward generalization of some

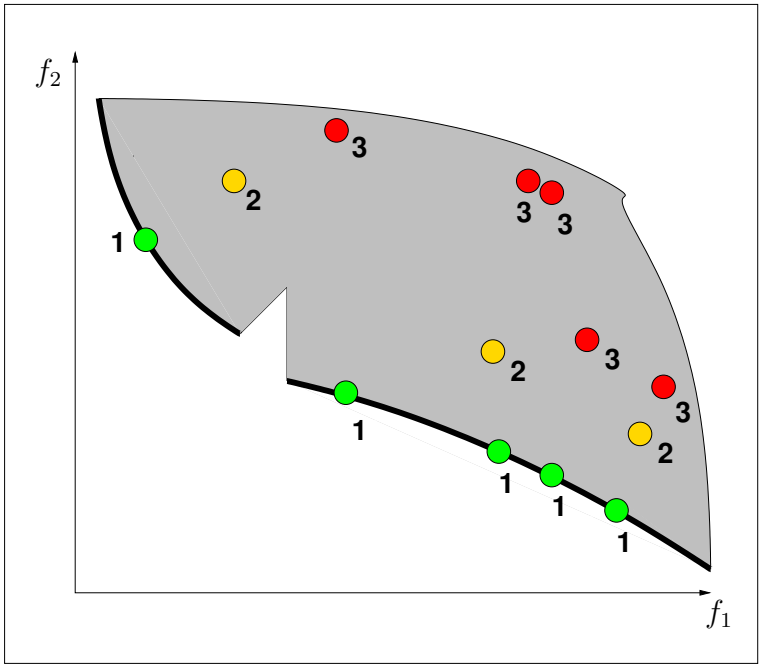


Figure 6.2.2: Illustration of the non-dominated sorting procedure for a population of 13 individuals in a two dimensional solution space. The population gets subdivided into three partitions. The numbers attached to the solutions indicate the rank that was assigned to them by the non-dominated sorting.

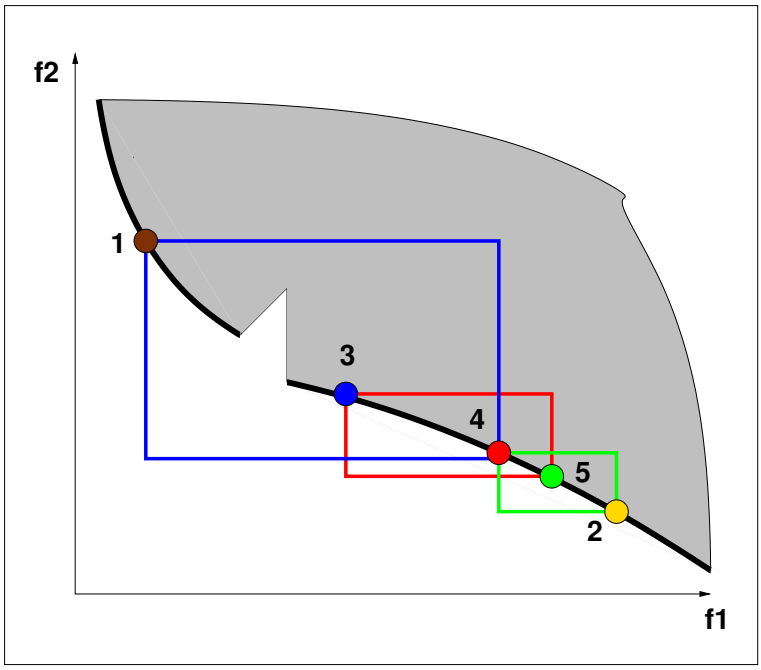


Figure 6.2.3: Illustration of the crowding distance sorting of members of the non-dominated subset of the population depicted in figure 6.2.2. The circumference of the boxes touching neighboring solutions are used as ranking criterion. Extremal solutions are always ranked better than non-extremal solutions. The numbers assigned to the solutions indicate their rank.

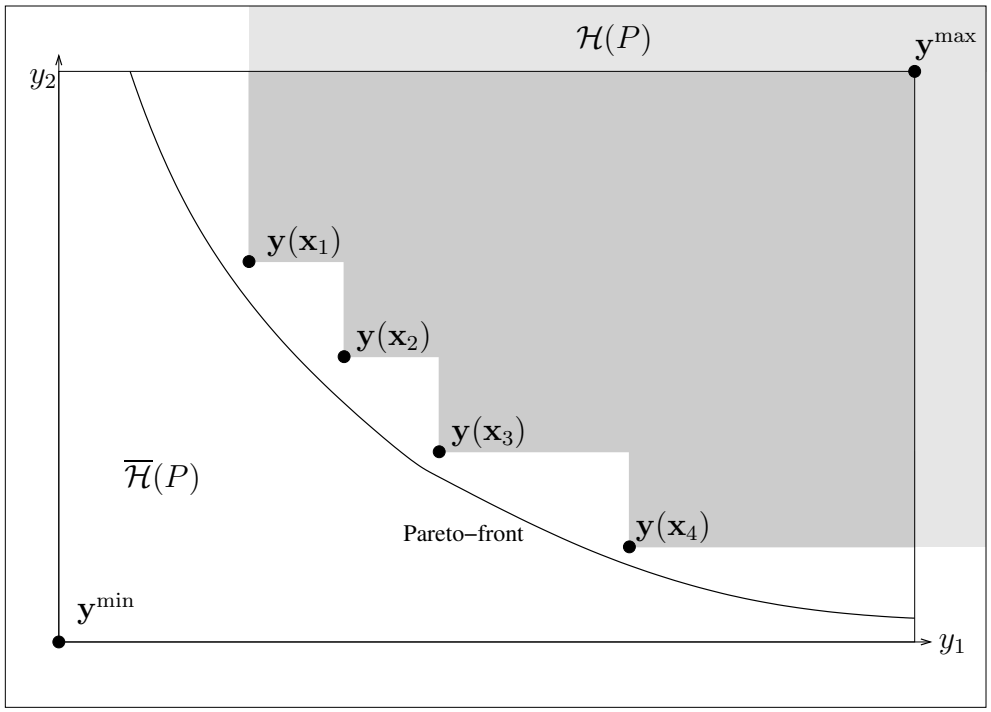


Figure 6.3.4: Illustration of the hypervolume measure for a two dimensional solution space. The gray area indicates the subspace $\mathcal{H}(P)$ dominated by the solution set $A = \{\mathbf{x}_1, \ldots, \mathbf{x}_4\}$ and $\overline{\mathcal{H}}(P)$ the non-dominated subspace. Given a reference point $\mathbf{y}^{\max}$ in the solution space, the Lebesgue measure Λ of $\bigcup_{\mathbf{y}\in A}\{\mathbf{y}'|\mathbf{y} \prec \mathbf{y}' \prec \mathbf{y}^{\max}\}$ is termed the hypervolume measure of A, in brief $\mathcal{S}(A)$. In the figure, the part of the dominated subspace that is filled dark gray indicates the subset of $\mathcal{H}$, the area of which determines $\mathcal{S}(A)$.

of the pre-screening criteria introduced in 4.2. Hence, a method was sought that is based on a measure of improvement. The method that was found, namely the $\mathcal{S}$ metric selection EMOA (SMS-EMOA), turned out to be very powerful, even without making use of metamodel-assistance. The remainder of this chapter provides a detailed analysis of this new algorithm, which forms the basis of the metamodel-assisted EMO that will be introduced in chapter 8.

The $\mathcal{S}$ metric selection EMOA (SMS-EMOA) has been recently proposed by Emmerich, Beume and Naujoks [EBN05]. It uses the hypervolume measure – a scalar criterion for the quality of an Pareto front approximation – as a criterion for comparing solutions of the same dominance rank. The SMS-EMOA is especially designed for approximating a small, well-distributed set of pareto-optimal solutions. It also allows for the straightforward integration of metamodeling techniques, in particular of those techniques based on integral expressions like the probability of improvement and the expected improvement. In this section, we first outline the new algorithm and compare it conceptually to the NSGA-II. Later, the SMS-EMOA shall be compared on a set of benchmark problems to other EMOA, including NSGA-II.

6.3.1 The hypervolume measure

The hypervolume measure (or $\mathcal{S}$ metric) was originally proposed by Zitzler and Thiele [ZT98], who called it the *size of dominated space* [Zit99]. Later, Fleischer [Fle03] defined it as the Lebesgue measure Λ of the union of hyper-rectangles defined by a set of non-dominated solution vectors A and a reference solution vector $\mathbf{y}^{\max}$ that is dominated by all solution vectors in A:

$$\mathcal{S}(A) := \Lambda(\bigcup_{\mathbf{y} \in A} \{\mathbf{y}' | \mathbf{y} \prec \mathbf{y}' \prec \mathbf{y}^{\max}\}).$$

An illustration of the hypervolume metric for a solution set of a problem with two objectives is given in figure 6.3.4.

Given a solution set $R = \{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$ with $\mathbf{x}_i \in \mathbb{S}$, $i = 1, \ldots, m$, for notational convenience, we will sometimes write $\mathcal{S}(\{\mathbf{x}_1, \ldots, \mathbf{x}_m\})$ instead of $\mathcal{S}(\{\mathbf{y}(\mathbf{x}_1), \ldots, \mathbf{y}(\mathbf{x}_m)\})$, provided that it is clear to which objective function we refer.

The ranking of different sets of solutions by means of the $\mathcal{S}$ metric is influenced by the reference point. Generally, the ranking of different sets by means of the $\mathcal{S}$ metric is sensitive to its choice. Knowles and Corne [KC02, KC03] gave an example with two Pareto fronts, A and B, in the two dimensional case. They showed either $\mathcal{S}(A) < \mathcal{S}(B)$ or $\mathcal{S}(B) < \mathcal{S}(A)$ depending on the choice of the reference point. Nevertheless, the $\mathcal{S}$ metric was used in several comparative studies of EMOA, e.g. [DMM03b, DMM03a, Zit99]. Note that the contribution of the extremal points to the hypervolume measure depends on the choice of the reference point.

However, until recently, the hypervolume has never been used as a selection criterion. Fleischer [Fle03] attributes this to the fact that the time consumption of existing algorithms for computing the hypervolume measure scales exponentially. The operational time complexity of the recursive procedure described by Knowles and Corne [KC03] was $O(k^{n_f+1})$ with k being the number of solutions in the non-dominated subset of the population and n_f being the number of objectives (dimensionality of the solution space). Though it seems that for high dimensions the computation of the hypervolume seems to be expensive, for moderate dimensions and small populations its computation can be affordable, especially in the context of time consuming evaluations.

Fleischer [Fle03] proved that the detection of a set that maximizes $\mathcal{S}$ is equivalent to the detection of the pareto-optimal set for any finite search space. On the basis of these new results, Fleischer suggested to recast the multi-objective optimization problem as a single-objective optimization problem and to employ single-objective optimization algorithms for finding a set of solutions that maximizes the $\mathcal{S}$ metric.

Knowles and Corne also recognized the advantages of using the hypervolume measure as integral part of multi-objective optimization algorithms. In [KC03] suggested an adaptive archiving strategy, they termed $\text{AA}_{\mathcal{S}}$, based on the hypervolume measure. It processes a sequence of solutions by trying to integrate them one by one into an archive with bounded size, thereby possibly discarding solutions from the archive or the currently processed solution. The adaptive archiver can be employed to process the sequence of solutions generated by an EMOA. The update of the archive is described in algorithm 8 and 9: Given an archive A_t and a new solution $\mathbf{x}$, the new archive A_{t+1} is given by the

Algorithm 8 $\mathrm{AA}_{\Delta\mathcal{S}}$.

$P_0 \leftarrow \oslash$ {Initialize archive}
$t \leftarrow 0$
repeat
  $\mathbf{x}_{t+1} \leftarrow$ **get**() {Get next individual from stream of solutions}
  $P_{t+1} \leftarrow$ **reduce**$_{\Delta\mathcal{S}}(P_t \cup \{\mathbf{x}_{t+1}\})$ {Select a maximum of μ individuals for the new archive}
  $t \leftarrow t + 1$
until stop criterium reached

Algorithm 9 **reduce**$_{\Delta\mathcal{S}}(Q)$.

$Q' \leftarrow \mathrm{nd}(Q \cup \{\mathbf{x}\})$ {reduce to non-dominated subset}
if $|Q'| = \mu + 1$ **then**
  for all $\mathbf{x} \in Q'$ **do**
    $\Delta_{\mathcal{S}}(\mathbf{x}, R_\ell) \leftarrow \mathcal{S}(R_\ell) - \mathcal{S}(R_\ell \setminus \{\mathbf{x}\})$
  end for
  $\mathbf{x} \leftarrow \arg\min_{\mathbf{x} \in Q'}[\Delta_{\mathcal{S}}(\mathbf{x}, Q')]$ {detect element of Q' with lowest $\Delta_{\mathcal{S}}(\mathbf{x}, R_\ell)$}
  $Q'' \leftarrow Q' \setminus \{\mathbf{x}\}$ {eliminate detected element}
else
  $Q'' \leftarrow Q'$
end if
return Q''

non-dominated subset of $A_t \cup \{\mathbf{x}\}$, or, if the size of this set exceeds the maximum bound μ, the subset of size μ of $A_t \cup \{\mathbf{x}\}$ that covers the maximal hypervolume. This subset is computed in the **reduce**$_{\Delta\mathcal{S}}$ procedure by eliminating the element the hypervolume of which contributes least to the hypervolume. A polynomial time implementation of this update procedure for the general multi-objective problem was proposed by Knowles, Corne and Fleischer [KCF03]. Later, in subsection 6.3.5, a simplified version of this procedure for the two-objective case is devised.

6.3.2 $\mathcal{S}$ metric selection

Emmerich, Beume, and Naujoks [EBN05] proposed a new selection operator for EMOA by combining the archiving procedure by Knowles and Corne [KC03] with the non-dominated sorting procedure by Deb [Deb01]. The resulting steady-state EMOA was termed SMS-EMOA. Instead of maintaining an archive that is separate from the EA the SMS-EMOA directly employs the hypervolume measure to decide whether individuals are selected in the replacement or not.

Algorithm 10 describes the generational loop of the SMS-EMOA. There is not much of a difference to a standard $(\mu+1)$-EA, except that the **replace**$_{\Delta\mathcal{S}}$ procedure was introduced as a replacement operator. This procedure is described in Algorithm 11. First, the partitions of the population $P_t \cup \{\mathbf{x}_{t+1}\}$ with respect to the non-domination level are computed using the fast non-dominated sorting algorithm by Deb et al. [DAPM00b]. Afterwards, one individual is discarded from the worst ranked front. Whenever this front

Algorithm 10 SMS-EMOA.

1: $P_0 \leftarrow$ **initialize**() {Initialize random start population of μ individuals}
2: $t \leftarrow 0$
3: **repeat**
4: $\quad \mathbf{x}_{t+1} \leftarrow$ **generate**(P_t) {Generate one offspring by variation operators}
5: $\quad P_{t+1} \leftarrow$ **replace**$_{\Delta\mathcal{S}}(P_t \cup \{\mathbf{x}_{t+1}\})$ {Select μ individuals for the new population}
6: $\quad t \leftarrow t+1$
7: **until** stop criterium reached

Algorithm 11 **replace**$_{\Delta\mathcal{S}}(Q)$.

1: $\{R_1, .., R_\ell\} \leftarrow$ **non-dominated-sort**(Q) {all ℓ partitions of Q in increasing order}
2: **for all** $\mathbf{x} \in R_\ell$ **do**
3: $\quad \Delta_{\mathcal{S}}(\mathbf{x}, R_\ell) \leftarrow \mathcal{S}(R_\ell) - \mathcal{S}(R_\ell \setminus \{\mathbf{x}\})$
4: **end for**
5: $\mathbf{x} \leftarrow \arg\min_{\mathbf{x} \in R_\ell}[\Delta_{\mathcal{S}}(\mathbf{x}, R_\ell)]$ {detect element of R_ℓ with lowest $\Delta_{\mathcal{S}}(\mathbf{x}, R_\ell)$}
6: $Q' \leftarrow Q \setminus \{\mathbf{x}\}$
7: **return** Q'

comprises $m_\ell > 1$ individuals, the individual $\mathbf{x} \in \mathcal{R}_\ell$ that minimizes

$$\Delta_{\mathcal{S}}(\mathbf{x}, R_\ell) := \mathcal{S}(R_\ell) - \mathcal{S}(R_\ell \setminus \{\mathbf{x}\}) \tag{6.3.7}$$

is eliminated. Thereby, it is guaranteed that the subset of size $m_\ell - 1$ of R_ℓ remains in the population that covers the maximal hypervolume compared to all m_ℓ possible subsets (for a proof we refer to Knowles et al. [KC03]). With regard to the replacement operator this also implies that the covered hypervolume of the population P_t cannot decrease by application of **replace**$_{\Delta\mathcal{S}}$, i. e. for algorithm 10 we can state the invariant:

$$\mathcal{S}(P_t) \leq \mathcal{S}(\textbf{replace}_{\Delta\mathcal{S}}(P_t \cup \{q_{t+1}\})). \tag{6.3.8}$$

6.3.3 Theoretical characteristics of the $\mathcal{S}$ metric selection

The algorithm presented is very similar to the adaptive archiver $\mathrm{AA}_{\mathcal{S}}$ presented by Knowles et al. [KC03]. However, there is an important difference between instantiations of $\mathrm{AA}_{\mathcal{S}}$ and the SMS-EMOA: Since $\mathrm{AA}_{\mathcal{S}}$ is a class of archivers, non-dominated solutions are always discarded. This is not the case for the SMS-EMOA. In the latter algorithm the population provides the basis for variations obtained with the variation operators. In order to provide a good diversity of solutions for the variation procedures, and thus avoid stagnation, too small population sizes should be avoided. This is achieved by keeping the population size constant, even for the price of accepting dominated solutions.

Nevertheless, some theoretical characteristics of $\mathrm{AA}_{\mathcal{S}}$ are inherited by the SMS-EMOA. The first one is the fact that the SMS-EMOA produces a series of populations with $\mathcal{S}(P_{t+1}) \leq \mathcal{S}(P_t)$. It has been proven, that this characteristic is sufficient to prove that the SMS-EMOA converges in probability to non-dominated solutions in $\mathrm{nd}(\mathbb{S})$, provided that the search operators generate each solution in $\mathbb{S}$ with a finite probability.

Another characteristic that is preserved from the $AA_{\mathcal{S}}$ algorithm is, that the SMS-EMOA converges in probability to a local optimum for the optimization problem. A local optimum means that no replacement of a solution in P_t by any other solution in $\mathbb{S}$ would yield in an improvement with regard to $\mathcal{S}(P_{t+1})$. By means of empirical investigations, Knowles et al. [KC03] came to the conclusion that the points of local optima of the $\mathcal{S}$-Metric are "well distributed". They used the term "well distributed" in an intuitive way, meaning that a set of solutions covers interesting regions of the Pareto front. Further support for their appreciation is provided in subsection 6.3.4.

A common objection to the hypervolume measure is that it critically depends on the choice of the reference point and the scaling of the search space. The particular hypervolume measure for a set of points actually depends on the distance to the reference point and its position relative to the entire solution space. It might also turn out that the reference point gets infeasible. This would be the case, if a point is found during the course of optimization that does not dominate the reference point. To circumvent these problems, in the SMS-EMOA an infinite reference point

$$\mathbf{y}^{\max} = \underbrace{(\infty, \ldots, \infty)}_{n_f \text{ times}} \tag{6.3.9}$$

is chosen by default. Of course, this decision entails that there is no more comparable value for $\mathcal{S}(M)$ anymore for any set of solutions M, because this value turns out to be infinite. However, the increase in hypervolume $\mathcal{S}(M \cup \{\mathbf{x}\}) - \mathcal{S}(M)$ can well take a finite value, if $\mathbf{x}$ is non-extremal. Here, an extremal solution is sought as an solution for that at least one of its objective function values takes its minimum w.r.t. an entire solution set. Since there is no meaningful value for $\Delta_{\mathcal{S}}$ for extremal solutions, it has been suggested in [EBN05] that extremal solutions are always ranked best, like it is done in crowding distance sorting.

Furthermore, the SMS-EMOA method is independent from the scaling of the objective space, in the sense that the order of solutions is not changed by multiplying the objective functions with a constant scalar vector. This is obvious since our metric value itself is a sum of products.

6.3.4 Comparison of the difference in hypervolume to the crowding distance

At first sight, the conceptual design of the SMS-EMOA looks very similar to that of the NSGA-II. However, there are some important differences. First of all, the NSGA-II typically works with the $(\mu + \mu)$ selection, whereas the SMS-EMOA employs the $(\mu + 1)$ selection. Another difference that deserves attention is the measure employed to compare solutions of the same rank of non-dominance, namely the crowding distance and the increase in hypervolume.

Let us now compare the crowding distance measure, that functions as ranking criterion for solutions of equal Pareto rank in NSGA-II, to the hypervolume based measure $\Delta_{\mathcal{S}}$ that gets employed for ranking solutions in the SMS-EMOA. Recall, the crowding distance was chosen for the purpose to distribute solution points uniformly on the Pareto front.

In contrast to this, selection by means of the $\Delta_{\mathcal{S}}$ criterion distributes them in a way that maximizes the covered hypervolume. A solution set distributed in the latter way will provide a better result for the practitioner, since it concentrates the solution in regions of the Pareto front where good compromise solutions are found without losing extremal points.

In order to further clarify this appreciation, let us discuss the example illustrated in figure 6.2.3. Here, a set P of non-dominated solutions is depicted in a two dimensional solution space. The left hand side figure shows the lines, the lengths of which contribute to the sum for the crowding distance and thus determine the ranking of the solutions in the NSGA-II. The right hand side figure depicts the same solutions, and their corresponding value of $\Delta_{\mathcal{S}}$, which is given by the area of the attached rectangles. Note that for the crowding distance the value of a solution $\mathbf{x}_i$ depends on its neighbors and not directly on the position of the point itself, in contrast to $\Delta_{\mathcal{S}}(\mathbf{x}, P)$. In both cases the extremal solutions are ranked best, provided we choose a sufficiently large reference point for the $\mathcal{S}$ metric. Concerning the inner points of the front, $\mathbf{x}_5$ outperforms $\mathbf{x}_4$ in figure 6.3.5, if the crowding distance is used as a ranking criterion. On the other hand, $\mathbf{x}_4$ outperforms $\mathbf{x}_5$, if $\Delta_{\mathcal{S}}$ gets employed. This indicates that good compromise solutions, that are located near knee-points of convex parts of the Pareto fronts are given better ranks in the SMS-EMOA than in the NSGA-II algorithm. Practically, solution x_5 is less interesting than solution x_4, since in the vicinity of x_5 little gains in objective f_2 can only be achieved at the price of large concessions with regard to objective f_1. Thus, the new approach might lead to more interesting solutions. This is of particular importance, if the decision maker is only interested in a small, limited number of solutions in the pareto-optimal set.

6.3.5 Implementation

Before the SMS-EMOA is studied by means of computer experiments, let us spend a few more words on its implementation. Fleischer's algorithm could be used for the determination of the hypervolume metric in any dimension. However, in two and three dimensions more efficient procedures can be found, that will be outlined next.

Two objective functions

For the case of two objective functions, a straightforward procedure for $\Delta_{\mathcal{S}}(\mathbf{x}, R_\ell)$ (cf. algorithm 11) can be implemented to determine the solution which contributes least to the current hypervolume of the entire set of points. We take the points of the worst-ranked partition (due to non-dominated sorting) and sort them ascending concerning their value of the first objective function f_1 and get a sequence that is additionally sorted in descending order concerning the f_2 values, because the points are mutually non-dominated. In the two objective case, the dominated area of each solution is shaped like a rectangle. The difference concerning f_1 between a point $\mathbf{x}_i$ and its successor $\mathbf{x}_{i+1}$ in the sorted sequence represents the width of this rectangle while the distance of point $\mathbf{x}_i$ to its predecessor $\mathbf{x}_{i-1}$ in the f_2 values equals its height (see figure 6.2.3, right). The product of these two differences equals $\Delta_{\mathcal{S}}$, the hypervolume only dominated by solution $\mathbf{x}_i$. This is actually the rectangle spanned by the point $\mathbf{x}_i$ and the corner of the Pareto front generated from point $\mathbf{x}_{i-1}$ and $\mathbf{x}_{i+1}$ without point $\mathbf{x}_i$.

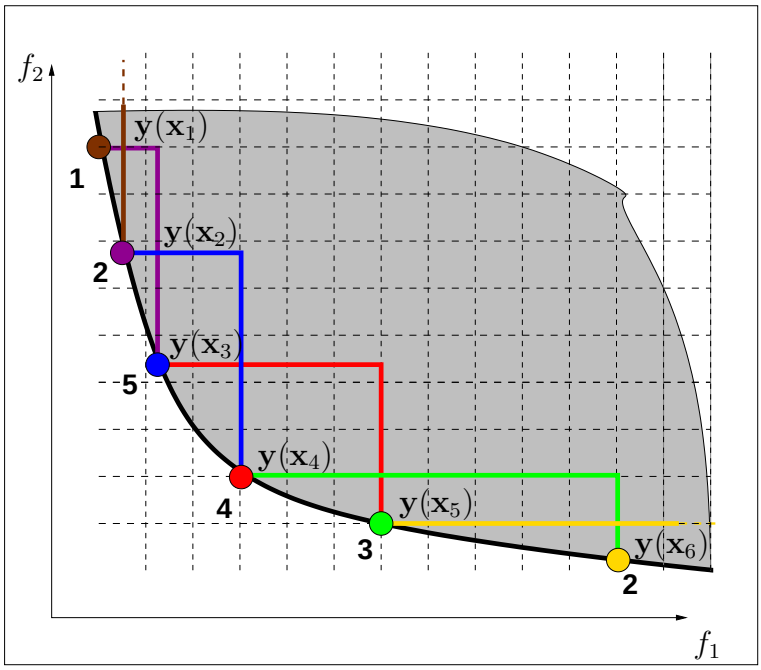


Figure 6.3.5: Ranking due to crowding distance.

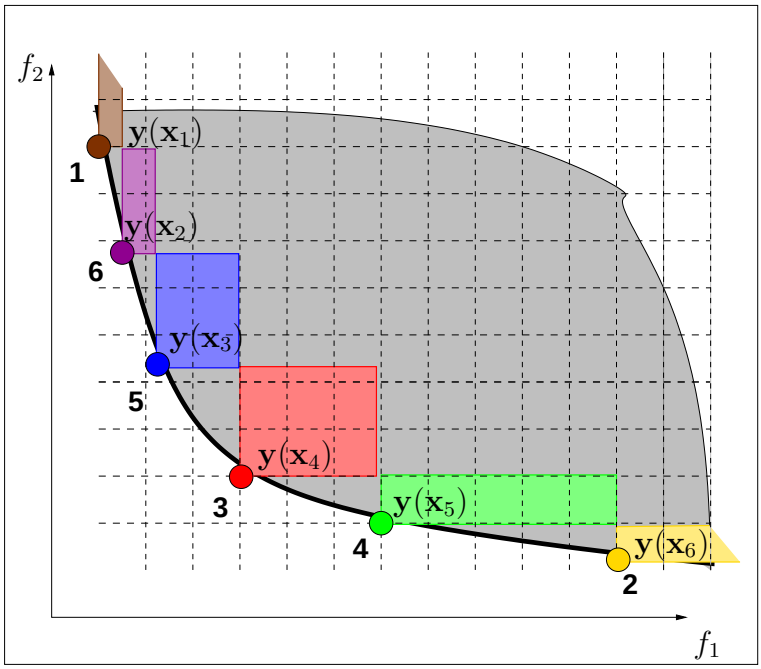


Figure 6.3.6: Ranking due to hypervolume measure of solutions on the Pareto front. The circumference of the boxes touching neighboring solutions are the ranking criterion. The numbers assigned to the solutions indicate their rank.

Algorithm 12 Hypervolume(R_ℓ) for $n_f = 2$.

1: $l \leftarrow |R_\ell|$ {number of solutions on the worst ranked partition due to non-dominated sorting.}
2: $R_\ell \leftarrow \text{sort}(R_\ell, f_1)$ {sort elements of R concerning their f_1 value}
3: $\Delta_\mathcal{S}(R_\ell[1]) \leftarrow \infty; \Delta_\mathcal{S}(R_\ell[l]) \leftarrow \infty$ {boundary points are always kept}
4: **for all** $i \in \{2, ..., l-1\}$ **do**
5: {for all inner points of the front}
6: $\Delta_\mathcal{S}(R_\ell[i]) \leftarrow (R_\ell[i+1].f_1 - R_\ell[i].f_1) \cdot (R_\ell[i-1].f_2 - R_\ell[i].f_2)$
7: {rectangle of the hypervolume only dominated by $R_\ell[i]$}
8: **end for**

The runtime complexity of the hypervolume procedure in the case of two objective functions is governed by the sorting algorithm and so is $\mathcal{O}(\mu \cdot \log\mu)$ if all points lie on the same non-dominated front. As in the case with two objective functions, the hypervolume of a new point can only influence that of the two neighboring solutions in the population and since only these points have to be updated. One can decrease the runtime by using a suitable data structure.

Three objective functions

The computation of $\Delta_\mathcal{S}(R_\ell[i])$ gets significantly more complex for three objective functions, since the removal of a node can have a non-local influence on the dominated hypervolume. A simple way to deal with this problem is to use existing algorithms for the computation of the dominated hypervolume. For three dimensions the best known algorithm [Fle03] has a running time of $\mathcal{O}(k^3)$. Thus, the repeated computation of $\Delta_\mathcal{S}(R_\ell[i])$ for each point $i \in \{1, \ldots, k\}$ would have a running time $\mathcal{O}(k^4)$. Next, with algorithm 13, we propose a new algorithm that computes $\Delta_\mathcal{S}(R_\ell[i])$ with a runtime complexity $\mathcal{O}(k^3)$.

A first intuition was to look on the dominated set from a bird's eye perspective ($-f_3$ is considered to be the height) and to investigate separately k^2 cells, the coordinates of which are given by first and the second objective function values of the solutions in R_ℓ and the reference point (cf. figure 6.3.8 and 6.3.7). To each of these cells we can attach a height (cf. figure 6.3.7). The height could be the minimal value of the third objective function among all solutions in R_ℓ that dominate the points of a cell in the first two objective function values. In this case, the described volume equals the hypervolume dominated by R_ℓ.

Algorithm 13 first partitions the search space into the aforementioned cells $B_{i,j} := [\binom{a_i}{b_j}, \binom{a_{i+1}}{b_{j+1}}[$, $i \in \{1, \ldots, k\}$, $j \in \{1, \ldots, k\}$, whereas the coordinates a_i and b_j correspond to the sorted function values of the first and second objective function, respectively.

In the second part of the algorithm, for each cell $B_{i,j}$ the lowest function value for f_3 is computed among all solutions that weakly dominate $B_{i,j}$ with regard to the first two objective function values. This value is stored in the variable $h_1(i,j)$. In addition, we compute $h_2(i,j)$, the value of which is the second lowest function value of f_3 for all solutions in R_ℓ that dominate the rectangle $B_{i,j}$ in the first two objective function values.

In the third part of algorithm 13 the values of h_1 and h_2 are used in order to compute

Algorithm 13 Hypervolume(R_ℓ) for $n_f = 3$

$k \leftarrow |R_\ell|$
{Part 1: Initialize cell coordinates (cf. figure 6.3.8)}
$(a_1, \ldots, a_k) \leftarrow$ sort $(R_\ell[1].f_1, \ldots, R_\ell[k].f_1)$ in ascending order
$(b_1, \ldots, b_k) \leftarrow$ sort $(R_\ell[1].f_2, \ldots, R_\ell[k].f_2)$ in ascending order
$b_{k+1} = y_1^{\max}, b_{k+1} = y_2^{\max}$
{Part 2: Compute $h_1(i,j)$ and $h_2(i,j)$, $i,j \in \{1,\ldots,k\}^2$}
for all $(i,j) \in \{1,\ldots,k\}$ **do**
  $h_1(i,j) \leftarrow y_3^{\max}$; $h_2(i,j) \leftarrow y_3^{\max}$
end for
for all $t \in \{1,\ldots,k\}$ **do**
  for all $(i,j) \in \{1,\ldots,k\}^2$ **do**
    if $R_\ell[t].f_1 \leq a_i \wedge R_\ell[t].f_2 \leq b_j$ **then**
      {Update lowest and second lowest height for the cell}
      if $R_\ell[t].f_3^{(t)} \leq h_1(i,j)$ **then**
        $h_2(i,j) \leftarrow h_1(i,j)$ {Second lowest height}
        $h_1(i,j) \leftarrow R_\ell[t].f_3$ {Lowest height}
      else
        if $R_\ell[t].f_3 < h_2(i,j)$ **then**
          $h_2(i,j) \leftarrow R_\ell[t].f_3$ {Lowest height}
        end if
      end if
    end if
  end for
end for
{Part 3: Sum up partial volume loss due to the removal of $R_\ell[t]$, $t = 1,\ldots,k$.}
for all $t \in \{1,\ldots,k\}$ **do**
  $\Delta\mathcal{S}_t \leftarrow 0$
  for all $(i,j) \in \{1,\ldots,k\}^2$ **do**
    if $R_\ell[t].f_1 \leq a_i \wedge R_\ell[t].f_2 \leq b_j \wedge R_\ell[t].f_3 = h_1(i,j)$ **then**
      {The height of the cell is determined by solution t}
      $\Delta\mathcal{S}_t \leftarrow \Delta\mathcal{S}_t + (a_{i+1} - a_i) \cdot (b_{i+1} - b_i) \cdot (h_2(i,j) - h_1(i,j))$
    end if
  end for
end for
{Return index of solution that least contributes to the dominated hypervolume}
return $\arg\min(\Delta\mathcal{S}_1, \ldots, \Delta\mathcal{S}_k)$

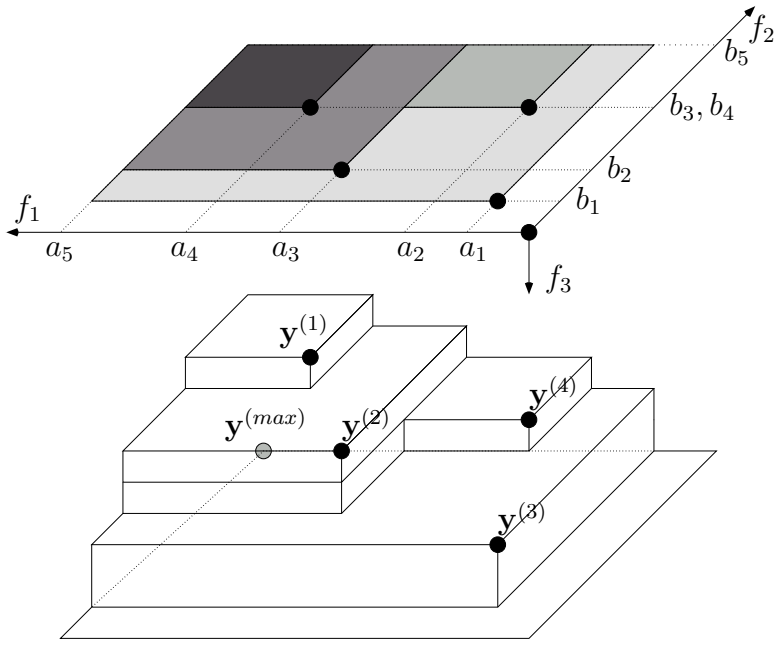


Figure 6.3.7: Visualization of the 3D-volume dominated by some points $\mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(4)}$. The values $a_1, \ldots, a_5, b_1, \ldots, b_5$ denote grid coordinates as used in the block partitioning of algorithm 13.

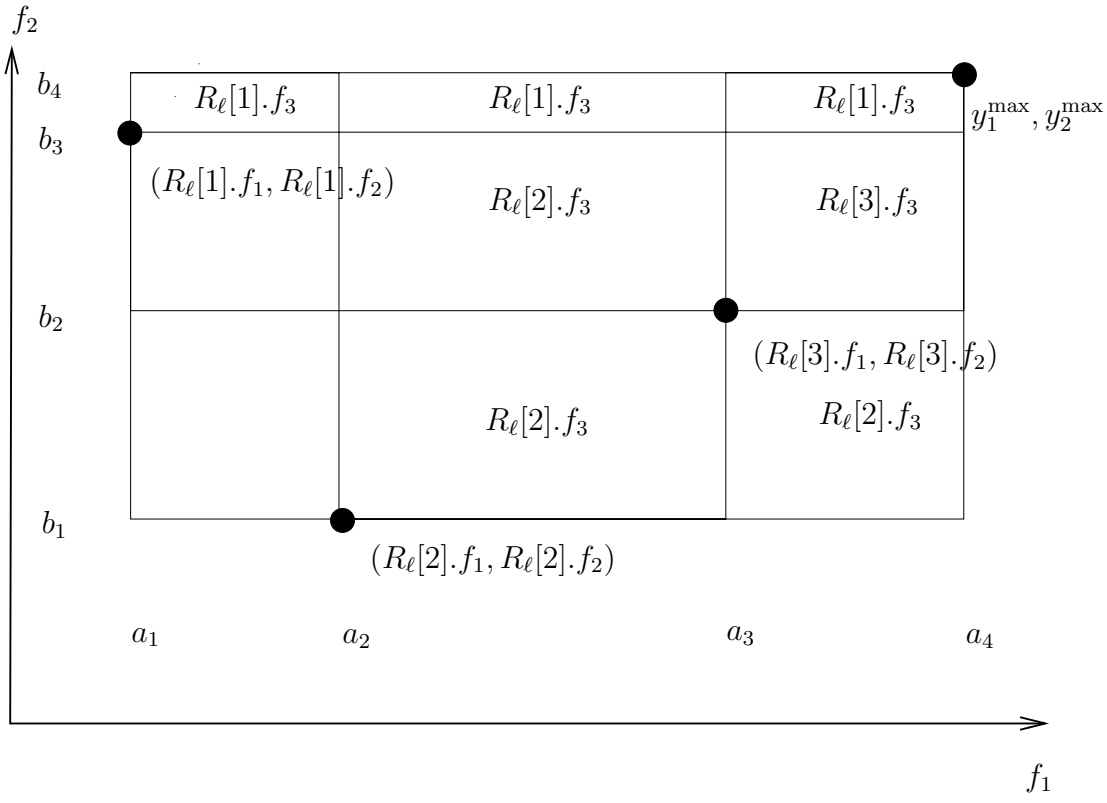


Figure 6.3.8: Partitioning of hypervolume in algorithm 13. The value of the height h_1 of each partially dominated cell after the execution of the algorithm is placed in its middle. For the given example, assume $R_\ell[1].f_3 < R_\ell[3].f_3 < R_\ell[2].f_3$.

the loss in hypervolume, whenever the t-th solution $\mathbf{y}^{(t)} := R_\ell[t]$ would be removed from R_ℓ, in other words the part of the dominated hypervolume that is exclusively dominated by the t-th solution. Note, that the volume of a box in our partitioning reduces only if the value of $h_1(i,j)$ equals the value of the third objective function of the solution that is removed. In that case the second best value has to be chosen for the height of the box in order to determine the box-shaped dominated volume with projection $B_{i,j}$ that remains, whenever only $\mathbf{y}^{(t)}$ is removed. Hence, the volume reduces by the volume of the intersection of that new box and the old box, the value of which can now be computed by $(a_{i+1} - a_i) \cdot (b_{j+1} - b_j) \cdot (h_2(i,j) - h_1(i,j))$.

The running time of the algorithm is $\mathcal{O}(k^3)$. This running time is governed by updating the values of h_1 and of h_2 for all points in the set. Since an update has to be made for each of the k points in R_ℓ, the procedure needs a total running time of $\mathcal{O}(k^3)$. Since we have stored the second best function values of f_3 for $B_{i,j}$, we can update h_1 within one scan over the $B_{i,j}$ boxes, taking a running time of $\mathcal{O}(k^2)$ for each point, for which the exclusively dominated hypervolume has to be determined. Hence, the total running time remains $\mathcal{O}(k^3)$. This is considerably lower than the running time $\mathcal{O}(k^4)$ for the repeated call of Fleischer's algorithm [Fle03] for subsets of R_ℓ.

More than three objectives

For more than three objectives, the algorithm proposed by Knowles, Corne and Fleischer [KCF03] should be used, that is a modified version of Fleischer's algorithm [Fle03] for the efficient calculation of the hypervolume. However, the running time of this algorithm grows exponentially with the number of objectives, why the choice of the hypervolume measure as selection criterion should be handled with care for more than three objectives.

6.3.6 Distribution of solutions

In order to get an impression on how the SMS-EMOA distributes solutions on Pareto fronts of different curvature, we conducted a study on simple but high dimensional test functions. The aim was to observe the algorithms behavior on convex, concave and linear Pareto fronts. For the study we propose the following family of simple generic functions:

$$f_1(\mathbf{x}) = (\sum_{i=1}^{n} |x_i|)^\gamma \cdot n^{-\gamma}, f_2(\mathbf{x}) := (\sum_{i=1}^{n} |x_i - 1|)^\gamma \cdot n^{-\gamma}, \mathbf{x} \in [0,1]^d \tag{6.3.10}$$

The extremal solutions of these two-dimensional functions (which we will abbreviate EBN) are given by $\mathbf{x}_1^* = (0,\ldots,0)^T$, $f_1(\mathbf{x}_1^*) = 1$, $f_2(\mathbf{x}_1^*) = 0$ and $\mathbf{x}_2^* = (1,\ldots,1)^T$, $f_1(\mathbf{x}_2^*) = 0$, $f_2(\mathbf{x}_2^*) = 1$. It is proven in appendix A.7 that the Pareto fronts can be described by the following function:

$$y_2(y_1, \gamma) = (1 - y_1^{1/\gamma})^\gamma, \gamma > 0, y_1 \in [0,1] \tag{6.3.11}$$

Thus, the choice of the parameter γ determines the characteristics of the curvature for the Pareto fronts of these functions.

Curvature of Pareto front	strongly convex	convex	linear	concave
EBN parameter γ	$\gamma = 4$	$\gamma = 2$	$\gamma = 1$	$\gamma = 0.5$

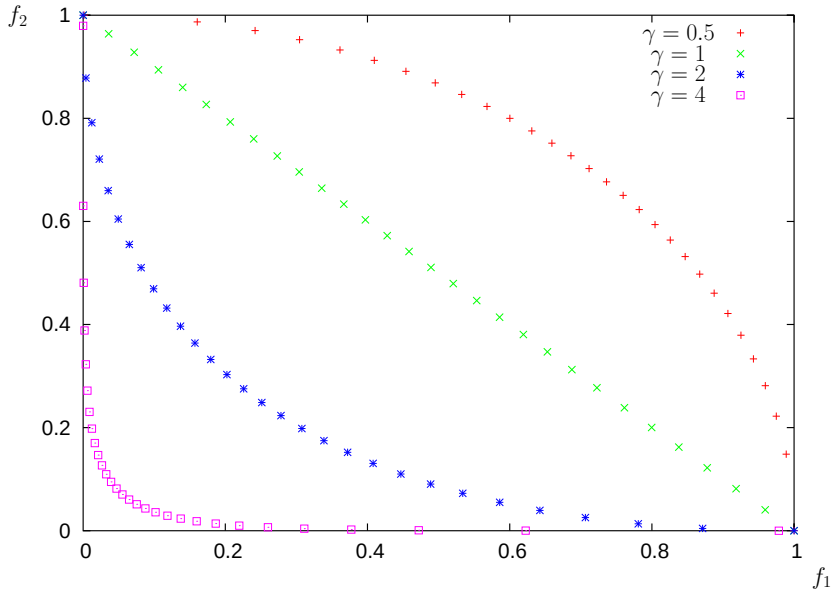


Figure 6.3.9: This figure visualizes the different solution sets for the SMS-EMOA with a population size of 30 on Pareto fronts of different curvature. In order to achieve different shapes of the Pareto front the 20-D EBN family of functions was employed with different values for γ (cf. expression 6.3.10). For each setting of γ, 20000 evaluations of the two objective functions were conducted, in order to obtain the displayed population.

An interesting feature of the EBN class of functions is, that the Pareto optimal set is given by the hypercube $\mathcal{C}_d = [0,1]^d$ (appendix A.7). Thus, if $\mathcal{C}_d$ is chosen as the search space, the only relevant selection criterion is the distribution of points on the Pareto front.

The results presented in figure 6.3.9 demonstrate that the solutions are not always distributed in a uniformly spaced way on the Pareto front. Rather, the SMS-EMOA concentrates solutions in regions where the Pareto front has knee-points. Furthermore, the results demonstrate that the SMS-EMOA produces a good approximation for concave Pareto fronts ($\gamma = 0.5$), where it also avoids a high sampling frequency for extremal solutions. Finally, it has been found that the SMS-EMOA distributes points almost uniformly on linear Pareto fronts ($\gamma = 1$).

6.3.7 Results on standard benchmarks

The SMS-EMOA has been tested on several test problems from literature, aiming at comparability to recent papers of Deb et al. presenting their ϵ-MOEA approach [DMM03b, DMM03a]. For example, exactly the same variation operators (polynomial mutation, simulated binary crossover, uniform initialization) and their parametrization have been used to test the approach. The test problems named ZDT1, ZDT2, ZDT3, ZDT4 and ZDT6 from [DMM03a, ZDT00] have been considered. The parameters and reference points were chosen according to the ones given in [DMM03b, DMM03a]. The population size was set to μ=100, and 20000 evaluations of the two objective functions were conducted for each run. For reasons of comparability we copied the results for the hyper-

volume measure and the convergence achieved in [DMM03a] to table 6.3.1. All algorithms listed in [DMM03a], namely NSGA-II, C-NSGA-II, SPEA, and ϵ-MOEA, have been compared to the SMS-EMOA.

The hypervolume or $\mathcal{S}$ metric of the set of non-dominated points is calculated as described above, using the same reference point as in [DMM03b, DMM03a]. As proposed by Deb et al. [DMM03b], the convergence measure is the average closest Euclidean distance to a point of the true pareto-optimal front. Note that the convergence measure is calculated concerning a set of 1000 equally distributed solution of the true Pareto front. Therefore, even pareto-optimal points do not have a convergence value of zero, except those being equal to one of the 1000 points in the sample.

The SMS-EMOA is ranked best concerning the $\mathcal{S}$ metric in all functions but ZDT6. Concerning the convergence measure, it has two first, two second and one third rank. According to the sum of ranks of the two measures on each function, one can state that the SMS-EMOA provides best results on all considered functions, except for ZDT6, where it is outperformed by SPEA2 (table 6.3.1). With regard to the sum of achieved ranks over all functions (table 6.3.2), the SMS-EMOA obtains best results concerning both the convergence measure (with 9) and the $\mathcal{S}$ metric (with 6). Summing up, concerning this benchmark and performance measures, the SMS-EMOA can be regarded as the best algorithm.

Let us now go into details about the different benchmark functions observed: ZDT1 has a smooth convex Pareto front (cf. appendix A.1). On this problem all algorithms achieve nearly optimal metric values. ZDT4 (cf. appendix A.4)is a multimodal function with multiple parallel Pareto fronts, whereas the best front is equivalent to that of ZDT1. On the basis of the given values from [DMM03b, DMM03a], it can be assumed that all algorithms, including the SMS-EMOA, achieved to pass the second front with most solutions and aimed at the first front. ZDT2 has a smooth concave front and the SMS-EMOA has nearly optimal hypervolume (cf. appendix A.2). This resolves doubts that the SMS-EMOA succeeds in concave regions, that might be casted, because it is well known that the $\mathcal{S}$ metric favors convex regions. ZDT3 has a discontinuous Pareto front (cf. appendix A.3)that consists of five slightly convex parts. Here, the SMS-EMOA is little better concerning the $\mathcal{S}$ metric than the ϵ-MOEA and significantly better concerning the convergence. ZDT6 has a concave Pareto front that is equivalent to that of ZDT2, except the differences that the front is truncated to a smaller range and that points are non-uniformly spaced. The SMS-EMOA is ranked second on both measures, only outperformed by SPEA2, which obtained worse results on the other easier functions.

The performance concerning the $\mathcal{S}$ metric is a very encouraging result, even though good results have been expected, because the $\mathcal{S}$ metric itself served as selection criterion. However, one should not forget, that the new approach is a rather simple one with only one population and it is steady-state, resulting in a low selection pressure. Neither there are any special variation operators fitted to the selection strategy, nor it is tuned for performance in any way. All these facts would normally imply not that good results. The good results in the convergence measure are maybe more surprising. It is worth to know, that for all but ZDT4, where the optimum is located exactly in the middle of the search space, optimal values lie at the boundaries of the search space, which simplifies the exact optimization. The SMS-EMOA, like many other algorithms, sets values that exceed the bounds of the search space exactly on these bounds. This allows on the

ZDT functions for good convergence without adaptation of the mutation jump length. However, since we used the same variation operators as in ϵ-MOEA, the good results for the SMS-EMOA have been achieved in a fair scenario.

As it has been said before, the SMS-EMOA is well-suited for the integration of metamodel. The reason for this is mainly, that integral criteria like the ExI and the PoI can be easily formulated on the basis of $\Delta_{\mathcal{S}}$. Accordingly, the SMS-EMOA has been considered as a basic algorithm for the metamodel-assisted design optimization with multiple objectives.

6.4 Conclusions

A brief introduction into Pareto optimization was given. Besides an introduction to the NSGA-II algorithm, the SMS-EMOA has been proposed as a new algorithm. This algorithm uses the hypervolume measure in its selection criterion. The motivation for introducing a new algorithm was to make a generalization of filters based on criteria of improvement possible.

The SMS-EMOA has been tested on the standard ZDT benchmark with two objective functions. It turned out to be superior to results published earlier on this benchmark with well established EMOA variants (e. g. SPEA, ϵ-MOEA, and NSGA-II). Only for the ZDT6 function SPEA performed slightly better than the SMS-EMOA with regard to the convergence metric.

Efficient computational procedures for the SMS-EMOA have been proposed for problems with two and three objective functions. In particular we proposed methods that compute differences in the dominated hypervolume that are asymptotically faster than the methods published so far. Furthermore the distribution of solutions for this algorithm on Pareto fronts with different curvature (convex, concave, linear) has been measured. It turned out that knee points and extremal solutions are well covered.

It shall also be remarked, that for the latter analysis a new family of multi-objective test problems has been introduced, namely the EBN family of functions. Also we conducted a rigorous analysis of this problem family that allows to gradually adjust the curvature of the Pareto front by means of a single parameter.

Very recently, the SMS-EMOA was studied on problems with three objectives. For these new results we refer to Naujoks et al. ([NBE]). The results prove that the approach outperforms established algorithms like NSGA-II and SPEA-II on standard benchmarks, using the convergence as well as the hypervolume metric as a performance measure.

Test-function	Algorithm	Convergence measure			$\mathcal{S}$ measure		
		Average	Std. dev.	Rank	Average	Std. dev.	Rank
ZDT1	NSGA-II	0.00054898	6.62e-05	3	0.8701	3.85e-04	5
	C-NSGA-II	0.00061173	7.86e-05	4	0.8713	2.25e-04	2
	SPEA2	0.00100589	12.06e-05	5	0.8708	1.86e-04	3
	ϵ-MOEA	**0.00039545**	1.22e-05	1	0.8702	8.25e-05	4
	SMS-EMOA	0.00044394	2.88e-05	2	**0.8721**	2.26e-05	1
	true Pareto front	0	0	0	0.8761	-	0
ZDT2	NSGA-II	**0.00037851**	1.88e-05	1	0.5372	3.01e-04	5
	C-NSGA-II	0.00040011	1.91e-05	2	0.5374	4.42e-04	3
	SPEA2	0.00082852	11.38e-05	5	0.5374	2.61e-04	3
	ϵ-MOEA	0.00046448	2.47e-05	4	0.5383	6.39e-05	2
	SMS-EMOA	0.00041004	2.34e-05	3	**0.5388**	3.60e-05	1
	true Pareto front	0	0	0	0.5427	-	0
ZDT3	NSGA-II	0.00232321	13.95e-05	3	1.3285	1.72e-04	3
	C-NSGA-II	0.00239445	12.30e-05	4	1.3277	9.82e-04	5
	SPEA2	0.00260542	15.46e-05	5	1.3276	2.54e-04	4
	ϵ-MOEA	0.00175135	7.45e-05	2	1.3287	1.31e-04	2
	SMS-EMOA	**0.00057233**	5.81e-05	1	**1.3295**	2.11e-05	1
	true Pareto front	0	0	0	1.3315	-	0
ZDT4	NSGA-II	0.00639002	0.0043	4	0.8613	0.00640	2
	C-NSGA-II	0.00618386	0.0744	3	0.8558	0.00301	4
	SPEA2	0.00769278	0.0043	5	0.8609	0.00536	3
	ϵ-MOEA	0.00259063	0.0006	2	0.8509	0.01537	5
	SMS-EMOA	**0.00251878**	0.0014	1	**0.8677**	0.00258	1
	true Pareto front	0	0	0	0.8761	-	0
ZDT6	NSGA-II	0.07896111	0.0067	4	0.3959	0.00894	5
	C-NSGA-II	0.07940667	0.0110	5	0.3990	0.01154	4
	SPEA2	**0.00573584**	0.0009	1	**0.4968**	0.00117	1
	ϵ-MOEA	0.06792800	0.0118	3	0.4112	0.01573	3
	SMS-EMOA	0.05043192	0.0217	2	0.4354	0.02957	2
	true Pareto front	0	0	0	0.5427	-	0

Table 6.3.1: Results of SMS-EMOAon ZDT Test-suite.

Algorithm	Convergence measure Ranks					$\sum$ of ranks	$\mathcal{S}$ measure Ranks					$\sum$ of ranks
NSGA-II	3	1	3	3	4	14	5	5	3	1	5	19
C-NSGA-II	4	2	4	2	5	17	2	3	5	4	4	18
SPEA2	5	5	5	4	1	20	3	3	4	2	1	13
ϵ-MOEA	1	4	2	1	3	11	4	2	2	5	3	16
SMS-EMOA	2	3	1	1	2	**9**	1	1	1	1	2	**6**

Table 6.3.2: Ranks and sum of ranks from table 6.3.1.

7 Metamodel-assisted multi-objective optimization

Every generalization is dangerous, especially this one.

S. L. Clemens

In this chapter it is discussed how metamodeling techniques can be integrated into the NSGA-II and SMS-EMOA. In section 7.1, we discuss general aspects of the integration of metamodels into these EMOA. Then, in section 7.2, we focus on the generalization of filters used for pre-screening solutions. Finally, the metamodel-assisted EMOA are evaluated on test problems (section 7.3).

7.1 Introduction

The integration of filters based on metamodels into the NSGA-II can be done in a similar manner than for the MAES. From the offspring generated by the variation operators, only a subset is precisely evaluated and considered in the replacement. All other individuals are rejected by the filter. If filters are employed that let pass always a constant number of ν offspring individuals, the modified NSGA-II will be termed a $(\mu, \kappa, \nu < \lambda)$-NSGA-II. A simple design for a filter would be to compute the predictions and then select the ν best solutions due to non-dominated sorting. However, more sophisticated procedures can be sought, that also make use of confidence information. Different possibilities will be discussed in the subsequent sections.

For the SMS-EMOA, not only the evaluation procedure, but also the generation procedure needs to be adapted. This is due to the fact that the SMS-EMOA is a steady state EMOA, and hence only one individual is generated and evaluated in each iteration. It proves to be a good strategy (cf. [EBN05]) to produce a surplus of λ individuals and then extract the most promising solution by means of a filter based on metamodels. Only this single solution is considered in the replacement. The resulting strategy will be termed a $(\mu + 1 < \lambda)$-SMS-EMOA.

The basic procedure of the $(\mu + \lambda)$-MA-SMS-EMOA as proposed by Emmerich, Beume, and Naujoks [EBN05] is outlined in algorithm 14. After initialization of a database of results, from which a start population is randomly extracted, the generational loop starts. In each iteration t, an individual is chosen from the population P_t. Then, λ offspring individuals are generated by application of the mutation operator to this individual. After this, the most promising solution among all candidate solutions is chosen by means

of approximate objective function evaluations. This solution is evaluated by the time consuming simulator and considered in the replacement.

Algorithm 14 Metamodel-assisted SMS-EMOA.

1: $P_0 \leftarrow \text{init}()$ {Initialize and evaluate start population of μ individuals}
2: $D \leftarrow P_0$ {Initialize database}
3: $t \leftarrow 0$
4: **repeat**
5: Draw $\mathbf{x}_t$ randomly out of P_t
6: $\mathbf{x}_{t,i} = \text{mutate}(\mathbf{x}_t), i = 1, \ldots, \lambda$ {Generate λ solutions via mutation}
7: approximate$(D, \mathbf{x}_{t,1}, \ldots, \mathbf{x}_{t,\lambda})$ {Approximate results with local metamodels}
8: $\mathbf{x}_{t+1} \leftarrow$ filter$(\mathbf{x}_{t,1}, \ldots, \mathbf{x}_{t,\lambda})$ {Detect 'best' approximate solution}
9: evaluate $\mathbf{x}_{t+1}$
10: $D \leftarrow D \cup \{\mathbf{x}_{t+1}\}$
11: $P_{t+1} \leftarrow$ **reduce**$(P_t \cup \{\mathbf{x}_{t+1}\})$ {Select new population}
12: $t \leftarrow t + 1$
13: **until** stop criterion reached

Having explained the basic generational loop of the metamodel-assisted EMOA, the question remains open of how to select a subset of promising solutions from a set of individuals. This topic is addressed in the next section.

7.2 Generalization of IPE-filters

The generalization of the IPE-filters, that were introduced in section 4.2, to multi-objective optimization is addressed in this section. Promising solutions according to the Pareto dominance relation (cf. expression 6.1.1) are sought. Moreover, an increase of the diversity of the non-dominated set is envisaged. Similarly to the constrained case, a multivariate (n_f-dimensional) distribution for each point $\mathbf{x} \in \mathbb{S}$ is provided by the gaussian random field model. By default we assume independency of the predictive multivariate distributions, but, whenever it seems suitable, we shall also examine the case of correlated multivariate distributions.

Figure 7.2.1 visualizes the predictive probability densities for three approximated solutions in a two-dimensional solution space. Whenever we make the assumption of independent output variables (cf. section 5.4), the vector $\hat{\mathbf{y}}(\mathbf{x})$ shall denote the mean value of the n_f-dimensional gaussian distribution and $\hat{\mathbf{s}}(\mathbf{x})$ the vector of standard deviations attributed to the predictions.

The main difficulty in the generalization of the pre-screening procedures to the multi-objective case is that the notions of best found solution and of improvement, if attributed to a single solution, stop making sense in the context of Pareto optimization with conflicting objectives, because we have to deal with incomparable solutions. In order to resolve this problem, in the presence of incomparable solutions the contribution of the solutions to the diversity or to the increase in the hypervolume measure ($\mathcal{S}$ metric) shall govern the choice of promising candidate solutions.

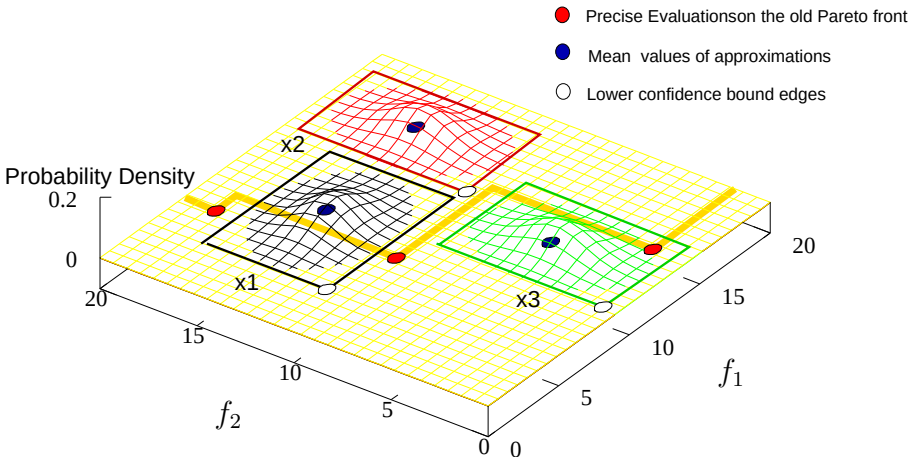


Figure 7.2.1: Example for the prediction of solutions in a solution space with two objectives: The picture visualizes the probability density functions of the predictive distributions for three search points $\mathbf{x}_1$, $\mathbf{x}_2$ and $\mathbf{x}_3$ for a two-objective problem. The black points mark the mean values $\hat{\mathbf{y}}$ of the probability density functions. The white points mark the lower confidence bounds and the rectangles depict confidence interval boxes, symmetrically surrounding the mean value approximations. By comparison of the predictive distribution with the precise solution vectors of the current population (here given by the red points), promising solutions can be detected (cf. section 7.2).

7.2.1 Mean value and lower confidence bound filters

First, let us discuss the generalization of the mean value and lb_ω filter for the NSGA-II algorithm. The pre-selection procedure that was first mentioned and evaluated by Emmerich and Naujoks [EN04a, EN04b] is both simple and effective. In the metamodel-assisted NSGA-II for all λ offspring individuals $\mathbf{x}_1, \ldots, \mathbf{x}_\lambda$ the mean value $\hat{\mathbf{y}}(\mathbf{x})$ is determined. Then the subset of the most promising ν solutions is detected by non-dominated sorting on the predicted values. Instead of using $\hat{\mathbf{y}}$ as pre-screening function, also

$$\mathbf{lb}_\omega(\mathbf{x}) = \mathbf{y}(\mathbf{x}) - \omega \cdot \hat{\mathbf{s}}(\mathbf{x}) \tag{7.2.1}$$

can be employed as a pre-screening function. In correspondence with the single objective case, this procedure will be termed lower confidence bound (lb_ω) filter. Both, the predicted solution vectors for the mean value filter and the lower confidence bound filter are visualized for the example in figure 7.2.1. Again, the idea behind this choice is, to reward solutions that are placed in unexplored regions of the search space, in order to prevent premature stagnation of the search. The parameter ω scales the quantity of this reward. The results of Emmerich and Naujoks [EN04a] indicated that in the presence of multiple objectives this strategy leads not only to a higher robustness but also to a better coverage of the Pareto front.

The values of $\mathbf{lb}_\omega$ can be interpreted as lower confidence confidence bounds for the true solution vector. Similar to the constrained case (expression 5.4.16), we can adjust the confidence level of this lower confidence bound by means of

$$p_\alpha = \Phi(-\omega)^{n_f}, \Phi(y) = \frac{1}{2}(1 + \text{erf}(\frac{y}{\sqrt{2}})). \tag{7.2.2}$$

Here, p_α measures the probability that all objective function values are higher than the corresponding values of the lower confidence bound, assuming, of course, that the model assumptions are valid.

The $\hat{y}$ and lb_ω filters can also be generalized for the SMS-EMOA. It is not sufficient to employ the increase in hypervolume $\Delta_\mathcal{S}$ as a criterion, since this would not provide a criterion for selecting a new solution, if the approximations of all candidate solutions are dominated and thus non of them increases the hypervolume. Instead, we need a criterion that provides also a rank for dominated candidate solutions.

A possible approach to solve this problem is to modifying the non-dominated sorting procedure again: Let G_t denote the set of generated solutions from which we want to obtain the most promising candidate solution (cf. algorithm 14). For each $\mathbf{x} \in G_t$, we determine the partitions of equal dominance rank $R_1(\mathbf{x}), \ldots, R_\ell(\mathbf{x})$ of the set $P_t \cup \{\mathbf{x}\}$. Next, each partition is sorted by the $\Delta_\mathcal{S}$ criterion. After these two steps, a non-dominance rank $r(\mathbf{x})$ and the contribution in hypervolume $\Delta_\mathcal{S}(\mathbf{x}, R_{(r(\mathbf{x}))})$ for $\mathbf{x}$ in the partition of rank $r(\mathbf{x})$, is detected. After having computed the two values r and $\Delta_\mathcal{S}$ for each individual $\mathbf{x} \in G_t$, the pairs $(r(\mathbf{x}_i), -\Delta_\mathcal{S}(\mathbf{x}, R_{(r(\mathbf{x}_i))}))$, $i = 1, \ldots, |G_t|$ are sorted lexicographically. Then, the best ranked solution is selected by the IPE-filter.

In particular, among the non-dominated offspring solutions this procedure selects the solution that maximizes $\mathcal{S}(P \cup \{\mathbf{x}\}) - \mathcal{S}(P)$.

Now, by replacing the predicted vector $\hat{\mathbf{y}}(\mathbf{x})$ by the lower confidence bound vector $\mathbf{lb}_\omega(\mathbf{x})$ (cf. figure 7.2.1), we can generalize also the lb_ω filters. Here, we are more optimistic about the outcome of the experiment, meaning that, in general, we assume a larger increase in hypervolume measure of the population, whenever a particular new solution is selected. The difference in the estimated increase, compared to the mean value, depends again on the degree of uncertainty and on the choice of ω.

7.2.2 Filters based on measures of improvement

Next, we consider the generalization of filters that work with criteria based on a measure of improvement. The definition of an improvement will be based on the hypervolume measure. Thus, the resulting filters are predisposed for the application within the SMS-EMOA. However, as it will be shown later, they may also be used in the pre-screening algorithm of the metamodel-assisted NGSA-II, though, this might imply the introduction of a reference point for the latter algorithm.

We define that any new point that is non-dominated by all points in a set of solutions $P = \{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$ is an *improvement* with regard to this set. Furthermore, the impact of an improvement is measured by the gain in dominated hypervolume, if the solution would be added to P. Let A denote the set of output vectors for search points in P, i. e. $A := \{\mathbf{y}(\mathbf{x}) | \mathbf{x} \in P\}$. Then the improvement measure reads:

$$\mathrm{I}(\mathbf{y}) := \mathcal{S}(\{\mathbf{y}\} \cup A) - \mathcal{S}(A). \tag{7.2.3}$$

Since the integration of non-dominated points always increases the hypervolume of a set, $\mathrm{I}(\mathbf{y})$ takes a positive value, iff $\mathbf{y}$ is non-dominated by solutions in P. Otherwise, the value

of $\mathrm{I}(\mathbf{y})$ is zero. Based on this definition, the criteria for improvement can be generalized as discussed in the following subsections.

Probability of improvement

The probability of improvement (PoI) criterion has been proposed by Ulmer et al. [USZ03] for single-objective optimization. Now, it shall be generalized for multi-objective optimization.

Let $\overline{\mathcal{H}}(A)$ denote the non-dominated subspace of $\mathbb{R}^{n_f}$

$$\overline{\mathcal{H}}_f(A) := \{\mathbf{y} \in \mathbb{R}^{n_f+n_g} | \mathbf{y} \preceq A\} \tag{7.2.4}$$

and $\mathrm{PDF}_{\mathbf{x}}$ denote the probability density function of the predictive gaussian distribution for a new point $\mathbf{x}$. The probability that the new point is an improvement, i. e. it is non-dominated by A, is then given by the integral of $\mathrm{PDF}_{\mathbf{x}}$ over the non-dominated region

$$\mathrm{PoI}(\mathbf{x}) = \int_{\mathbf{y} \in \overline{\mathcal{H}}_f(A)} \mathrm{PDF}_{\mathbf{x}}(\mathbf{y}) d\mathbf{y}. \tag{7.2.5}$$

When working with independent predictive distributions (cf. section 5.4), we can compute this value from the mean values $\hat{\mathbf{y}}$ and standard deviations $\hat{\mathbf{s}}$ of the predictive distribution. For the unconstrained case $n_g = 0$ we get:

$$\mathrm{PoI}(\mathbf{x}) := \int_{\mathbf{y} \in \overline{\mathcal{H}}_f(A)} \prod_{i=1}^{n_f} \varphi(\frac{y_i - \hat{y}_i}{\hat{s}_i}) d\mathbf{y}. \tag{7.2.6}$$

For the constrained multi-objective case we can extend this expression to

$$\mathrm{PoI}(\mathbf{x}) := \int_{\mathbf{y} \in \overline{\mathcal{H}}_f(A)} \prod_{i=1}^{n_f} \varphi(\frac{y_i - \hat{y}_i}{\hat{s}_i}) \cdot \prod_{i=n_f}^{n_f+n_g} \varphi(\frac{-\hat{y}_i}{\hat{s}_i}) d\mathbf{y}. \tag{7.2.7}$$

This expression combines the probability that a solution is non dominated with regard to its objective function values (first factor in the integral) with the probability that a solution is feasible (second factor in the integral). For a motivation of the second part of the integrand we refer to section 5.4.2, where the treatment of constraints in improvement-based filters gets fully explicated.

The computation of the PoI can be done by means of a modified version of Fleischer's algorithm [Fle03] for computing the hypervolume measure. One by one, this algorithm lops off n_f-dimensional hyper-boxes from the dominated subspace until the space has been reduced to the empty set. Given an infinite reference point $\mathbf{y}^{\max} = (\infty, \ldots, \infty)^T$, by means of Fleischer´s algorithm the dominated hypervolume gets partitioned into a set of closed and half-open interval hyperboxes[1]. Instead of summing up the Lebesgue measures of the hyper-boxes the finite probabilities that a solution is placed inside one of the hyper-boxes is summed up. Now, the probability of improvement can be obtained as $\mathrm{PoI}(\mathbf{x}) = 1 - p_d$.

[1] Some of the coordinates can take the value ∞.

To put things into more concrete terms, let us denote the sequence of interval hyper-boxes lopped off by Fleischer's algorithm as $B_1, \ldots, B_m$ and denote the lower bound edges of these boxes with

$$(b_{i,1}^{min}, \ldots, b_{i,n_f}^{min})^T, i = 1, \ldots, m \tag{7.2.8}$$

and their upper bound edges with

$$(b_{i,1}^{max}, \ldots, b_{i,n_f}^{max})^T, i = 1, \ldots, m. \tag{7.2.9}$$

Moreover, let us assume that the infinite reference point $\mathbf{y}^\infty$ is chosen. Then the integral for the PoI reads:

$$\text{PoI}(\mathbf{x}) := 1 - \sum_{i=1}^{m} \int_{\mathbf{y} \in B_i} \text{PDF}_{\mathbf{x}}(\mathbf{y}) d\mathbf{y}. \tag{7.2.10}$$

Working with mutually independent distributions (cf. section 5.4), we get

$$\begin{aligned} \text{PoI}(\mathbf{x}) &:= 1 - \sum_{i=1}^{m} \int_{\mathbf{y} \in B_i} \prod_{j=1}^{n_f} \varphi(\frac{y_j - \hat{y}_j}{\hat{s}_j}) d\mathbf{y} & (7.2.11) \\ &= 1 - \sum_{i=1}^{m} \prod_{j=1}^{n_f} \Phi(\frac{b_{i,j}^{max} - \hat{y}_j}{\hat{s}_j}) - \Phi(\frac{b_{i,j}^{min} - \hat{y}_j}{\hat{s}_j}). & (7.2.12) \end{aligned}$$

Now, with expression 7.2.12, we have a formula that can be directly computed. A welcome characteristic of the discovered expression is that it does not demand for a finite reference point, the choice of which would be up to the user.

Given independent outputs (cf. section 5.4), expression 7.2.12 can be easily generalized to the constrained case by multiplying it with the probability that a solution is feasible:

$$\text{PoI}(\mathbf{x}) := (1 - \sum_{i=1}^{m} \prod_{j=1}^{n_f} (\Phi(\frac{b_{i,j}^{max} - \hat{y}_j}{\hat{s}_j}) - \Phi(\frac{b_{i,j}^{min} - \hat{y}_j}{\hat{s}_j}))) \prod_{i=n_f+1}^{n_f+n_g} \Phi(\frac{-\hat{y}_i}{\hat{s}_i}). \tag{7.2.13}$$

Still, this term can be directly computed. Summing up, it has been found that both in the unconstrained case and in the constrained case, the value of the PoI can be computed by means of a direct formula.

The running time complexity for computing expression 7.2.13 is governed by the running time complexity of Fleischer's algorithm. Accordingly, we obtain the running time complexity $\mathcal{O}(k^3 \cdot n_f^2 + n_g)$ with k being the number of non-dominated solutions in P. The first addend stems from Fleischer's algorithms, while the second addend describes the computation time for the probability that the new solution is feasible.

Provided $\mathbf{s}(\mathbf{x}) > 0$, the probability of improvement takes always positive values greater than zero. Thus, unlike for the mean value and lower confidence bound criterion, dominated solutions can easily be compared to non-dominated solutions. This makes the PoI criterion applicable in a straightforward manner as a ranking criterion for a filter with fixed output-size.

Like in the single-objective case, it might also be a problem with the PoI filter in the multi-objective case, that the probability of improvement criterion has the tendency to favor very small improvements to larger improvements. For example, if the PoI indicates

that a solution has a probability of improvement of 0.5, it does not matter, whether this improvement is large or very small. Generally, one would consider the second option as favorable.

Thus, as an alternative measure, the expected improvement might be considered, that takes into account not only the probability but also the quantity of possible improvements.

Expected improvement

Based on the improvement measure (expression 7.2.3), the expected improvement is described as:

$$\mathrm{ExI}(\mathbf{x}) = \int_{\mathbf{y}\in]-\mathbf{y}^{\infty},\mathbf{y}^{\max}]} \mathrm{PDF}_{\mathbf{x}}(\mathbf{y}) \cdot \mathrm{I}(\mathbf{y}) d\mathbf{y}, \tag{7.2.14}$$

where $\mathbf{y}^{\infty} \in \mathbb{R}^{n_f}$ denotes the vector $(\infty, \ldots, \infty)^T$. If we would choose an infinite reference point, this expression would also be infinite, provided $\hat{\mathbf{s}}$ is greater than zero. The only choice we have is to restrict the solution space by choosing a finite reference point $\mathbf{y}^{\max}$ for the computation of the hypervolume (cf. expression 6.3.1). For most practical problems in multi-objective optimization it is very easy to find a rough restriction for the solution space. However, it is often difficult to normalize a restricted solution space a-priori. Thus, the improvement measure might easily be biased towards measuring improvements to one of the objectives. Accordingly, a quantitative interpretation of the ExI criterion should be handled with care.

In contrast to the PoI (expression 7.2.4), the integrand in expression 7.2.14 is not solely the gaussian probability density function, but it includes further factors that stem from the improvement measure $\mathrm{I}(\mathbf{y})$. This is the reason, why even in the case of integrations over a rectangular region, the integral cannot be factorized into one-dimensional integrals. As a consequence, no direct formula can be given for the expected improvement.

Instead of providing a closed expression , numerical integration techniques shall be employed, in order to compute the ExI. An algorithm to compute the ExI, that is comparably easy to implement, is monte carlo integration. This integration method can be implemented by means of the following procedure: A large sample $\mathbf{y}_1, \ldots, \mathbf{y}_m$ is drawn from the gaussian distribution with mean $\hat{\mathbf{y}}$ and $\hat{\mathbf{s}}$.

Based on this sample, we compute the expected value by

$$\mathrm{ExI}(\mathbf{x}) \approx \frac{1}{m} \sum_{i=1}^{m} \mathrm{I}(\mathbf{y}_i). \tag{7.2.15}$$

Note that the reference point needs to be chosen sufficiently large, to avoid samples outside the box $[-\mathbf{y}^{\infty}, \mathbf{y}^{\max}]$. The generation of gaussian distributed samples for an independent gaussian distribution can be simply done by adding a standard normal distributed pseudo-random number scaled by the standard deviation $\hat{s}_i(\mathbf{x})$ to each vector position of $\hat{\mathbf{y}}(\mathbf{x})$. If $\mathrm{PDF}_{\mathbf{x}}$ is a multivariate distribution that does not factorize, the Metropolis algorithm can be used instead to generate samples ([Wei00], pp. 27).

An error estimate for the monte carlo estimate is given by:

$$(\frac{1}{m}\sum_{i=1}^{m}\mathrm{I}(\mathbf{y}_i)) - \mathrm{ExI}(\mathbf{x}) \approx S^2/m. \tag{7.2.16}$$

Here the variance S^2 is defined as

$$S^2 = \frac{1}{m-1}\sum_{i=1}^{m}(\mathrm{I}(\mathbf{y}_i) - \frac{1}{m}\sum_{i=1}^{m}\mathrm{I}(\mathbf{y}_i))^2. \tag{7.2.17}$$

This result can directly be obtained from the theory of monte carlo integration ([Wei00], pp. 11). Regardless of the dimension of the search space, the error scales like $1/\sqrt{m}$. Better error bounds can be obtained by variance reducing techniques described in ([Wei00], pp. 13).

We shall not stick to the details of numerical integration here. Rather, we remark that a precise estimation of the expected improvement measure might be very time consuming. However, it is still possible to use rough estimates of the ExI in order to pre-screen solutions, and in practice this might also yield good results. Moreover, in low dimensional spaces and/or for small population sizes, the computation of the improvement is fast and thus a large number of samples can be evaluated, yielding in small approximation errors.

Provided we model constraint functions independently from objective functions, we can easily generalize to the constrained multi-objective problem. This can be done in the same way as it has been done for the PoI, i. e. by multiplying the ExI expression for the unconstrained case with the probability that a solution is feasible.

Again, there are several possibilities of how to employ the ExI as a filter criterion in an EMOA. A simple method would be to rank solutions by means of the ExI and select a user-defined number of ν best solutions. For the SMS-EMOA, that always evaluates a single solution per generation, choosing $\nu = 1$ seems to be the adequate way of how to design such a filter.

7.2.3 Filters with adaptive output-size

Simple constructions for filters with adaptive output-size are the generalized MLI- and LBI_ω-filter. The *most likely improvement* measures the gain in hypervolume that is realized with the highest probability, i. e.

$$\mathrm{MLI}(\mathbf{x}) = \mathcal{S}(\{\hat{\mathbf{y}}(\mathbf{x})\} \cup A) - \mathcal{S}(A) \tag{7.2.18}$$

In contrast to the MLI, the *potential improvement* LBI_ω takes into account the standard deviation $\hat{\mathbf{s}}$ of the approximation:

$$\mathrm{LBI}_\omega(\mathbf{x}) = \mathcal{S}(\{\hat{\mathbf{y}}(\mathbf{x}) - \omega \cdot \hat{\mathbf{s}}(\mathbf{x})\} \cup A) - \mathcal{S}(A). \tag{7.2.19}$$

Both criteria can be applied for filtering solutions, accepting only solutions with values greater than zero. In case of the LBI_ω criterion, this means that we would dismiss

non-dominated solutions with probability of $1 - p_\alpha(\omega)$. Recall, that $p_\alpha(\omega)$ denotes the confidence level, the value of which can be adjusted by means of expression 7.2.2.

Moreover, versions of the ExI and PoI filter with adaptive output size can be sought. In order to allow for an adaptive output size, a threshold value τ can be provided that has to be surpassed by the solutions in order to get accepted. As in the single-objective case, such parameterized filters will be termed PoI_τ- and ExI_τ-filter.

A conceptual drawback of the ExI_τ-filter is, that a constant value of τ for the ExI_τ might not be a guarantee for a good performance in all stages of the search. Typically, in the starting phase of the Pareto optimization large values of the threshold seem to be appropriate, in order to move the population quickly towards the Pareto front. Later, small values for the expected improvement should also be accepted, since this allows for a refinement.

Adaptation problems like these are not to be faced when working with the PoI_τ-filter, because the probability of improvement is invariant to the absolute value of the hypervolume increment. However, the PoI_τ-filter might accept too many small improvements, since it is often easier to obtain a success when using only a small step size. This might slow down the convergence in the beginning of the search and even lead to pre-mature convergence.

Unlike in the single-objective case, in Pareto optimization the LBI_ω-filter is not equivalent to PoI_τ-filters with a threshold probability of $p = \Phi(-\omega)$. Neither is the MLI-filter equivalent to the PoI_τ filter with a threshold value of $\tau = 0.5$. Situations where the $\text{PoI}_{\Phi(-\omega)}$ filter accepts solutions that are rejected by the LBI_ω-filter can be considered, as well as situations where the $\text{PoI}_{\Phi(-\omega)}$ filter rejects solutions that are accepted by the LBI_ω-filter. An example for both situations an example is given in figure 7.2.2.

7.2.4 Interval filters

Interval based filters for multi-objective optimization have been suggested by Emmerich et. al. [EN04a] where they have been applied for the optimization of airfoil shapes. We will not go into details here. Rather, we note that the design of these filters is straightforward, if we consider confidence interval boxes rather than confidence intervals and compare upper and lower confidence bound vectors as we did for the constrained case 5.4.4.

Note that theorem 2 and 3 can be generalized to the multi-objective case (replace scalar upper and lower bounds by vector valued upper and lower bound for interval boxes), while theorem 4 does no longer hold, since it is based on the linear ordering of solutions. Accordingly, we have to use the modified version of algorithm 6 and 7 in order to determine output sets of the filters.

As an example let us consider the set of three solutions given in figure 7.2.1. Set $\mu = 1$ and let us also consider the solutions on the 'old' pareto front. In that case solution $\mathbf{x}_2$ will be rejected by the R_ω-filter , since its lower bound edge is dominated by one of the old solutions. None of the solutions would be accepted by the P_ω-filter , since all of the solutions are potentially dominated by one of the precisely evaluated 'old' solutions,

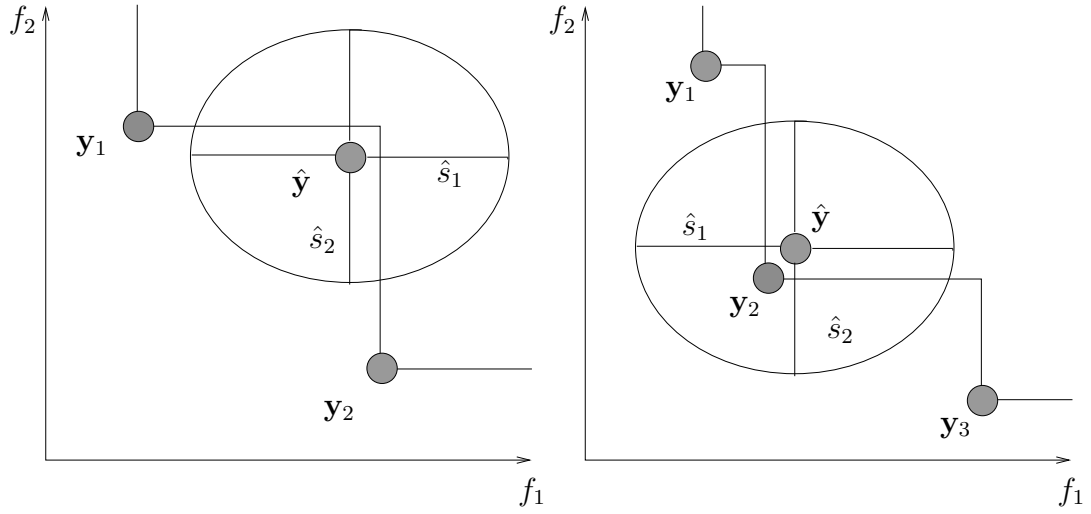


Figure 7.2.2: Comparison of the MLI and the $\mathrm{PoI}_{0.5}$ criterion in the multi-objective case. $\mathbf{y}_1, \mathbf{y}_2$ and $\mathbf{y}_3$ denote non-dominated points of the current population. Moreover, $\hat{\mathbf{y}}$ denotes the predicted output value, $\hat{s}_1$ and $\hat{s}_2$ denote standard deviations attributed to the prediction. In the first example (left) the $\mathrm{PoI}_{0.5}$ criterion rejects the new point while the MLI criterion does not. Contrary to this, in the second example the MLI rejects the new solution while the $\mathrm{PoI}_{0.5}$ criterion does not.

even under the condition that all random variables realize within the confidence interval boxes.

It turned out in the studies [EN04a] that the filters are not very effective for speeding up search. Either they hardly select any solution (in case of the P_ω-filter) or they hardly reject any solution (in case of the R_ω-filter). However, as from a theoretical point of view the design of these filters seem to be interesting, we still hope to find a scenario of metamodel-assisted optimization, where they are useful. However, in such a scenario, the separability of sets must be much easier as that seems to be the case for the approximations given in the typical populations of the MAES.

7.3 Studies on artificial test problems

Next, experimental studies with the metamodel-assisted EMOA on artificial test problems will be discussed. These studies are meant to prove the feasibility of the new approach on simple test problems. Later, in chapter 8, we assess the performance of the metamodel-assisted EMOA on representative test-problems from design optimization.

7.3.1 Metamodel-assisted non-dominated sorting genetic algorithm

Firstly, we will discuss the behavior of the metamodel-assisted NSGA-II. The feasibility of this is proven on the 10-dimensional generalized Schaffer problems A.6 proposed by Emmerich [Emm05]. For these problems, the Pareto front curvature depends on the

choice of the parameter γ. By setting $\gamma > 1$, a convex Pareto front is the solution of the problem. Whenever $\gamma < 1$, the Pareto front is concave. By setting $\gamma = 1$, the Pareto front gets linear. Discussion on measures for the convexity of Pareto fronts and its relevance for Pareto optimization can be found in [Bow76, CC77, AP96].

Figure 7.3.3 (convex Pareto front), figure 7.3.4 (linear Pareto front), and figure 7.3.5 (concave Pareto front) display results on differently shaped Pareto fronts. For the statistical comparison, 50% attainment surfaces were calculated. Here, 50% attainment surfaces are defined as follows:

Let $P_1, \ldots, P_n$ denote n pareto front approximations obtained with the same algorithm but different random seeds. Then a point $\mathbf{y} \in P_1 \cup \cdots \cup P_N$ belongs to the 50% attainment surface, if and only if $\mathbf{y} \preceq P_{i_1} \wedge \ldots \ \mathbf{y} \preceq P_{i_K}$ for any subset $P_{i_1}, \ldots, P_{i_k}$ of $k = (n-1)/2$ approximations to Pareto optimal sets. In the 50% attainment surface plots we consider only non-dominated solutions among the solutions that belong to the 50%-attainment surface. The motivation to use these points for average plots is that they mark the boundary of the space covered by the Pareto set in at least half of the run.

This corresponds to the non-constructive definition of Fonseca, who stated that 50% attainment surface consist of *'goal vectors, which each on its own, would have a 50% chance of being attained'* ([Fon95], page 107).

All experiments have been conducted with an initial step-size of 1. The $(20 + 20 < 100)$-NSGA-II (with mean value, lb_ω, ExI and PoI filter) was opposed to the $(20 + 100)$-NSGA-II and the $(20 + 20)$-NSGA-II. The same variation procedure to that used in the single-objective ES was employed. In the multi-objective case, a larger population size was used, in order capture a greater variety of solutions.

On the average, pre-screening through the mean value is less successful than that through the confidence information. The ExI criterion yields the best performance, followed by the LBI and PoI criteria. In particular, on the concave and linear problems, the ExI criterion leads to significantly better results. However, we note that - unlike the PoI and the LBI criteria - the performance of the ExI criterion depends on the choice of the reference point, which was set to $\mathbf{f}^{max} = (20, 20)^T$ for the given problems. Furthermore, the cost for computing the ExI criterion is significantly higher than that associated with the LBI and PoI filter. Hence, the two latter criteria should be considered as efficient pre-screening alternatives. In all problems examined, the performance deviation between all metamodel-assisted NSGA-II and the two versions of the standard NSGA-II is significant.

In addition, different variants of the NSGA-II and its metamodel-assisted NSGA-II, were tested on the ZDT1 and ZDT2 function. For all tested variants of the metamodel assisted NSGA-II the remaining non-dominated hypervolume in the box $[\mathbf{f}_{min}, \mathbf{f}_{max}]$ was measured, meaning that lower values of the remaining hypervolume measure correspond with a better performance of the algorithm. For the ZDT1 and ZDT2 function the values for the $\mathbf{f}_{min} = (0, 0)^T$ and $\mathbf{f}_{max} = (10, 10)^T$ was set.

The convergence history of different MA-EMOA versions on these functions is described in figures 7.3.6 and 7.3.7. All results are averaged runs for the metamodel-assisted NSGA-II with different pre-screening criteria. For each strategy variant 20 runs have been performed and the median of the non-dominated hypervolume of all solutions found so far was measured after each 50 evaluations of the two objective functions. The results

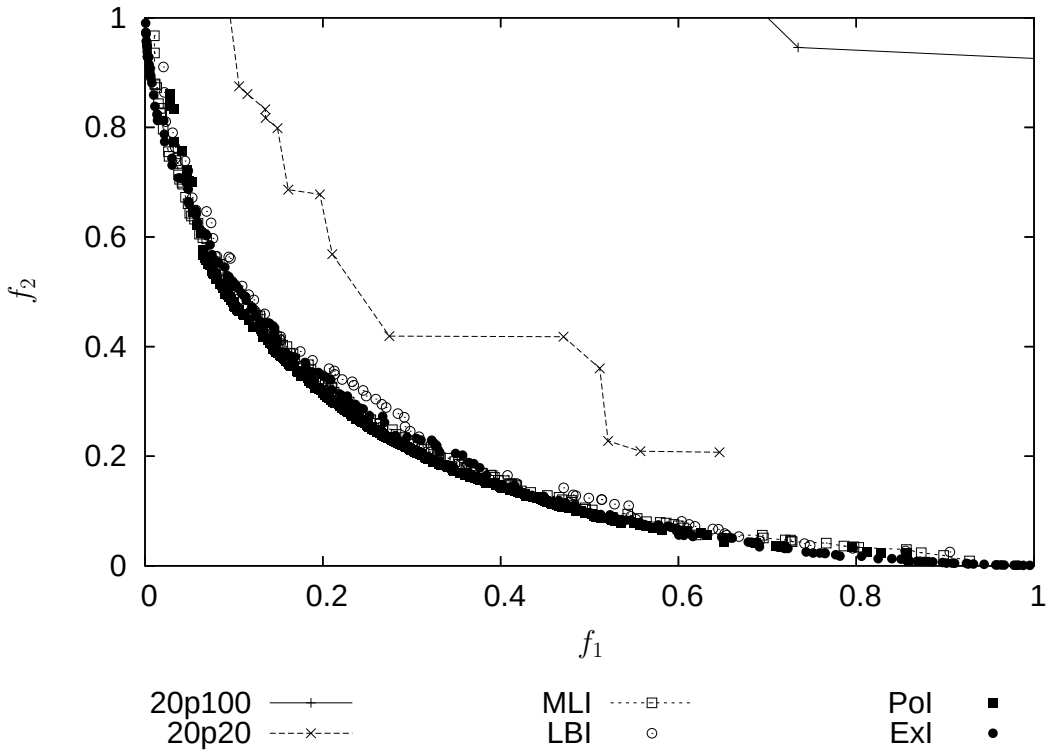


Figure 7.3.3: Approximation to a convex Pareto front. The 50% attainment surface on the 10-dim. generalized Schaffer problem with $\gamma = 2$ is displayed (10 runs, 1000 evaluations).

on this problem indicate that the lb_ω filter performs very satisfactory for convex and concave problems. On average, pre-screening through the mean value is less successful than pre-screening using the lower confidence bound criterion. It also seems that the use of the integral-based PoI and ExI criteria lead to worse results than the use of the lower confidence bound criterion. The margin between the results obtained with the lower bound filter and the other strategies is most significant for the concave problem. This can be explained by the characteristic of the lb_ω filter to reward solutions in unexplored regions of the search space and therefore also of the solution space.

In addition, figures 7.3.6 - 7.3.7 allow to compare the convergence dynamics of the tested algorithms. The acceleration due to the use of metamodels is especially high in the beginning of the run. In the long run the EMOA without metamodel-assistance catch up and only EMOA, using the lb_ω filter, can keep their margin.

7.3.2 Metamodel-assisted S-metric selection algorithm

So far, the metamodel-assisted SMS-EMOA has been tested on examples from airfoil design optimization [EBN05, NBE05] part of them will be discussed in chapter 8. Next, we present a test study on the EBN family of functions (cf. appendix [EBN05]),in order to assess the quality of results on Pareto fronts of different curvature. The initial population was equal for each run and sampled in the search space $\mathbb{S} = [0, 10]^{10}$.

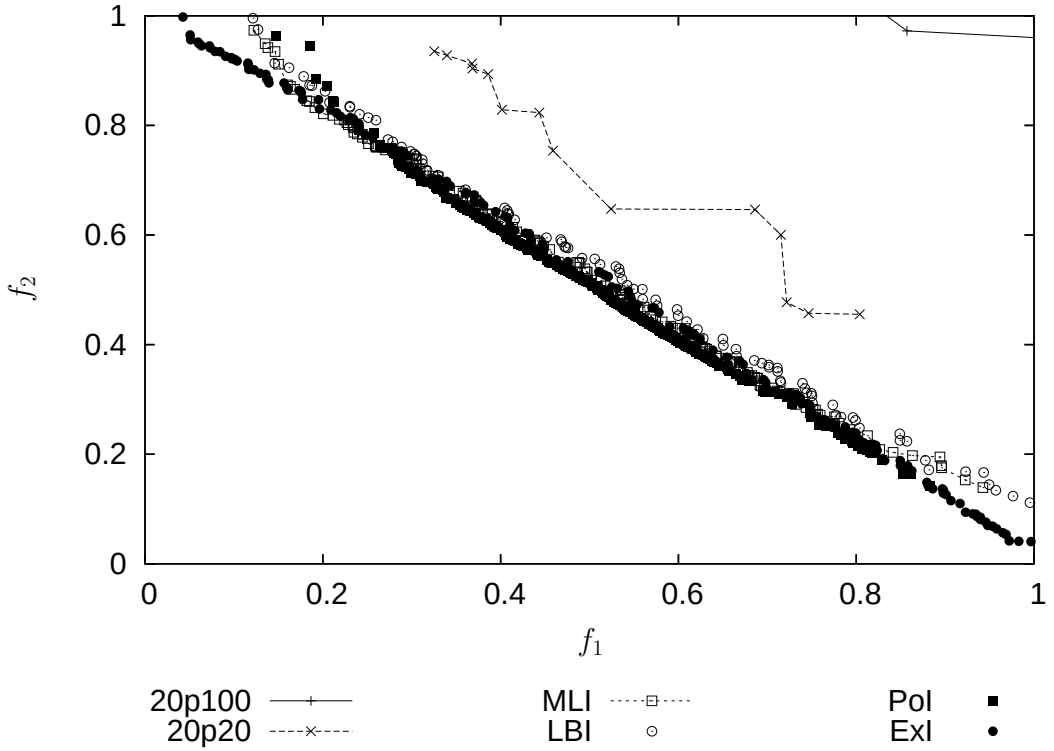


Figure 7.3.4: Approximation to a linear Pareto front. The 50% attainment surface on the 10-dim. generalized Schaffer problem with $\gamma = 1$ is displayed (10 runs, 1000 evaluations).

Table 7.3.1 provides a detailed description of the results. It summarizes the results of five runs for each combination of an algorithm and a test problem. Each run was stopped after 1000 evaluations of the vector valued objective function. From the small figures in the convergence value it can be concluded that almost all strategies managed to come close to the Pareto front. Table provides 7.3.2 a summary of this table, focussing on relative rankings of the strategy. Here it gets apparent that the metamodel-assisted EMOA using the lb_ω filter almost always performs better than other EMOA, if the dominated hypervolume is taken as the performance measure. Another observation that has been made was that the EMOA using the PoI criterion in the pre-selection tends to concentrate search points in a certain region. Though this leads to a much better convergence, this strategy fails to achieve a good value for the hypervolume and thus to provide a good coverage of the region near the Pareto front. The behavior of the mean value criterion and of the ExI criterion is characterized by a similar behavior. As a conclusion, we found that in order to achieve a diverse set on the Pareto front the LBI criterion seems to be best suited.

It has to be admitted, that the number of results presented is far from being sufficient in order to cover all common situations in which multi-objective MAES might be used. However, the results prove that the assistance by a metamodel can accelerate standard EMOA and also the SMS-EMOA. It also became apparent that the explorative power of the metamodel-assisted EMOA is especially high if the lb_ω criterion is employed, that best serves to reward solutions in yet unexplored regions of the search space.

Test-function	Algorithm	Convergence measure			$\mathcal{S}$ measure		
		Average	Std. dev.	Rank	Average	Std. dev.	Rank
strongly convex ($\gamma = 4$)	nokrig	0.0007	0.0001	2	0.9738	0.0033	4
	Mean	0.0014	0.0019	5	0.9782	0.0031	2
	lb	0.0009	0.0001	3	**0.9798**	0.0011	1
	PoI	**0.0002**	0.0002	1	0.9711	0.0041	5
	ExI	0.0013	0.0008	4	0.9781	0.0024	3
convex ($\gamma = 2$)	nokrig	**0.0011**	0.0001	1	0.9781	0.0024	3
	Mean	0.0012	0.0002	2	0.7860	0.0163	3
	Lb	0.0015	0.0009	4	**0.7921**	0.0087	1
	Poi	0.0012	0.0002	2	0.7659	0.0141	4
	ExI	0.0013	0.0002	3	0.7900	0.0078	2
linear ($\gamma = 1$)	Nokrig	0.0013	0.0001	2	0.4048	0.0130	5
	Mean	0.0013	0.0003	2	0.4348	0.0105	2
	Lb	0.0013	0.0003	2	**0.4363**	0.0130	1
	PoI	**0.0012**	0.0002	1	0.4038	0.0183	4
	ExI	0.0015	0.0005	3	0.4303	0.0141	3
concave ($\gamma = \frac{1}{2}$)	nokrig	0.0009	0.0002	2	0.1556	0.0065	5
	Mean	0.0015	0.0006	4	**0.1720**	0.0066	1
	Lb	0.0010	0.0001	3	0.1712	0.0046	2
	PoI	**0.0008**	0.0001	1	0.1560	0.0061	4
	ExI	0.0009	0.0001	2	0.1690	0.0081	3

Table 7.3.1: Results of various metamodel-assisted SMS-EMOA on the EBN family of multi-objective test functions.

Algorithm	Convergence measure Ranks				Convergence measure $\sum$ of ranks	$\mathcal{S}$ measure Ranks				$\mathcal{S}$ measure $\sum$ of ranks
Nokrig	2	1	2	2	7	4	5	5	5	19
Mean	5	2	2	4	13	2	3	2	1	8
Lb	3	4	2	2	11	1	1	1	2	**5**
PoI	2	1	1	1	**5**	4	4	4	5	17
ExI	2	3	3	4	12	3	2	3	3	11

Table 7.3.2: Ranks and sum of ranks from table 7.3.1.

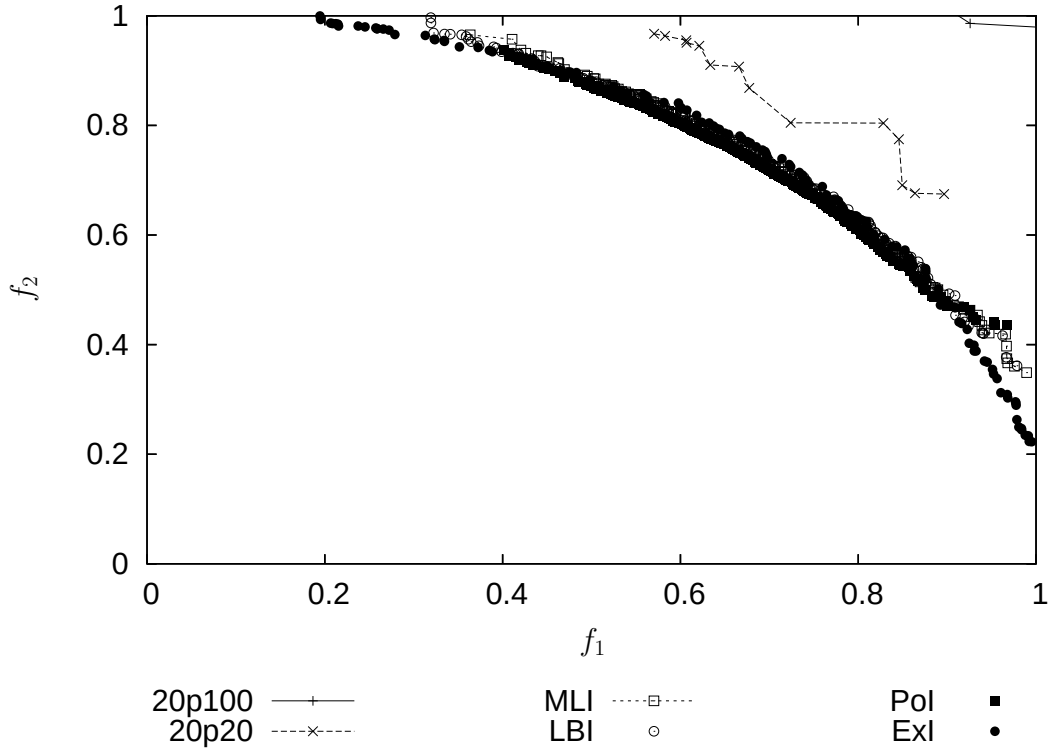


Figure 7.3.5: Approximation to a concave Pareto front. The 50% attainment surface on the 10-dim. generalized Schaffer problem with $\gamma = 0.5$ is displayed (10 runs, 1000 evaluations).

7.4 Conclusions

The NSGA-II algorithm as well as the SMS-EMOA were augmented with metamodel-assistance. For the first time criteria like the probability of improvement and the expected improvement have been generalized for Pareto optimization. This was possible by defining the improvement by means of the increase in the dominated hypervolume of the current population. In particular the generalization of the PoI criterion turned out to be very elegant, because neither it demanded for a reference point, nor did it matter if the expected value $\hat{\mathbf{y}}(\mathbf{x})$ of newly generated points $\mathbf{x}$ is part of the non-dominated set of the current elite population. A conceptual comparison of the new criteria have been given that revealed that some of the invariant properties of the IPE filters get lost when it comes to multi-objective optimizations. So are the equivalence of the MLI criterion to the PoI_τ criterion with $\tau = 0.5$. First results on 10-D problems of different curvature (convex, linear, quadratic) have been conducted in order to assess the performance of the proposed algorithms. It was found that rewarding solutions with large confidence margins helps to achieve an improved coverage of the Pareto fronts. However, there is a trade-off between achieving a good value of the convergence metric and achieving a good coverage of the Pareto fronts. In the next chapter, further examples for the application of metamodel-assisted optimization methods will be given.

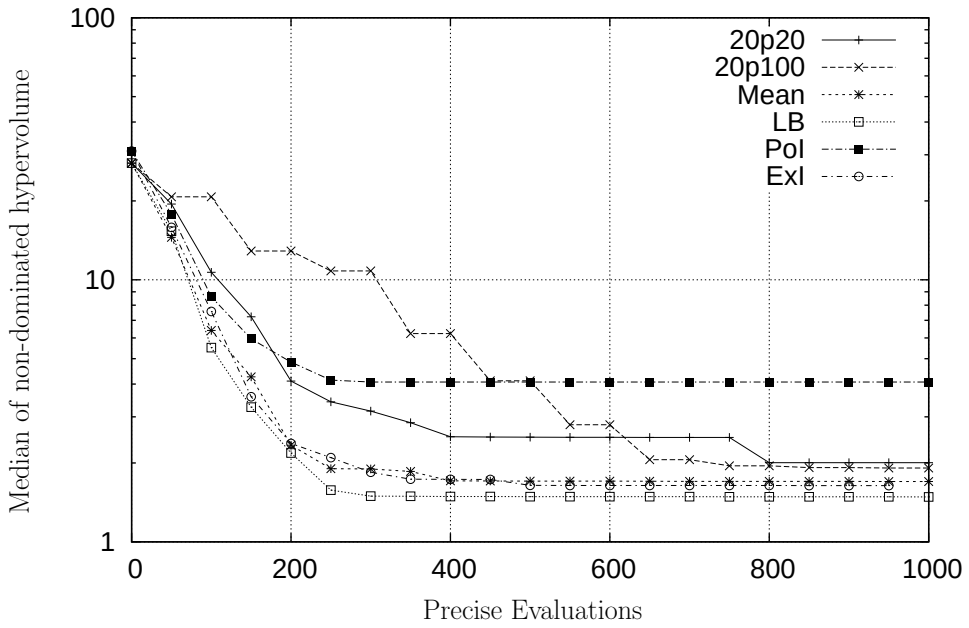


Figure 7.3.6: Median of the non-dominated hypervolume value of different EA on the 10-D ZDT1 function.

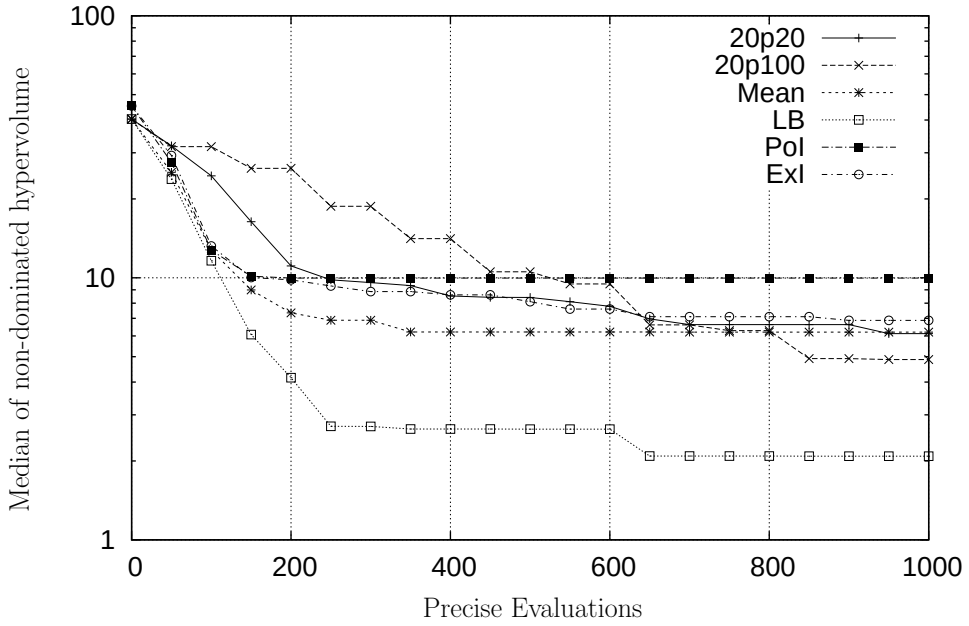


Figure 7.3.7: Median of the non-dominated hypervolume of different EA on the 10-D ZDT2 function.

8 Applications in industrial design optimization

The final test of a theory is its capacity to solve the problem which originated it.

G.B. Dantzig

Finally, some results in industrial design optimization are reported. These results provide examples for the successful application of metamodel-assisted evolutionary algorithms for real-world problems. The case studies were carried out in collaboration with partners from industry and engineering departments in academia. They provided evaluation tools and defined the goals of the optimization studies.

In the engineering domain, the term design optimization usually refers to optimization studies carried out in the detailed engineering stage of a project. In contrast to the conceptual engineering phase the requirements on the accuracy of computer models applied in this phase are very high. Thus evaluations of designs on a computer tend to be very time consuming and may take several minutes or even hours.

In section 8.1 we report on applications of the MAES in single-objective optimization. A problem from electromagnetic compatibility design serves as an example for this problem domain. Then, in section 8.2, we turn to single-objective optimization with constraints, discussing the optimization of a gas turbine blade casting process. Moreover, section 8.3 presents results on an airfoil re-design problem with two objective functions. Finally, in section 8.3.2, the design optimization of an airfoil geometry is presented, dealing with three objectives as well as with nonlinear constraints.

8.1 Single-objective design optimization

The metamodel-assisted evolution strategy was applied in three fields of single-objective design optimization. First, we report on results in the domain of electromagnetic compatibility design. Then, we turn to problems in metal forging, and finally we discuss studies that were conducted in airfoil design.

8.1.1 Electromagnetic compatibility design

The connection of high voltage cables is done by a sleeve. In the connection area the geometry of the sleeve can be changed to improve the *electromagnetic compatibility* (EMC). By doing so, technical restrictions of maximal field forces have to be kept. The field force can be controlled by using layers of different materials and geometry. Free parameters in such a configuration are for example the thickness of the layers and the expansion of the chambers.

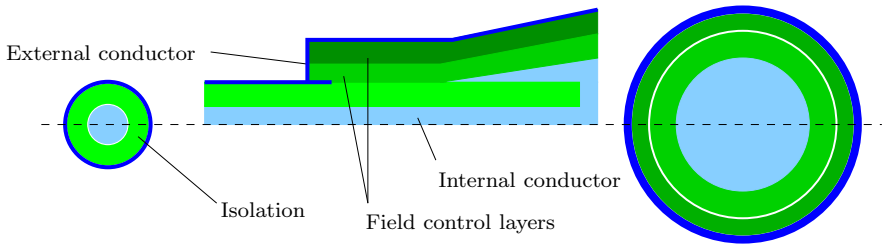


Figure 8.1.1: Schematic cuts through a high voltage sleeve that is used for connecting cables. In order to control the electromagnetic field, Field control layers consisting of different materials are integrated into the sleeve. They are part of the isolation that separates the internal conductor from the external conductor.

In collaboration with the Electrical Engineering Department of the University of Dortmund a study on the shape optimization of a sleeve was conducted. Our aim in the case study was to find a geometry that leads to an improved quality of the sleeve with regard to its electromagnetic compatibility. A set of 10 design variables was identified. All design variables were related to the geometry of the sleeve. Figure 8.1.2 explains the meaning of these parameters. The set of geometry parameters was subdivided into five radial sizing parameters (rs1, rs2, ri, ra, rf) and five axial sizing parameters (as, aa, af1, af2, a0).

To be comparable with previous studies, the design variables have been restricted by means of interval bounds:

sizing variable	lower bound /$[m]$	upper bound /$[m]$
ri	0.00265	0.00515
ra	0.01404	0.02945
rf	0.06686	0.077
rs1	0.09502	0.09696
rs2	0.03357	0.04144
af1	0.01545	0.075991
af2	0.119641	0.170552
aa	0.193052	0.215582
as	0.263373	0.271323
a0	0.00462	0.01467

In the evolutionary algorithm, every time an infeasible solution was sampled, it was rejected and sampled anew, unless λ feasible solutions were generated.

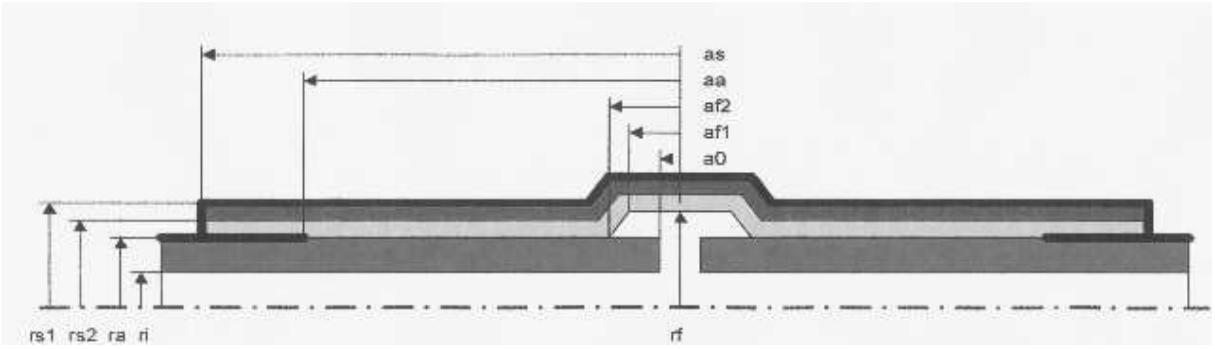


Figure 8.1.2: An axial cut through a high voltage sleeve. Five radial sizing parameters (rs1, rs2, ri, ra, rf) and five axial sizing parameters (as, aa, af1, af2, a0) determine the geometry of the sleeve.

The evaluation of the design was based on finite integration methods that solve the Maxwell equations, given the boundary conditions of the problem that resulted from the geometry and material parameters. A detailed description of the simulation procedures is found in [Var03]. In order to accelerate the evaluation procedure, the computation method makes use of the rotational symmetry of the sleeve. Despite this effort on decreasing computational time, the simulator-based evaluation of the objective function still took about 10 minutes on the available computing system (Pentium III, 600 MHz). This was orders of magnitudes higher than the effort for an approximate objective function evaluation and made the use of metamodeling techniques attractive.

The objective in our case study was to minimize the maximal value of the electromagnetic field along the 0.95 equipotential line [Var03]. This value is an important indicator of the electromagnetic field compatibility. After each finite element simulation, the quality value was calculated from the values of the electromagnetic field. For all positions of a finite element mesh the electromagnetical field potential was computed. Then for the grid positions that are nearest to the 95% equipotential line, the maximum of the field force was detected. Figure 8.1.5 displays the equipotential lines for an example solution.

For the geometry optimization several variants of the ES were tested. The averaged results for twenty runs with the $(5, 5, 100)$-ES and the $(5, 5, 5 < 100)$-MAES with lower confidence bound and mean value filter are displayed in figure 8.1.3. The best results were found with the $(15 + 15 < 100)$-MAES. While the strategy using a mean value criterion for pre-screening already found a good result, the use of the lower confidence bound criterion further improved it. The reliability of the MAES on this problem was tested by repeating the run 10 times, with different random seeds. In the series of runs we observed a low deviation from the average behavior. Figure 8.1.4 displays the 0.25 and 0.75 quartiles of the observed fitness histories.

Finally, we took a closer look at the obtained result. The electromagnetic field force along the 95% equipotential line, the maximum of which had to be minimized, is displayed in figure 8.1.6. A significant reduction of the maximal field force of about 7% in comparison to the baseline design was achieved.

In conclusion, the results on this test problem indicated, that the MAES is an interesting

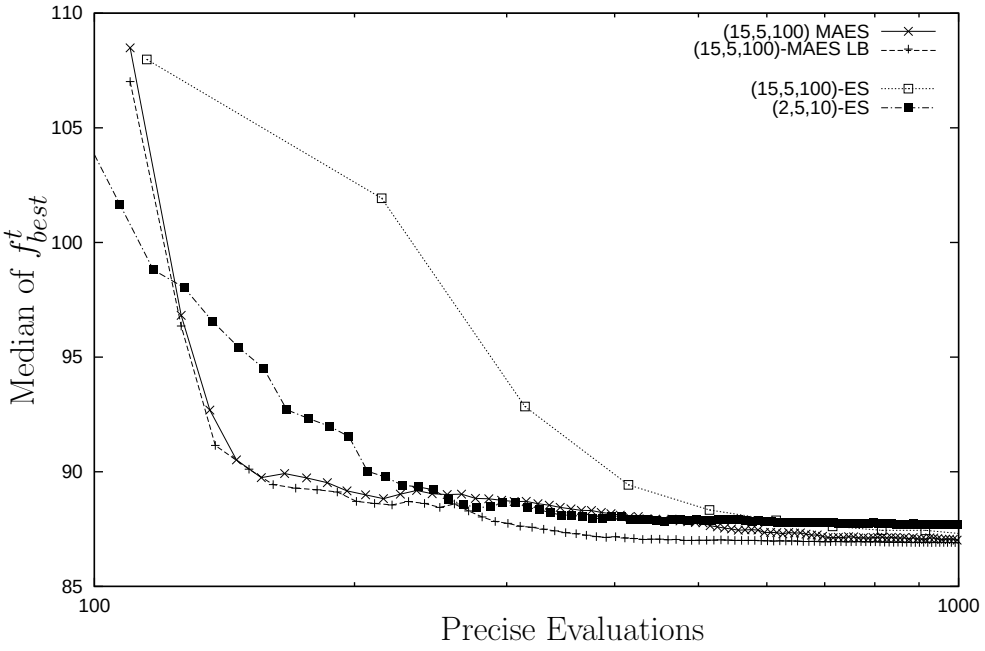


Figure 8.1.3: Averaged histories for different strategies for the electromagnetic compatibility optimization.

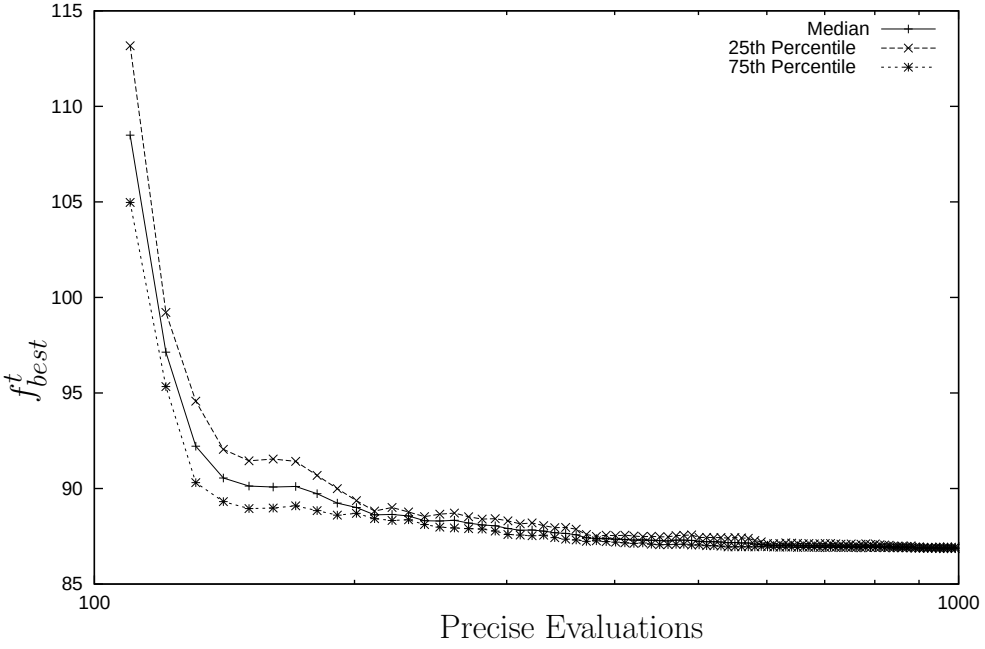


Figure 8.1.4: The median, 25th and 75th percentile of the observed histories.

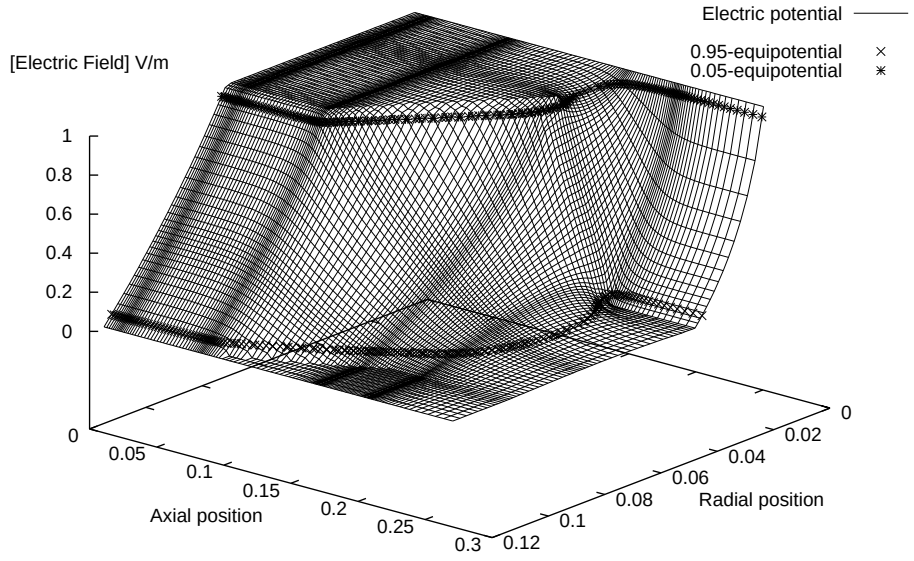


Figure 8.1.5: Visualization of the shape for the electromagnetic field obtained with the MAES and the 95% and 5% equipotential line.

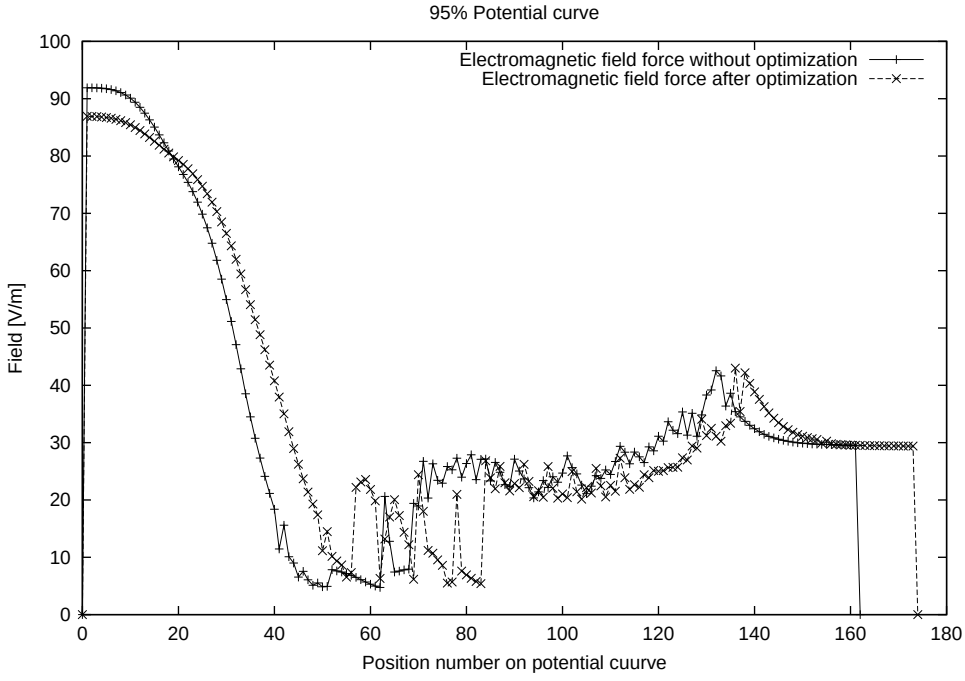


Figure 8.1.6: The electromagnetical field in [eV] measured on the 95% equipotential curve.

Simulation	Initial	Adjoint quasi-Newton method	$(5 + 5 < 25)$-MAES
Coarse mesh	1.48	1.22 (-18%)	1.18 (-20%)
Fine mesh	1.52	1.26 (-17%)	1.23 (-19%)

Table 8.1.1: Numerical results from the optimization of a forging process with the MAES reported in [DF04]. A coarse grained model was used during the optimization run. The finally obtained solution was re-evaluated by means of a fine grained model. The re-evaluation took about 40 hours of time for each design. The results indicate the good performance of the MAES in comparison to a quasi-Newton method.

approach for optimization in the field of electromagnetic compatibility. Despite these encouraging first results, further studies will be needed, in order to establish the MAES as a tool in the field of electromagnetic compatibility design.

8.1.2 Metal forging design

The MAES was used for other optimization studies with a single-objective as well. Studies for 3D forging design were conducted by Fourment, Do and Larroussi [DFL04], using the MAES implementation that has been proposed in chapter 4. In contrast to the problems that were described previously, they solved a very low dimensional problem with only three variables and a low number of 40 objective function evaluations. For the solution of their problem they performed 10 generations with a $(1+4 < 20)$-MAES. Three parameters that determine the shape of an piece of metal (cf. figure 8.1.7 and figure 8.1.8) were due to optimization. They found a much better result than with a problem specific gradient-based method.

The simulations during the optimization run were carried out with a coarse grained computer model that needed about one hour for a single objective function evaluation. The final result was verified by means of a fine grained computer model that needed 40 hours for a single objective function evaluation. In the verification step it turned out that not only the numerical value obtained for the coarse grained model improved significantly, but also the value obtained with the fine grained model. Recently, new results on this problem were reported by Fourment and Do [DF04]. In figure 8.1.1 some of these recent results are displayed. These results reconfirm the high performance of the MAES for this problem domain.

8.1.3 Applications in aerospace and turbo-machinery design

Further case studies in single-objective optimization with the MAES have been carried out in collaboration with A. Giotis and K. Giannakoglou from the National Technical University Athens on applications in the application domain of aerospace and turbo-machinery design [EGÖ$^+$02, GEN$^+$01]. The case study reported in [EGÖ$^+$02] dealt with the re-design of an airfoil. In the test runs performed, the MAES using the lower confidence bound pre-screening outperformed the MAES that worked with mean value pre-screening. In Giotis et al. [GEN$^+$01] the MAES was applied for the design of a turbine blade in

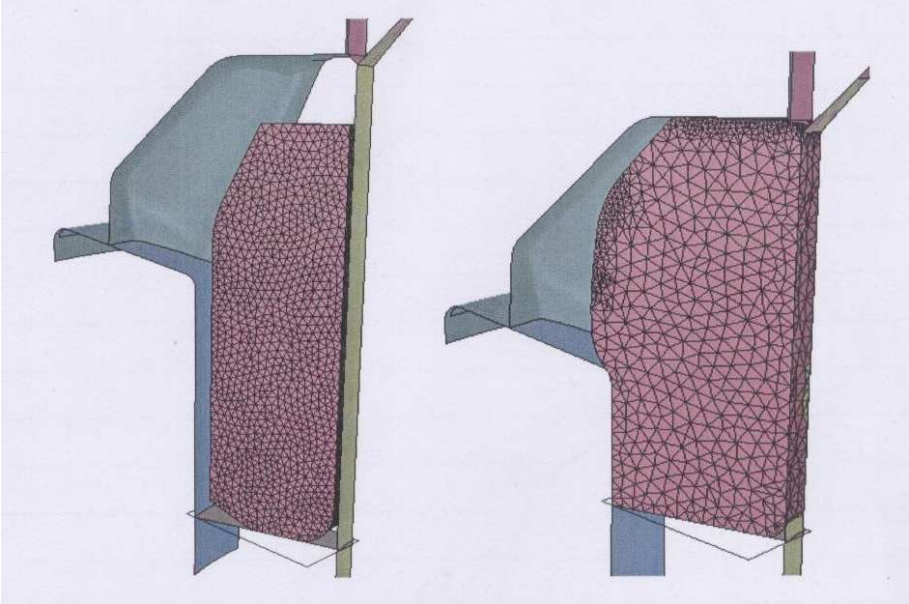


Figure 8.1.7: Metal forming simulation with FORGE-3D^{TM}: The figure on the left hand side shows the 3D visualization of the initial piece of metal before it gets compressed in the forging device. The figure on the right hand side depicts the piece of metal in an intermediate stage of the forging process.

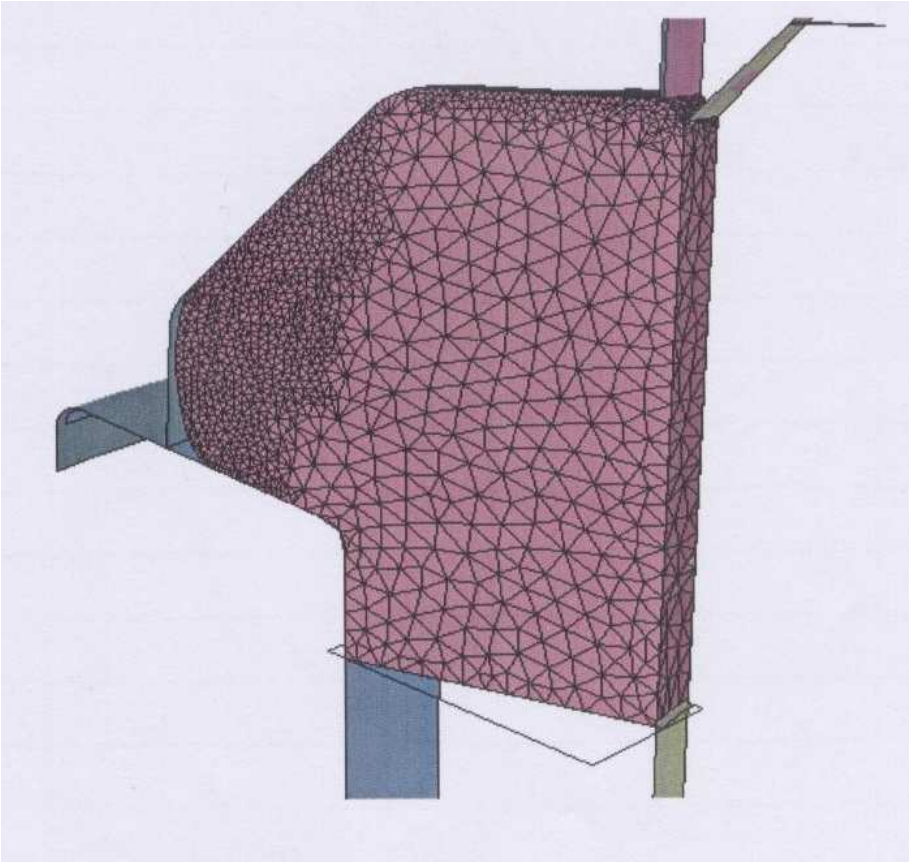


Figure 8.1.8: Deformed piece of metal after the forging process. It is desired that the form of the piece of metal adapts to the shape of the preform.

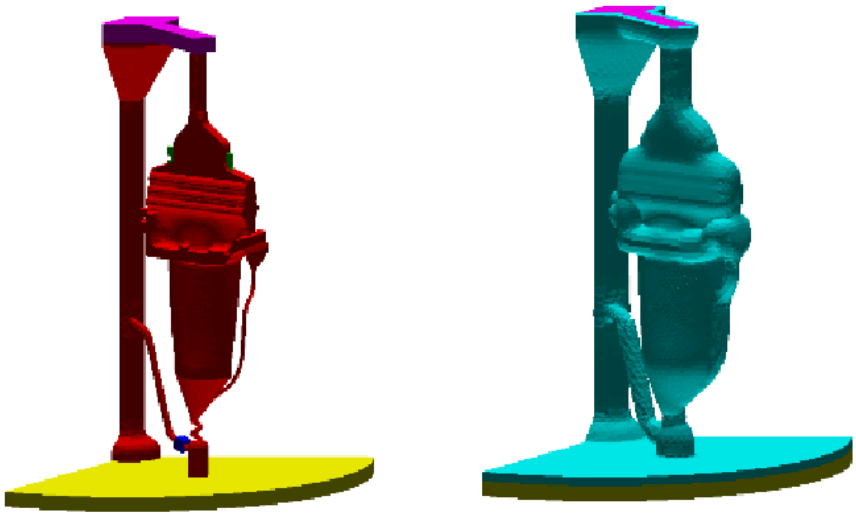


Figure 8.2.9: A gas turbine blade before (left) and after (right) the casting process.

a compressor cascade. They compared its performance to a metamodel-assisted genetic algorithm. The genetic algorithm was a real-coded genetic algorithm with truncation selection and a constant mutation step-size. The results indicated a better performance of the approach based on evolution strategies, which they explained by the capability of the latter algorithm to adapt its mutation step-size.

In section 8.3 and section 8.3.2 further results on airfoil optimization are discussed. There, the focus is on multi-objective problem formulations.

8.2 Optimization of a casting process for gas-turbine blades

Another example for the successful application of metamodel-assisted evolution strategies is the optimization of gas turbine blade casting processes, that was carried out in collaboration with the research institute ACCESS e.V., Aachen, whose area of expertise is the simulation and optimization of solidification processes.

8.2.1 Problem definition

During the manufacturing of a gas turbine, the casting of turbine blades is the most expensive process. This makes an optimization of this process very interesting for industry. The highest gas turbine efficiency is achieved today with single-crystal (SX) and directionally solidified (DS) blading material, commonly produced in a Bridgman furnace. A sketch of this solidification process is given in figure 8.2.10. Basically, the turbine blade is withdrawn slowly from a radiation heater and the blading material solidifies gradually on its surface. Both, the heating temperatures and the withdrawal speed can be controlled by the casting engineer during the process. It is also possible to control it automatically

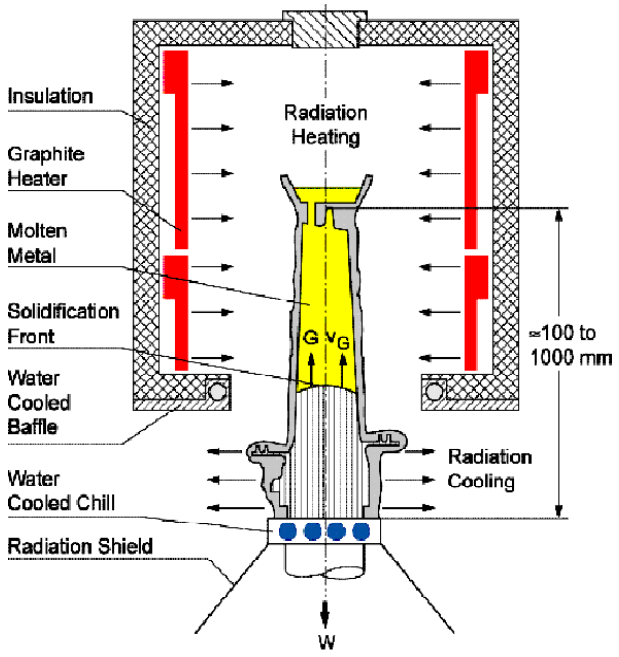


Figure 8.2.10: Schematic drawing of the Bridgman casting process. The turbine blade is slowly withdrawn from a radiation heater. As it leaves the heater, the surface is rapidly cooled down by water cooling and radiation, causing the superalloy mould on its surface to solidify in a directional manner.

by describing the changing withdrawal velocity by a poly-line with eleven parameters.

The design optimization aimed at the minimization of the total process time subject to several constraints that stem from the quality requirements for the material. A constraint was also formulated for the overall process time, in order keep the process time within a reasonable range. In summary, the following constraint functions were formulated:

- The total process time should not exceed 5000 seconds.
- The probability for freckles[1] formation should stay below a threshold value for each node. In our studies we demand for a probability of zero.
- Dendritic crystal growth should be achieved, i. e. the G/v value should stay below 600. Here G denotes the temperature gradient in withdrawal direction, and v denotes the solidification speed (cf. figure 8.2.11).
- The local curvature of the solidification front should be kept within a predefined range (cf. figure 8.2.11, right) for each node. The angle between the normal vector and the vector in direction of the withdrawal should stay below 20 degrees.

The latter three constraints are local constraints, meaning that they are computed for every node of the finite volume mesh. Since the mesh comprises a large number of about 50000 nodes, it had been decided to integrate the information on local constraint violations. This was done by computing a single constraint value for each of the three classes

[1]Freckles are small defects of the blading material on the turbine blade's surface.

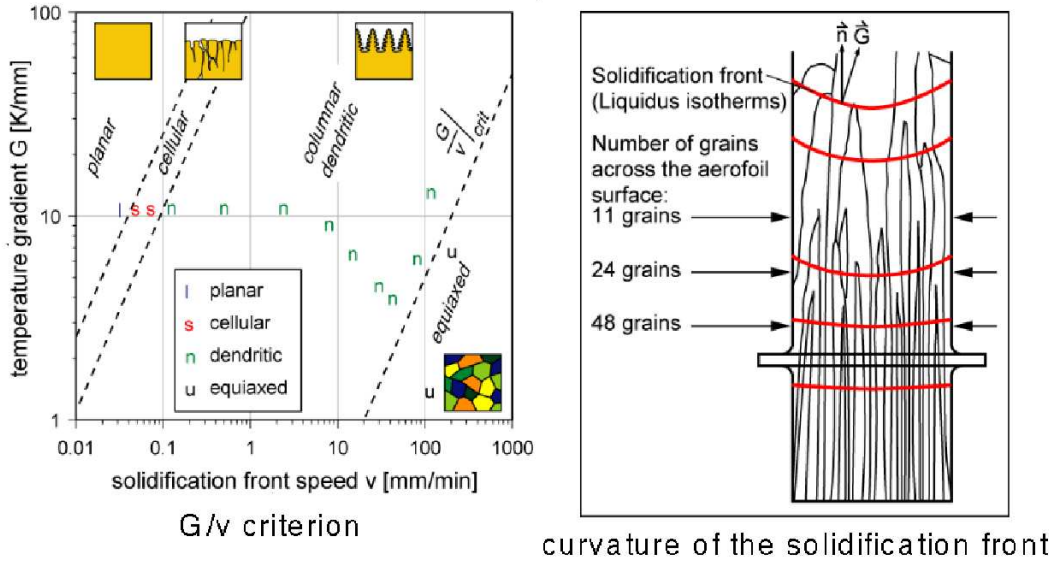


Figure 8.2.11: Visualization of two local constraints. The figure on the left hand side shows the G/v criterion, where G denotes the temperature gradient and v the solidification front speed, both measured in the direction of the withdrawal. The value of G/v needs to be in a certain range in order to guarantee the desired dendritic growth. The figure on the right hand side visualizes the curvature constraint. In order to achieve the desired fine grained structure on the surface, the local curvature of the solidification fronts (liquidus isotherms) should stay below a certain threshold value.

of local constraints. Emmerich and Jakumeit [EJ03], proposed to count the number of *bad nodes* on the surface of the turbine blade. By bad nodes they denoted those nodes for which local constraints were violated.

An improved version of this penalizing procedure was suggested by Emmerich and Jakumeit [EJ04]. There, they weighted each 'bad' node by its control volume, and thereby put more emphasize to constraint violations in larger control volumes. Furthermore, distinct weighting factors were attributed to the three different types of local constraints, reflecting their different importance [EJ04]. The sum of total weights of the nodes serve as penalty term for the objective function.

8.2.2 Optimization algorithms

First, studies were conducted for the optimization of a *dummy blade*, i. e. a blade with a simplified geometry. The complete design evaluation of one process variant for the casting of this blade lasted one hour, in contrast to eight hours running time for the evaluation of an blade casting process for a blade with realistic geometry.

Various optimization strategies have been tried for the optimization of the dummy blade:

- Kriging Monte Carlo strategy [JHN05]

- Downhill simplex strategy by Nelder and Mead [Sch95]
- Evolution strategy with derandomized step size control (DES)
- Metamodel-assisted DES (MA-DES)

The evolution strategy with derandomized step size control that was used here, differed from the evolution strategy that was introduced in section 3.5 by the way it adapts the mutation step-sizes. The cumulative step-size adaptation algorithm (CSA) of Ostermeier et al. [OGH94] was applied here, that allows the adaptation of individual standard deviations for very small population sizes (e. g. $\mu = 1, \lambda = 7$). We will not go into details about this procedure here, since a sufficiently detailed explanation of the CSA would extend the scope of this section. However, it shall be noted that the metamodel was integrated in the same manner than described in chapter 4. In each generation, the subset of $\nu = 4$ most promising solutions among the λ offspring was chosen, evaluated precisely, and considered for selection. The resulting algorithm was termed metamodel-assisted DES (MA-DES). Two different versions of the MAES were tried: The MAES with mean value pre-screening and the MAES with lower confidence bound pre-screening.

8.2.3 Numerical results

In figure 8.2.12 the convergence dynamics of the four different optimization strategies are described. The MA-DES variants clearly outperform the conventional DES and the downhill simplex algorithm. Note that the use of confidence value does not lead to a better result for this example.

It can also be seen that the choice of the confidence factor $\omega = 0$ leads to slightly better results. An objective function value below 5000 cannot be found by any optimization strategy, i. e. no point was found, with none of the constraints violated. It is very likely that the change of the withdrawal profile is not enough to gain a turbine blade without 'bad' nodes.

The best solutions found for each strategy are depicted in figure 8.2.13. On each turbine blade the nodes with too high curvature or too low freckle tendency are marked with a specific color. Freckles can not be found on this simple turbine blade geometry. Clearly, the MA-DES could reduce the size of the regions with bad nodes best while keeping the process time below 5000 seconds.

More recent results for a realistic turbine blade with the weighted penalty function are reported by Emmerich and Jakumeit [EJ04]. The casting process was improved significantly leading to a process time that is about 20% lower than the design suggested by an expert. Again, the MAES outperformed other optimization strategies, like the downhill simplex.

8.3 Airfoil design optimization

Next, we present results with the metamodel-assisted SMS-EMOA(subsection 8.3.1) and metamodel-assisted NSGA-II (subsection 8.3.2) applied in multi-point airfoil design.

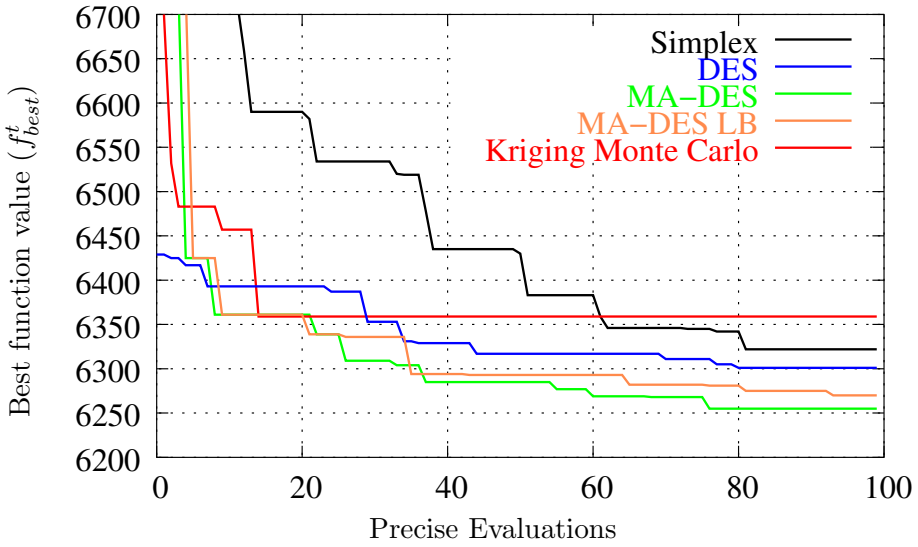


Figure 8.2.12: A comparison of different conventional and metamodel-assisted optimization strategies for optimization of the Bridgman casting process. The objective function value is plotted for each of the 100 precise evaluations. The metamodel-assisted strategies are significantly faster in the beginning of the search. The metamodel-assisted evolution strategies also lead to better final results.

8.3.1 Two-objective NACA airfoil re-design problem

The NACA airfoil re-design test-case is a well known test case from literature (cf. Naujoks et al. [NWTW02]). Two-objective functions need to be minimized simultaneously for this problem. They stem from the task to re-design two target airfoils that themselves are nearly optimal for given flow conditions. These flow conditions can be taken from table 8.3.1.

For both flow conditions the pressure distribution around the airfoil has been calculated. All design and simulation conditions, e. g. the flow models and mesh generation methods, have been fixed for the study. The following two-dimensional design problem had to be solved by the optimization:

$$f_{1,2}(s) = \int_0^1 (C_p(s) - C_{p,target_{1,2}}(s))^2 ds \rightarrow \min \tag{8.3.1}$$

Here C_p denotes the pressure distribution along the airfoil, with s being the arc-length.

In order to compare results, the found Pareto front approximations were averaged by means of a method described by Naujoks et al. [NWTW02]. A brief explanation of this method shall be given next: A bisector is drawn through the positive quadrant of the search space and equidistant lines that are parallel to this bisector are considered. In each run the points on the Pareto front with the shortest distance to these lines are considered for the calculation of the averaged front. If we receive at least three points out of five runs within a predefined maximum distance, these points are averaged to become a member of the averaged front. The resulting points of five runs from our studies can be seen next to the Pareto front of the runs in the lower right part of figure 8.3.14. This procedure

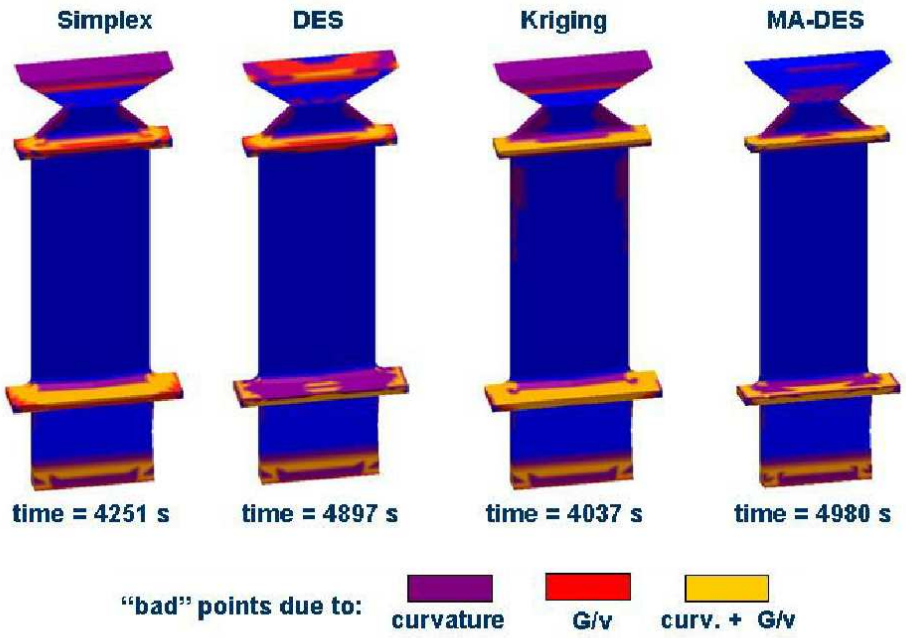


Figure 8.2.13: Comparison of the best turbine blades found with the different optimization strategies introduced in subsection 8.2.2. The "bad" points are due to a too high curvature, to low G/v value or both are plotted. The formation of freckles was not observed in this simple turbine blade.

does not reward high densities of points on the Pareto front, but could lead the engineer to focus on designs with significant changes.

In a case study the SMS-EMOA, the classical NSGA-II and the metamodel-assisted versions of the SMS-EMOA have been compared. Five runs for each algorithm in the comparison have been evaluated. In the left hand part of figure 8.3.15 the different dotted sets describe three out of the five Pareto fronts received from the different runs utilizing SMS-EMOA without function approximations. The line describes the received averaged Pareto front.

Figure 8.3.15 compares the averaged fronts received using SMS-EMOA with and without fitness function approximations. In addition, the best result achieved with a metamodel-assisted NSGA-II (taken from [EN04a]) has been included.

A clear superiority of the algorithms utilizing metamodels can be recognized. The averaged front without metamodel integration is the worst front all over the search space except for the upper left corner, the extreme f_2 flank of the front. In most other regions the SMS-EMOA with lb_ω filter seems to perform better than the other algorithms shortly followed by the results from the metamodel-assisted NSGA-II with lb_ω filter. The SMS-EMOA with mean value criterion yielded the worst front among those obtained with metamodel-assisted EA.

In the extreme f_2 flank of the front the results seem to be turned upside down. Here, the averaged front from runs without model integration achieved the best results. This result might be caused by the averaging technique. One run achieved outstanding results

Property	Case	High lift	Low drag
M_∞	$[-]$	0.20	0.77
Re_c	$[-]$	$5 \cdot 10^6$	10^7
α	$[^0]$	10.8	1.0

Table 8.3.2: Summarized design conditions for high lift (starting phase) and low drag (stationary phase). M_∞ denotes the Mach number. From the given Mach numbers, we conclude that both flight conditions are subsonic. Re_c denotes the dimensionless Reynolds number (c=chord length). Finally, α denotes the angle of attack.

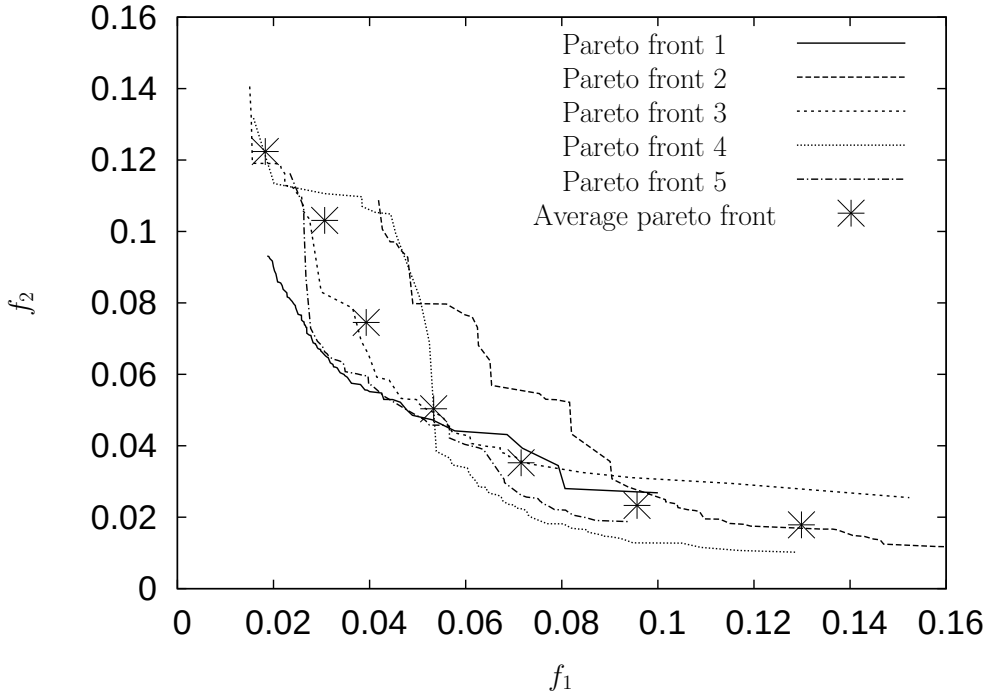


Figure 8.3.14: Example for averaging Pareto fronts.

here, that led to an unbalanced average point that is better than the averaged points of the other algorithms. This extreme effect could be avoided by averaging over much more than five runs.

Notice, that the lower lb_ω filter yielded better results than the mean value filter. This was also observed in [EN04a] and seems to be a general achievement, where more attention should be drawn to.

8.3.2 Multi-objective optimization with constraints: The RAE 2822 test case

Next, we discuss results on the RAE 2822 airfoil optimization with three flow conditions. This test case is particularly challenging, since it has more than two objectives and several

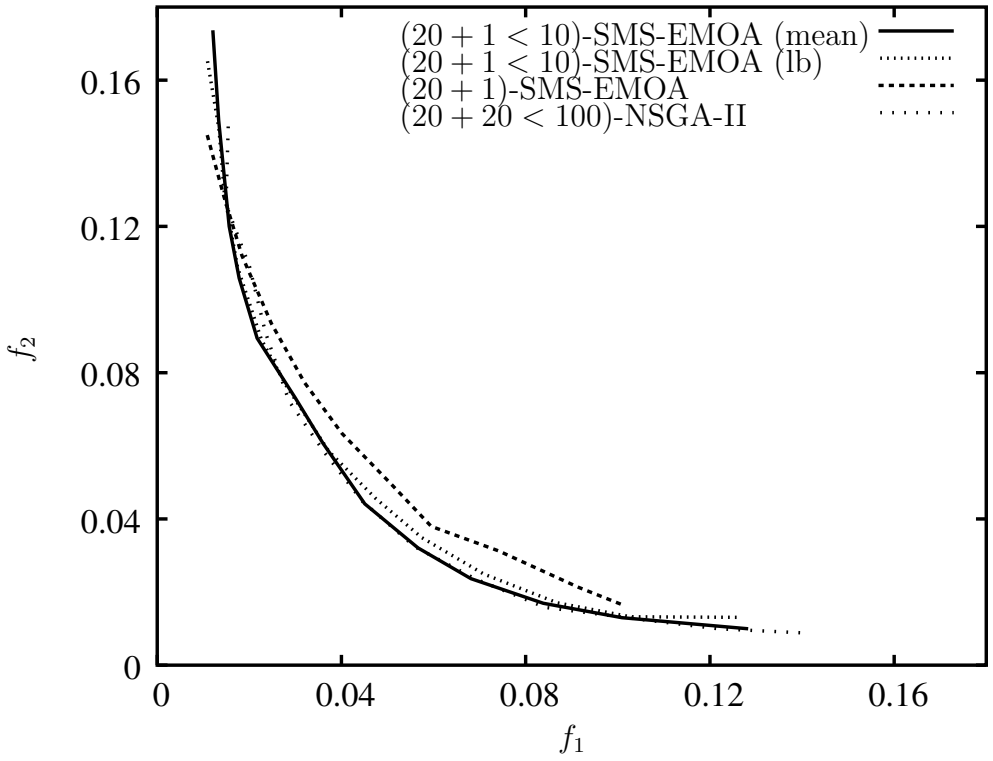


Figure 8.3.15: Averaged fronts for SMS-EMOA without using metamodels ((20+1 < 10)-SMS-EMOA (mean)), with metamodels and lower confidence bound filter ((20 + 1 < 10)-SMS-EMOA (lb)), mean value criterion ((20 + 1)-SMS-EMOA), and NSGA-II using metamodels and the lb_ω filter ((20 + 20 < 100)-NSGA-II).

	cruise	off-design 1	off-design 2
M	0.734	0.754	0.680
Re	$6.5 \cdot 10^6$	$6.2 \cdot 10^6$	$5.7 \cdot 10^6$
α	2.8	2.8	1.8
transition	3%	3%	11%

Table 8.3.3: Flow conditions for the RAE 2822 airfoil design problem.

implicit and explicit constraints. Over and above, it has already been studied extensively in literature. As it is shown next, it was possible to further improve the best design found for this problem by means of the metamodel-assisted NSGA-II.

The flow around the baseline design, the RAE 2822 airfoil, is calculated with respect to three different flow conditions, yielding different values for drag, lift and pitching moment for each of the flow conditions. The task is to minimize the drag values C_d^i while not losing lift and keep the pitching moment within a 2 % range. Here $i \in \{1, 2, 3\}$ corresponds to the three given flow conditions, one for cruising and two more off-design conditions. These conditions can be taken from table 8.3.3.

The aerodynamic constraints for lift C_l^i and pitching moment C_m^i read:

- $\forall i \in \{1, 2, 3\}: \quad C_l^i \geq C_{l,base}$ with $C_{l,base}$ being the lift coefficient of the baseline

RAE 2822 airfoil.

- $\forall i \in \{1,2,3\}: \quad C_m^i$ within +/- 2% of the pitching moment $C_{m,base}$ of the baseline RAE 2822 airfoil.

Furthermore, geometrical constraints have been defined:

- The thickness of the airfoil at 5% should be greater than or equal to the thickness at 5% of the baseline geometry.
- The maximum thickness should be greater than or equal to the maximum thickness of the baseline geometry.
- The leading edge radius should be greater than or equal to 90% of the leading edge radius of the baseline geometry.
- The trailing edge angle should be greater than or equal to 80% of the trailing edge angle of the baseline geometry.

The geometrical information about a proposed airfoil can be received from the simulation software just after the airfoil shape is generated. The whole time-consuming procedure of solving the flow and all post-processing tasks are not required to receive this information. Accordingly, the geometrical constraints are treated differently from the aeronautical ones, that require the costly flow calculation. This different treatment is described in detail later.

Again, the airfoil parametrization was done using Bezier weighting points. The y coordinates of these points serve as parameters for the optimization method. To be comparable with previous studies, three Bezier weighting points have been used for both surfaces of the airfoil, resulting in an optimization problem with 6 degrees of freedom. All other configurations and parameters concerning mesh generation, flow models in use are kept constant during the current investigation.

Note that for the RAE 2822 problem the geometrical constraint can be evaluated by a simple preliminary check. Therefore, they have been treated in a special way. Solutions were sampled for several times by the variation operators unless a feasible solution subject to the geometrical constraints was obtained or the maximal sampling number of 1000 was exceeded. In the latter case the violation of an implicit constraint is reported to the EA and this constraint function was treated in the standard way proposed for implicit constraints. The same procedure was implemented for both strategies, the metamodel-assisted NSGA-II and standard NSGA-II, in order to generate a higher ratio of feasible individuals.

The results of the study are summarized in figure 8.3.16 and figure 8.3.17. They demonstrate that the solution quality of the NSGA-II using metamodels is much higher than that for the standard NSGA-II. By using the metamodel-assisted EMOA succeeded to improve the diversity as well as the precision of the convergence to the Pareto front. Compared to the results achieved with the mean value filter the application of the lb_ω filter lead to a further improvement.

By means of metamodel-assistance it was possible to obtain a significantly higher number of feasible solutions in all five cases of the RAE 2822 problem (see figure 8.3.2). Furthermore, it was possible to find an improvement for the baseline design of the RAE 2822 test case. For $\mathbf{x}^* = (-0.000290, -0.000193, 0.000125, -0.000043, 0.000562, -0.000120)$ the vector of objective function values $f_1(\mathbf{x}^*) = 0.022266$, $f_2(\mathbf{x}^*) = 0.029198$, $f_3 = 0.011615$ clearly dominates that of the baseline design, that had been non-dominated by all solutions for this problem known so far.

8.4 Summary and conclusions

In all reported test cases the MAES outperformed the optimization tools that were formerly applied on these problems as well as standard implementations of the ES.

The results prove the applicability of the MAES even for difficult problem formulations with multiple constraints and more than two objectives.

Another important observation was, that for the application problems the acceleration of the MAES was particular high in the first iterations. In the long run, usually the standard ES versions reached similar quality values. The use of the confidence information in the lower confidence bound MAES usually made the MAES more robust and led to better final results. For the multi-objective problems, it also helped to get a high coverage of the Pareto front.

We note that customized versions of the MAES have now been integrated in commercial optimization packages like FORGE-3D$^{\text{TM}}$ and CASTS$^{\text{TM}}$ where it is used frequently and with high success for the solution of optimization tasks as an alternative to gradient-based methods.

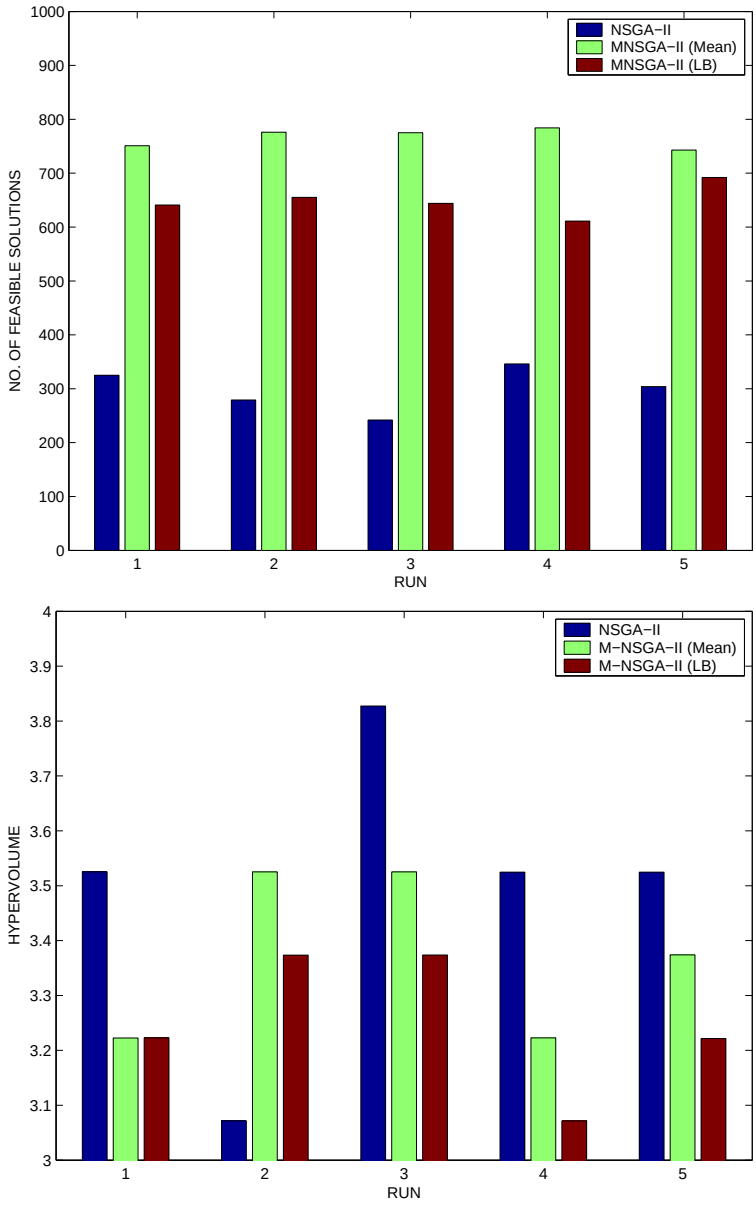


Figure 8.3.16: Results for airfoil shape optimization on the RAE test case. The upper figure displays the number of feasible solutions, i. e. solutions with no constrained violations, obtained with the metamodel-assisted NSGA-II ((20+20 < 100)-NSGA) with mean value and lower confidence bound filter, and standard (20 + 20)-NSGA-II. The lower figure depicts the hypervolume measure for the non-dominated region integrated over the interval-box $[0.022, 0.03], [0.028, 0.03], [0.0, 0.04]$ that contains all feasible solutions.

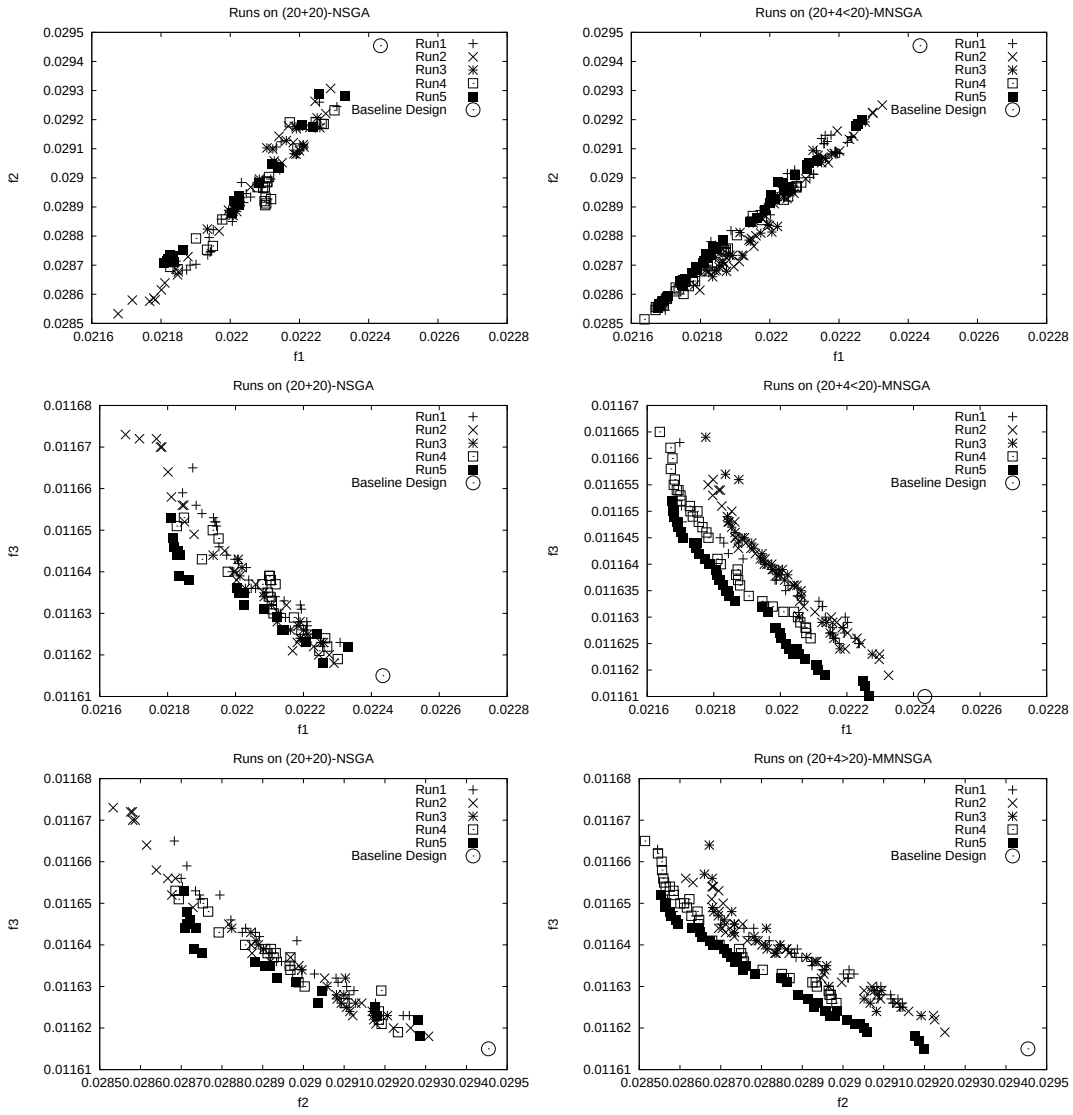


Figure 8.3.17: Scatter plot matrix for the three-objective RAE Airfoil Optimization. Each plot depicts projections from three-objective solutions space into a two-objective subspace. The data points in each particular plot denote pareto-optimal solutions obtained from 5 runs with the same strategy compared with the baseline design.

9 Summary and outlook

Our knowledge can only be finite, while our ignorance must necessarily be infinite.

Karl Popper

Gaussian random field metamodels (GRFM) were proposed for accelerating evolutionary optimization in the presence of time-consuming black box evaluations of deterministic computer models. We considered (constrained) single-objective optimization, as well as multi-objective optimization with conflicting objectives.

Chapter 2 dealt with a discussion of gaussian random field metamodels. Statistical assumptions and computational aspects were discussed in the context of modeling the output of deterministic functions. We pointed out that the number of training points mainly determines the computational effort for the calibration and prediction procedures. In order to speed-up the metamodel, so-called local metamodels were proposed, that work with a reduced training set existing of spatial neighbors of the input vector. Moreover, we conceptually compared metamodeling techniques. In particular, we established a close relationship between radial basis function networks and gaussian random field metamodels for the prediction of single and multiple outputs. It turned out, that gaussian random field metamodels extend the capabilities of (standard) radial basis function networks by providing calibration procedures for their correlation parameters, as well as by computing variances for the predictive distributions. Both features come at a computational cost, which cannot be neglected if the number of training points is high and precise function evaluations are comparably cheap.

Next, in chapter 3 an introduction of single-objective optimization was given. Firstly, we summarized some recent results on the black-box complexity of single-objective optimization. The results clarify, that both the dimensionality and the smoothness of the objective function determines the difficulty for optimization in the worst case. In the remainder of the chapter we reviewed some optimization techniques with a focus on methods that use metamodels or function approximations in order to accelerate search. The survey reveals that apart from recently proposed metamodel-assisted evolutionary algorithms there is a long history of deterministic optimization methods, such as bayesian optimization and model-assisted pattern search, that apply metamodels. Many concepts originally introduced for deterministic methods were later transferred into the context of evolutionary optimization. This holds also for some of the methods we developed within this thesis. In addition, chapter 3 introduced evolution strategies and pointed out their merits and limitations in the application domain of design optimization.

In chapter 4, we proposed metamodel-assisted evolution strategies (MAES) for single-objective optimization. As a key concept we introduced *filters* that are procedures that

draw a subset of ν promising individuals from the offspring population. Only this subset is considered for precise evaluation and, subsequently, in the replacement. All proposed filters consider predictions of the precise objective function value. Local gaussian random field metamodels, trained from all precise evaluations that are available, provide these predictions and also a confidence measure related to each predictions.

Section 4.3 provided first steps towards a general convergence theory of the MAES. Firstly, we proved for filters with $\nu > 0$ the global convergence of the corresponding MAES on regular functions with probability of one as the number of iterations approaches infinity. Concerning the convergence dynamics, which is the more important question in practice, we pointed out that amongst other difficulties the loss of the Markov property makes it difficult, if not impossible, to analyze the dynamical behavior of the MAES analytically. However, in section 4.3 by making an idealized assumption about the quality of the predictions, we derived some simple expressions for the speed-up of the MAES as compared to the standard ES.

The remainder of chapter 4 focussed mainly on the development of different kind of filters and the study of the algorithms that use these filters. Firstly, we proposed filters that select a subset of constant size ν from the offspring population. These filters pre-screen the population by means of scalar criteria based on the prediction and then select the ν best individuals. Besides the mean value filter that makes only use of the predicted value, with the lower confidence bound (lb_ω) filter, the expected improvement (ExI) filter, and the probability of improvement (PoI) filter we introduced also filters that take consider the confidence measure attributed to each prediction. In addition, filters with a variable output size are introduced. Whereas the R_ω-filter aims at a high recall, the P_ω-filter aims at a high precision of the selected subset with regard to the relevant solutions. Both filters rely upon the idea to use two-sided confidence bounds for the predictions, and thus are classified as interval filters. In addition, with the most likely improvement (MLI) and the lower confidence bound improvement (LBI) we identified straightforward generalizations of the mean value and lb_ω filter, that result in an output set of variable size. Note, that the application of the ExI and lb_ω filter were proposed by the author of this thesis [EGÖ+02], as well as the filters with variable output size, whereas the mean value filter (e.g. [GGP00], [Jin05]) and the probability of improvement filter [USZ03] were contributed by others. Finally, a brief discussion of stochastic selection methods in noisy optimization methods and related to the design of filters for the MAES.

The comparison of filters was carried out in three stages. Firstly, conceptual relationships between filters were deduced from their design. Secondly, empirical studies were conducted in order to gain insights into the behavior of the algorithms using the respective filters on different types of artificial test problems. As a final test of their applicability, some of the most promising algorithms were applied on real-world optimization problems.

The theoretical analysis revealed some interesting relationships between different types of filters. Firstly, we studied in more detail the influence of the variance of the predictive distribution on the filter. Whereas the lb_ω filter always rewards solutions with a high variance (describing the uncertainty of the prediction), for the PoI and ExI filter it depends on the difference between the predicted value and the so-far best found function value f^t_{best}, whether they reward a high variance or not. For filters with adaptive output size we pointed out invariant permeability relationships. These allows to establish general relationships between subsets selected by different filters without knowing the particular

input population (section 4.2). Moreover, we established a mapping between the LBI_ω and MLI filters and a threshold version of the PoI filter.

Various indicator measures were developed to observe the behavior of the MAES and to learn about its functioning. In order to measure the capability of a filter to select the relevant offspring individuals, we proposed the usage of recall and precision measures borrowed from the theory of information retrieval. Besides, the quality of the sorting achieved with the proposed pre-selection criteria has been measured. For this, an indicator based on the number of sorted pairs was proposed. Over and above, cross-validation and $y \sim y$ as well as $y \sim \mathrm{lb}_\omega$ diagrams were proposed to judge upon the numerical quality of uncertain predictions.

The proposed indicators alongside with common performance measures for the history of the best found function values were used for empirical studies on artificial test problems, including smooth quadratic problems, as well as discontinuous and multimodal problems. Some of the results from the empirical studies of the algorithm's performance are highlighted as follows:

- MAES with constant output size almost ever significantly outperformed counterparts of the standard ES on the test problems with 5 up to 20 dimensions.
- On local optimization problems the online training of the metamodel is sufficiently fast to allow for a high precision approximation of the optimum. This holds for all tested filters, including those using confidence measures.
- Filters with constant output size that made use of confidence measure, namely the mean value, lb_ω, PoI and ExI filter, were more reliable on multimodal functions.
- The behavior of filters with adaptive output size was rather instable. Filters with high precision tended to converge to local optima in multimodal optimization. Filters with high recall suffered from a very low local progress, which may be explained by the large number of individuals evaluated in each generation.

The analysis of the behavior of the filters and the metamodels during the run provided us with further valuable insights. By means of the precision and recall measures it has been verified that the MAES filters approximately behave how they are expected to behave due to their design. The plots of the number of inversion indicators suggest that the sorting of offspring populations was almost always significantly better than random sorting. As expected, the sorting based on the mean value criterion yielded the lowest number of inversions. An important observation was that the number of sorted pairs was higher for the mean value filter also on multimodal problems where it got stuck in local optima. This means, that despite the relatively high quality of the metamodel it has not been possible to find a better approximation to the global optimum. We concluded, that filters that stress on a good emulation of the EA are not necessarily filters that lead to the best performing algorithms on the more difficult test cases. Moreover, the good behavior of strategies using the confidence measure cannot be explained, by the assumption that sampling in unknown regions helps to increase the model quality. Rather, it is likely that the intelligent use of the confidence measure helps to lead the search into unexplored but promising regions of the search space.

In chapter 5 and 7 the filters developed for single-criterion optimization have been generalized for the more complex problem definitions of constrained and multi-objective optimization. For the constrained optimization straightforward generalizations have been suggested. First results, mainly for constant size output filters, were obtained on Keane's problem and on two application problems indicate the applicability of these extensions.

The generalization of the filter concepts to Pareto optimization with and without nonlinear constraints turned out to be considerably more difficult. Here, pre-selection criteria have to take account the improvement in the diversity of the non-dominated set of the current population and also in its convergence to the Pareto front. The hypervolume measure turned out to be a well-suited indicator that integrates both aspects in one scalar value. Since currently no EMOA used the hypervolume measure in its selection procedure, a new EMOA has been developed, called SMS-EMOA, that selects new individuals of a population by means of their contribution to the dominated hypervolume. This EMOA yielded superior results on standard test-suites (ZDT1 - ZDT4, ZDT6) for multi-objective optimization, even without assistance of a metamodel.

The idea to use a hypervolume metric in the context of multi-objective optimization is relatively new. So, efficient procedures had to be developed for computing hypervolume differences in the aforementioned algorithmic approach. Procedures were developed for the computation of the hypervolume differences in two- and three-dimensional solution spaces. For three dimensional sets the developed algorithm was significantly faster than state-of-the-art algorithms for this task.

After introducing this new EMOA in chapter 6 (and also the classical NSGA-II algorithm) the filters suggested for the single criterion case were generalized and integrated into the SMS-EMOA and NSGA-II (cf. chapter 7). The obtained metamodel-assisted EMOA have been tested both on simple artificial test functions and on challenging practical problems from airfoil design. It turned out that the criteria using uncertainty measures differ much more in the multi-objective problem domain than they do in the single-objective problem domain. In particular, it has been found that the lb_ω criterion leads to the best results with regards to the coverage of the Pareto front and its dominated hypervolume, while the generalized probability of improvement (PoI) criterion and the mean value criterion led to a good convergence, but tend to concentrate the population in a small region of the Pareto front.

In order to measure the properties of the proposed EMOA, we introduced two multi-objective function families which are scalable with respect to their dimension, namely the EBN family of functions and the generalized Schaffer problem. Both were analyzed in a rigorous manner, and it has been found that the curvature of their Pareto front can be gradually adjusted, in order to achieve concave, linear, and convex Pareto fronts.

Last but not least, applications in industrial design have been tackled. The survey comprised a wide range of subject areas including airfoil shape design, turbine blade casting, forging, and electromagnetic compatibility design. Single-objective, as well as multi-objective and constrained problems were tackled in this study. The results were very encouraging, and even complex nonlinear problems, dealing with multiple constraints and objectives were solved better than with state-of-the-art methods, like gradient based search methods, standard evolutionary algorithms, or pattern search methods.

Before looking at future extensions some *recommended parameter settings* for the MAES shall be provided for the practitioner: Based on the lessons learned in this thesis, we recommend the use of the MAES versions that utilize either the PoI filter or the lb_ω filter. The latter offers an extra parameter that allows us to scale between a fast local progress or a high robustness. A recommended setting is $\omega = 2.0$. Far more effective for increasing the local convergence speed, however, is a reduction of ν. Note, that for both measures that increase the local progress, we have to pay the price that a premature stagnation on multimodal landscapes gets more likely. For constrained optimization, we recommend the usage of the lower bound and PoI filter, too, though our evidence is still based on a very small test set, and further studies on artificial landscapes are encouraged for the future.

In multi-objective optimization there seems to be an advantage for the ExI filter and the lb_ω filter with $\omega \approx 2$. Since for the lb_ω filter computational procedures are more simple and do not require a problem specific reference point, this strategy could be a good starting point for practical applications. Furthermore, there are now already some studies that underpin the robustness of this criterion in practise. Whenever a small population is used and a distribution on the Pareto front is desired that emphasizes knee points, the use of the SMS-EMOA in favor of the NSGA-II as basic EMOA is strongly recommended.

As a parameter setting, for the problems studied in this thesis a $(5 + 20 < 100)$-MAES proved to be a good default value for the MAES parametrization. For multi-objective and constrained optimization μ was increased to 15 and 20, respectively, in order to achieve a higher diversity in the population. For optimization with a far smaller budget than 1000 precise evaluations smaller population sizes, as well as output sizes ν of the filters, should be considered. From experiences with practical optimization problems a ratio $\mu/\nu \approx 4$ proved to be a reasonable choice. The number of training points for the metamodel should be chosen proportionally to the dimensions of the search space. In this work we mainly worked with only $2d$ training points and already achieved quite good results. However, for practical applications a further increase of the number of training points is recommended.

9.1 Outlook

Though we belief that this work contributes to a deeper understanding and to an extension of the scope for metamodel-assisted design optimization, we also believe that there are still many open research questions. Obviously, the MAES approach would benefit from a further extension of experimental results on artificial and practical optimization problems going alongside with further studies of control parameters. An interesting approach for the tuning of algorithm parameters has been recently proposed by Bartz-Beielstein [BB05]. It would be interesting to apply this approach to the MAES for different optimization scenarios.

Another direction for future research is to apply GRFM as a data analysis tool for analyzing scattered data from optimization. A first step in this direction has been made in [Zho05]. Here metamodels were used to interpolate results from a database of (costly) objective function evaluations obtained during an optimization run. By means of the

proposed technique global and local properties of the objective function were estimated from the given data.

Furthermore, applications of the MAES in industrial design are envisaged. For example, the MAES is used in an ongoing project to optimize cooling designs for solidification processes, thereby using the GRFM as a difference model for modeling the deviation between a coarse grained and a fine grained model of the process. Moreover, the modeling of noisy objective function evaluations and the integration of techniques for dealing with noisy responses could extend the scope of the method significantly. For example physical experiments could be considered as evaluation functions, then. A starting point for could be recent work on noisy optimization with evolutionary algorithms and response surface models [BT05].

A particularly challenging endeavor would be to transfer the ideas of metamodel-assisted optimization for applications in discrete search spaces. GRFM with a single correlation parameter make no use of the vector field structure of $\mathbb{R}^d$ but only of the metric defined on $\mathbb{S}$. Provided an appropriate distance measure that yields a strongly correlated landscape is established on the search space, GRFM, or similar metamodels, could also be used for the prediction of function values in discrete search spaces. Techniques on how to define problem-specific distance functions for complex search spaces that lead to strongly correlated landscapes have for example been derived by Emmerich et al. [EGS01]. There, the idea was to use a distance measure based on the minimal sum of weighted minimal moves that are needed to transform one solution into another solution. As possible applications of discrete metamodel-assisted EA the accelerated simulator-based synthesis of truss constructions, the automatic parametrization of algorithms, the synthesis of energy production processes, or the de-novo design of proteins could be envisioned.

A Multi-objective test functions

A.1 ZDT1 problem

The ZDT1 problem is described as

$$f_1(\mathbf{x}) = x_1 \tag{A.1.1}$$

$$f_2(\mathbf{x}) = g(\mathbf{x})[1 - \sqrt{x_1/g(\mathbf{x})}] \tag{A.1.2}$$

$$g(\mathbf{x}) = 1 + \frac{9}{n-1}\sum_{i=d}^{n} x_i \tag{A.1.3}$$

$$\mathbf{x} \in [0,1]^d \tag{A.1.4}$$

$$x_i \in [0,1], i = 0, \ldots, d \tag{A.1.5}$$

The problem has a convex Pareto front. Its optima are given by $x_1 \in [0,1]$ and $x_i = 0, i = 2, \ldots, d$.

A.2 ZDT2 problem

The ZDT2 problem is described as

$$f_1(\mathbf{x}) = x_1 \tag{A.2.6}$$

$$f_2(\mathbf{x}) = g(\mathbf{x}) \cdot [1 - (x_1/g(\mathbf{x}))^2] \tag{A.2.7}$$

$$g(\mathbf{x}) = 1 + \frac{9}{n-1}\sum_{i=2}^{d} x_i \tag{A.2.8}$$

$$\mathbf{x} \in [0,1]^d \tag{A.2.9}$$

This problem has a concave Pareto front. Its optima are given by $x_1 \in [0,1]$ and $x_i = 0, i = 2, \ldots, d$.

A.3 ZDT3 problem

The ZDT3 problem is described as

$$f_1(\mathbf{x}) = x_1 \tag{A.3.10}$$

$$f_2(\mathbf{x}) = g(\mathbf{x}) \left(1 - \sqrt{x_1/g(\mathbf{x}))} - x_1/g(\mathbf{x}) \sin(10\pi x_1)\right) \tag{A.3.11}$$

$$g(\mathbf{x}) = 1 + \frac{9}{n-1} \sum_{i=2}^{n} x_i \tag{A.3.12}$$

$$\mathbf{x} \in [0,1]^d \tag{A.3.13}$$

This problem has a non-convex Pareto front. Its optima are given by $x_1 \in [0,1]$ and $x_i = 0, i = 2, \ldots, d$.

A.4 ZDT4 problem

The ZDT4 problem is described as

$$f_1(\mathbf{x}) = x_1 \tag{A.4.14}$$

$$f_2(\mathbf{x}) = g(\mathbf{x})(1 - (x_i/g(\mathbf{x}))^2) \tag{A.4.15}$$

$$g(\mathbf{x}) = 1 + 10(n-1) + \sum_{i=2}^{n} (x_i^2 - 10\cos(4\pi x_i)) \tag{A.4.16}$$

$$\mathbf{x} \in [0,1] \times [-5,5]^{d-1} \tag{A.4.17}$$

This problem has many local Pareto fronts. Its optima are given by $x_1 \in [0,1]$ and $x_i = 0, i = 2, \ldots, d$.

A.5 ZDT6 problem

The ZDT6 problem is described as

$$f_1(\mathbf{x}) = 1 - \exp^{-4x_1} \sin(6\pi x_1)^6 \tag{A.5.18}$$

$$f_2(\mathbf{x}) = g(\mathbf{x})(1 - (f_1(\mathbf{x})/g(\mathbf{x}))^2) \tag{A.5.19}$$

$$g(\mathbf{x}) = 1 + \frac{9}{n-1} \sum_{i=2}^{n} x_i \tag{A.5.20}$$

$$\mathbf{x} \in [0,1]^d \tag{A.5.21}$$

This problem is characterized by a low density of solutions near the Pareto front. Its optima are given by $x_1 \in [0,1]$ and $x_i = 0, i = 2, \ldots, d$.

A.6 Generalized Schaffer problem

The generalized Schaffer function is described as

$$f_1(\mathbf{x}) := \frac{1}{\gamma}(\sum_{i=1}^{d} x_i^2)^\gamma \rightarrow \min, \quad f_2(\mathbf{x}) := \frac{1}{\gamma}(\sum_{i=1}^{d}(1-x_i)^2)^\gamma \rightarrow \min \tag{A.6.22}$$

$$\mathbf{x} \in [0, 10]^d \tag{A.6.23}$$

The curvature of the Pareto front is scalable by means of the parameter γ. The equation describing the Pareto front reads

$$y_2 = (1 - y_1^{1/2\gamma})^{2\gamma}, y_1 \in [0, 1]. \tag{A.6.24}$$

Thus, $\gamma = 0.5$ results in a linear Pareto front, $\gamma < 0.5$ in concave Pareto fronts, and $\gamma > 0.5$ in convex Pareto fronts. The Pareto fronts are axis-symmetric to the bi-sector. The extremal points of this function are given by $(y_1, y_2)^T = (0, 1)^T$ and $(y_1, y_2)^T = (1, 0)^T$.

An analysis of this function, including the derivation of equation A.6.24, was provided by Emmerich [Emm05].

A.7 Analysis of the EBN family of functions

With expression 6.3.10 we introduced the EBN family of functions. For the reader's convenience, we will repeat the definition here:

$$f_1^\gamma(\mathbf{x}) = (\sum_{i=1}^{n} |x_i|)^\gamma \cdot n^{-\gamma} \rightarrow \min, \; f_2^\gamma(\mathbf{x}) := (\sum_{i=1}^{n} |x_i - 1|)^\gamma \cdot n^{-\gamma} \rightarrow \min$$

$$\mathbf{x} \in \mathbb{R}^d$$

The EBN function family, first proposed in [EBN05], is scalable in dimension and by choosing the parameter γ the curvature of its Pareto front can be controlled. In particular, values of $\gamma < 1$ result in a concave Pareto front, whereas values > 1 result in a concave Pareto front.

The equation describing the Pareto front reads

$$y_2 = (1 - y_1^{1/\gamma})^\gamma, y_1 \in [0, 1].$$

Next, we will provide a rigorous analysis of this function family, including a proof for A.7.25. In particular its pareto-optimal set will be derived, accompanied by expressions for the functions describing the Pareto front.

We start with the analysis of the instance with $\gamma = 1$, and later we generalize the results for arbitrary $\gamma > 0$.

Lemma 7. The Pareto-optimal set of the d-dimensional EBN function with $\gamma = 1$ is given by $[0, 1]^d \subset \mathbb{R}^d$

We will prove this lemma by first providing a number of propositions. From these propositions, the validity of lemma 7 can be easily concluded.

Let us prove that there is no non-dominated point outside of $[0,1]^d$:

Proposition 2. Every $\mathbf{x} \in \mathbb{R}^d - [0,1]^d$ is dominated by at least one point $\mathbf{x}' \in \mathbb{R}^d$.

To prove this, it suffices to prove the existence of a point $\mathbf{x}'$ with $\mathbf{x}' \prec \mathbf{x}$ for all $\mathbf{x} \in \mathbb{R}^d - [0,1]^d$, i. e.

$$\mathbf{x} \in \mathbb{R}^d - [0,1]^d \Rightarrow$$
$$\exists \mathbf{x}' \in \mathbb{R}^d : (f_1(\mathbf{x}') \leq f_1(\mathbf{x}) \text{ and } f_2(\mathbf{x}') < f_2(\mathbf{x})) \tag{A.7.25}$$
$$\text{or}$$
$$\exists \mathbf{x}' \in \mathbb{R}^d : (f_1(\mathbf{x}') < f_1(\mathbf{x}) \text{ and } f_2(\mathbf{x}') \leq f_2(\mathbf{x})). \tag{A.7.26}$$

Proof. In order to prove the disjunction stated above, we will first provide a set $\mathcal{L}_1 \subseteq \mathbb{R}^d$ for which the first part of the disjunction (expression A.7.25) always evaluates to true, and then provide a set $\mathcal{L}_2 \subseteq \mathbb{R}^d$ for which the second part of the disjunction (expression A.7.26) always evaluates to true. The union of these sets, $\mathcal{L}_1 \cup \mathcal{L}_2$, will then include all points in $\mathbb{R}^d$, except $[0,1]^d$.

First, we will determine a set $\mathcal{L}_1$ for which condition A.7.25 holds. It can be simply verified that, $\mathbf{x} \in \mathcal{L}_1$ implies A.7.25, if we define

$$\mathcal{L}_1 := \{\mathbf{x} \in \mathbb{R}^d | \exists t \in [0,1]^d : \underbrace{(f_1(\mathbf{x}') = f_1(\mathbf{x}) \text{ and } f_2(\mathbf{x}') < f_2(\mathbf{x}) \text{ and } x'_1 = \cdots = x'_d = t)}_{Cond1}\}. \tag{A.7.27}$$

holds,

Whenever condition *Cond1* is true this implies

$$f_1(\mathbf{x}) = f_1(\mathbf{x}') = \sum_{i=1}^{d} |x'_i| = \sum_{i=1}^{d} |t| = \sum_{i=1}^{d} t = d \cdot t. \tag{A.7.28}$$

Hence, $t = f_1(\mathbf{x})/d$. Moreover, *Cond1* implies

$$f_2(\mathbf{x}') = \sum_{i=1}^{d} (1 - x'_i) = d \cdot (1 - t) = d \cdot (1 - f_1(\mathbf{x})/d) = d - f_1(\mathbf{x}). \tag{A.7.29}$$

Hence, if $Cond1$ holds, also

$$d - f_1(\mathbf{x}) < f_2(\mathbf{x}) \tag{A.7.30}$$

has to be true. Regardless the choice of $\mathbf{x} \in \mathbb{R}^d$, the value of $f_2(\mathbf{x})$ is always greater than zero. Hence, the condition is fulfilled for all $f_1(\mathbf{x}) > d$. This condition definitely holds for all $\mathbf{x} \in \mathbb{R}^d - [-1,1]^d$. Thus condition A.7.25 holds, at least, for $\mathbf{x} \in \mathbb{R}^d - [-1,1]^d$.

In a similar manner, let us now determine a set $\mathcal{L}_2 \in \mathbf{x}$ for which $\mathbf{x} \in \mathcal{L}_2$ implies $A.7.25$. Such a set is given by

$$\mathcal{L}_2 = \{\mathbf{x} \in \mathbb{R}^d | \exists t \in [0,1]^d : \underbrace{(f_1(\mathbf{x}') < f_1(\mathbf{x}) \text{ and } f_2(\mathbf{x}') = f_2(\mathbf{x}) \text{ and } x'_1 = \cdots = x'_d = t)}_{Cond2}\}. \tag{A.7.31}$$

Since $|1-t| = 1-t$ for $t \in [0,1]$, for all t and $\mathbf{x}$ that make *Cond2* in expression A.7.31 evaluate true we are allowed to write

$$f_2(\mathbf{x}) = \sum_{i=1}^{d} |1-t| = \sum_{i=1}^{d} (1-t) \quad = d \cdot (1-t). \tag{A.7.32}$$

Hence, we get

$$t = 1 - f_2(\mathbf{x})/d \tag{A.7.33}$$

Furthermore, we have to make sure that $f_1(\mathbf{x}') < f_1(\mathbf{x})$ is fulfilled for the given choice of $\mathbf{x}'$. Making use of $|t| = t$ for $t \in [0,1]$, we obtain:

$$f_1(\mathbf{x}') = \sum_{i=1}^{d} x_i' = \sum_{i=1}^{d} t = d \cdot t < f_1(\mathbf{x}). \tag{A.7.34}$$

Now, by substituting t from A.7.33, the qualifying condition for $\mathbf{x}$ reads:

$$d - f_2(\mathbf{x}) < f_1(\mathbf{x}) \tag{A.7.35}$$

Clearly, $f_1(\mathbf{x})$ is greater than 0. Hence, the condition $f_1(\mathbf{x}') < f_1(\mathbf{x})$ is fulfilled for all $f_2(\mathbf{x}) > d$ and thus for all $\mathbf{x} \in \mathbb{R}^d - [0,2]^d$. Accordingly, all solutions in $\mathcal{L}_2 := \mathbb{R}^d - [0,2]^d$ are dominated by at least one $\mathbf{x}' \in \mathbb{R}^d$.

Summing up, each point in

$$\mathcal{L}_1 \cup \mathcal{L}_2 = (\mathbb{R}^d - [-1,1]^d) \cup (\mathbb{R}^d - [0,2]^d) = \mathbb{R} - [0,1]^d \tag{A.7.36}$$

is dominated by some point in $[0,1]^d$. □

Hence, only solutions within the interval $[0,1]^d$ remain as candidates for non-dominated solutions.

As an auxiliary result we prove

Proposition 3. Let $\mathbf{l} : [0,1] \mapsto \mathbb{R}^d$ be defined as $\mathbf{l}(t) := t \cdot (1, \ldots, 1)^T$. For every solution vector $(f_1(\mathbf{x}), f_2(\mathbf{x}))^T$ with $\mathbf{x} \in [0,1]^d$ there exists $t \in [0,1]$. such that $f_1(\mathbf{l}(t)) = f_1(\mathbf{x})$ and $f_2(\mathbf{l}(t)) = f_2(\mathbf{x})$.

Proof. By inserting the definitions for f_1 and f_2 we get the equation system:

$$\sum_{i=1}^{d} |x_i| \quad = \quad \sum_{i=1}^{d} |t| \tag{A.7.37}$$

$$\sum_{i=1}^{d} |1-x_i| \quad = \quad \sum_{i=1}^{d} |1-t| \tag{A.7.38}$$

Since all addends in the sums are positive or zero, we can further simplify to

$$\sum_{i=1}^{d} x_i \quad = \quad d \cdot t \tag{A.7.39}$$

$$\sum_{i=1}^{d} (1-x_i) \quad = \quad d - d \cdot t. \tag{A.7.40}$$

From the first expression, we obtain $t = \sum_{i=1}^{d} x_i/d$. It remains to be proven, that for this choice of t the second expression evaluates to true. This holds, because

$$d - d \cdot t = d - d \cdot \sum_{i=1}^{d} x_i/d = d - \sum_{i=1}^{d} x_i = \sum_{i=1}^{d} (1 - x_i). \tag{A.7.41}$$

□

Now, we can prove the following important result:

Proposition 4. All points $\mathbf{x} \in [0,1]^d$ are mutually non-dominated.

Proof. We prove this for $\mathcal{L} = \{\mathbf{x} | \mathbf{x} = \mathbf{l}(t), t \in [0,1]\}$. For all other points in the cube $[0,1]^d$ proposition 3 states that they are equivalent in the solutions space to some point in $\mathcal{L}$, and hence the proposition will also hold for them. Let $\mathbf{x} = (t, \ldots, t)^T$ and $\mathbf{x}' = (t', \ldots, t')^T$ denote two distinct points in $\mathcal{L}$. Then $f_1(\mathbf{x}) = d \cdot t$ and $f_2(\mathbf{x}) = d \cdot (1 - t)$. Now, given that $t \in [0,1[, t' \in]0,1]$: $f_1(\mathbf{x}) < f_1(\mathbf{x}') \Leftrightarrow |t| < |t'| \Leftrightarrow |1 - t| < |1 - t'| \Leftrightarrow f_2(\mathbf{x}) > f_2(\mathbf{x}')$. Thus, $\mathbf{x}$ cannot dominate $\mathbf{x}'$. □

Next we can prove that all points in $[0,1]^d$ are non-dominated.

Proposition 5. For all $\mathbf{x} \in [0,1]^d$ there exists no point in $\mathbf{x}' \in \mathbb{R}^d$ such that $\mathbf{x}' \prec \mathbf{x}$.

Proof. Points in the set $[0,1]^d$ are mutually non-dominated, so there can be no point in $\mathbf{x}' \in [0,1]^d$ that dominates $\mathbf{x}$. Points, $\mathbf{x}'' \in \mathbb{R}^d - [0,1]^d$ are dominated by some point $\mathbf{x}'$ in $[0,1]^d$. Due to the transitivity of the Pareto preference relation, $\mathbf{x}' \prec \mathbf{x}''$ would imply $\mathbf{x}' \prec \mathbf{x}$ which again contradicts with the mutual non-dominance of points in $[0,1]^d$. □

Using these propositions as 'building blocks', we can easily assemble a proof for the lemma 7. From proposition 2 we know that points outside of $[0,1]^d$ are not candidates for non-dominated points. Moreover, as all points in $[0,1]^d$ are mutually non-dominated (proposition 4) and none of them is dominated by a point outside $[0,1]^d$ (proposition 5), they must necessarily be non-dominated points.

We continue with the generalization of lemma 7 for arbitrary $\gamma > 0$:

Theorem 4. The non-dominated set of the EBN problem for $\gamma > 0$ is given by $[0,1]^d$. The Pareto front is described by $\{(y_1, y_2) | y_1 = t^\gamma \wedge y_2 = (1 - t)^\gamma \wedge t \in [0,1]\}$.

Proof. We recall the well known fact that applying monotonous transformations

- multiplication by means of a positive number and
- empowering of positive expressions by means of positive exponents

on both sides of equalities or inequalities are equivalence transformations. Hence the introduction of does neither change the Pareto precedence order between points in the search space, nor does it affect the Pareto optimal set. Hence, the Pareto optimal set

$[0,1]^d$ is inherited by all members of the function family with $\gamma > 0$. However, the parameter γ influences the shape of the Pareto fronts.

Since all solutions of the Pareto front are described by the co-domain of $\mathbf{l}(t)$, we obtain the expression for the Pareto front: $f_1(t) = (dt)^\gamma / d^\gamma = t^\gamma$ and $f_2(t) = (d - dt)^\gamma / d^\gamma = (1-t)^\gamma$. In order to make f_2 dependent of f_1 and get rid of the curve parameter t we transform $y_1 = t^\gamma$ to $t = y_1^{1/\gamma}$ and inserting the expression for t into $y_2 = (1-t)^\gamma$ we obtain

$$y_2(y_1, \gamma) = (1 - y_1^{1/\gamma})^\gamma. \tag{A.7.42}$$

□

For $\gamma = 1$ this evaluates to the linear function $y_2 = 1 - y_2$, for $\gamma = 2$ to

$$y_2(y_1, 2) = (1 - \sqrt{y_1})^2, \tag{A.7.43}$$

and for $\gamma = 0.5$ we obtain

$$y_2(y_1, 0.5) = \sqrt{1 - y_1^2}. \tag{A.7.44}$$

The analysis of the generalized Schaffer problem (appendix A.6) can be carried out in a similar manner. The interested reader is referred to [Emm05].

B Related publications and history

In this section I would like to relate work that has been published previously to work published in this thesis. Over and above, I want to acknowledge the contributions of my co-authors. At the same time, to make the discussion more transparent, I will provide some background information about the historical development of this thesis. Not all of the work published is summarized in my thesis and, vice versa, a great part of my thesis has not yet appeared in publications. Hence, this appendix might also be interesting for those, who are searching for additional material on metamodel-assisted optimization.

Center for applied systems analysis (1999-2002)

The first publications on optimization were related to my involvement in the research projects 'Gesamtoptimierung verfahrenstechnischer Anlagen mit naturanalogen Methoden' founded by the German Volkswagen Stiftung and 'Automatic Optimization of Selected Chemical Processes' founded by the German BMB+F, both situated at the CASA/ICD, Dortmund, with partners in chemical industry, as well as academic research institutes (ACCESS (RWTH Aachen), UMSICHT e.V. (Oberhausen) and Chair for Technical Thermodynamics (RWTH Aachen)). First publications dealt with automatized optimal chemical process synthesis (e. g. [Emm99, ESGG00, EGH$^+$00, EGG$^+$00, Emm00] and [EGS01]). Among these first publications, I would like to highlight the article:

> *Michael Emmerich, Monika Grötzner, Martin Schütz:* **Design of graph-based evolutionary algorithms: A case study for chemical process networks**, *Evolutionary Computation, 9, 3, 329-354, 2001*

In this article, guidelines for designing problem-specific evolutionary algorithms were proposed and applied for the optimization of an chemical engineering plant. In my thesis, this work has not been addressed in detail, though in chapter 9 the proposed method of how to extend the MAES to discrete search spaces is based on the construction method for metric-based evolutionary algorithms developed by the author in [EGS01].

Within the AUTO-OPTI-CHEM project I also got confronted with *continuous design optimization* problems in chemical industry. There we faced single- and multi-objective optimization problems with very time consuming evaluation functions (ranging from minutes up to several hours). Moreover, existing gradient-based optimization methods suffered from a lack of robustness, and other methods, like for example evolution strategies, were sought to solve these problems.

The studies reported in the following publications were triggered by our efforts for designing appropriate optimization algorithms for such problems. Three ways of how to accelerate the evolution strategy in the presence of time consuming function evaluations: (1) the acceleration of the step-size adaptation, (2) exploitation of parallel computing and (3) the use of approximate function evaluations. The latter idea culminated in the conception of the metamodel-assisted algorithms described in this thesis.

Michael Emmerich, Rafael Hosenberg: **TEA - A C++ library for the design of evolutionary algorithms**, *Technical Report of the Collaborative Research Center 531 Computational Intelligence, CI-106/01, University of Dortmund, January, 2001*

Part of the software for this thesis was developed using the TEA library, the conceptual design of which is described in this paper. The credits for the authorship of this publication go in equal parts to both co-authors.

Thomas Bäck, Michael Emmerich, Martin Schallmo: **Industrial applications of evolutionary algorithms: A comparison to traditional methods**, *I. C. Parmee, P. Hajela, Optimization in Industry, 303-314, Springer, London, 2002*

This was the first publication on the topic of evolutionary optimization with a small budget of function evaluations. It compares evolution strategies to other direct optimization methods on representative problems for design optimization (among others a three-dimensional thermal design problem). This work is referred to in chapter 3 of this thesis.

M. Schallmo contributed the discussion of the thermal design problem and parameterized the simulator. The direct search algorithms were selected and compared by me. A general overview on EA was contributed by Th. Bäck, who also initiated the work on this topic.

Michael Emmerich, Lars Willmes, Thomas Bäck: **Asynchronous evolution strategies for distributed direct optimisation**, *K. Giannakoglou, D.T. Tsahalis, J. Périaux, K.D. Papailiou, T. Fogarty, Evolutionary Methods for Design, Optimization, and Control with Applications to Industrial Problems, 53-58, CIMNE, Barcelona, 2002*

As a first attempt to increase the speed of evolutionary algorithms with a limited budget of function evaluations, we tried to exploit parallel computing resources. Various steady state approaches that minimize idle time were compared to synchronous parallelization schemes by means of order statistic and discrete-event simulations. The paper is referenced in chapter 6, where a steady state EA is developed.Credits for the manuscript and the work on the results go equally to the co-authors of this paper.

Alexios Giotis, Michael Emmerich, Boris Naujoks, Kyriakos Giannakoglou, Thomas Bäck: **Low-cost stochastic optimization for engineering applications**, *K. Giannakoglou, D.T. Tsahalis, J. Périaux, K.D. Papailiou, T. Fogarty, Evolutionary Methods for Design, Optimization, and Control with Applications to Industrial Problems, 361-366, CIMNE, Barcelona, 2002*

This paper features a preliminary version of the MAES (cf. chapter 4). K. Giannakoglou and A. Giotis were the first who proposed approximative function evaluations to accelerate single- and multi-objective evolutionary algorithms [GG99]. Also they were among the first, who promoted the use of these methods in turbomachinery [Gia99] and airfoil optimization [GG99, GGP00].

The collaboration with the LTT started 2001 within the framework of the IKY2000 bilateral exchange project between the CASA/ICD e.V. and the NTU Athens (Greece), financially supported by the German DAAD and the Greek IKY. The aim of this project, proposed by Th. Bäck and K. Giannakoglou and me was to extend the scope of applications to process engineering problems and to integrate approximations also into evolution strategies.

This first publication that emerged from our collaboration, already presents a first version of an evolution strategy with approximate function evaluations. At this time, metamodels were based on radial-basis function networks[1], developed by Giannakoglou and Giotis [GGP00]. In contrast to the MAES proposed in this thesis, the preliminary version MAES worked with a significantly smaller population size and did not make use of any error prediction.

Michael Emmerich, Alexios Giotis, Mutlu Özdemir, Thomas Bäck, Kyriakos Giannakoglou: **Metamodel-assisted evolution strategies**, *J. J. Merelo Guervós, P. Adamidis, H.-G. Beyer, J. L. Fernández-Villacañas, H.-P. Schwefel, Parallel Problem Solving from Nature - PPSN VII, Proc. Seventh Int'l Conf., Granada, 361-370, Springer, Berlin, 2002*

This paper proposed the MAES as it is described in this thesis. Both ideas, to switch from radial basis function networks to Kriging and to use the confidence information were contributed by me. Also, I suggested to switch from a $(1, 10)$-ES to a $(15, 100)$-ES and make a more intensive use of the metamodel for the evaluation of the offspring population.

Many detailed problems had to be solved to get a first stable version of the Kriging emulator running. These problems were solved to a great deal by M. Özdemir, who also worked on the visualization of results. K. Giannakoglou and A. Giotis provided a study on a test case. Also the general framework for metamodel-integration proposed by them was adopted. , Th. Bäck provided a study for the adaptation of step-sizes. The work on the manuscript was shared by all authors.

Thomas Bäck, Michael Emmerich: **Evolution strategies for optimisation in engineering applications**, *H.A. Mang, F.G. Rammerstorfer, J. Eberhardsteiner, Proc. Fifth World Congress on Computational Mechanics (WCCM V), Vienna, July 7-12, 2002, Int'l, Association for Computational Mechanics, 2002, http://wccm.tuwien.ac.at/, Paper-ID: 81284*

This paper provides an overview of evolutionary strategies for applications in computational mechanics that was mainly written by Thomas Bäck. It contains a section on

[1] A comparison of these approximation methods to the Kriging method, the use of which was later proposed by me, can be found in chapter 2 of this work.

metamodel-assisted evolution strategies and its application to the optimization of an airfoil. This part was contributed by me and it includes a comparison of the MAES to other state-of-the art optimization methods, like pattern search and sequential quadratic programming (cf. chapter 8).

APOMAT-COST Network (2003-2005)

The aim of the European APOMAT-COST (from: Automatic process optimization in the material sciences) initiative was to develop and to apply numerical optimization methodologies for automatic materials process design. The next three publications are related to my involvement within this European network and report on the application of metamodel-assisted evolution strategies within an industrial context. Part of this work is reproduced in chapter 8. Also the development of constraint handling methods for metamodel-assisted evolution strategies (chapter 5) was triggered by this project work.

Michael Emmerich, Jürgen Jakumeit: **Metamodel-assisted optimisation with constraints: A case study in material process design**, *G. Bugeda, J. A. Désidéri, J. Périaux, M. Schoenauer, G. Winter, Evolutionary Methods for Design, Optimization, and Control with Applications to Industrial Problems (CD-ROM), CIMNE, Barcelona, 2003*

This paper proposes the first constraint handling approach for metamodel-assisted evolution strategies. Also we applied the MAES for the first time in the domain of turbine blade casting.

This paper was written in equal parts by J. Jakumeit and me. I contributed the optimization algorithm, the constraint handling method, and the development of the software module `QUALITY-CASTS`™ for the numerical integration of local constraints functions. Besides, I suggested to learn metamodels from simulator outputs (from which objective functions can be derived) instead of modeling a penalized or aggregated function value. J. Jakumeit initiated the project and contributed the test problem, the objective function formulation, and the interpretation of results. He contributed also a Kriging monte carlo strategy and Nelder Mead's simplex strategy, both of which were also tested on the application problem. Part of the work was done during my short-term employment at the RWTH Aachen and I would like to thank the ministry of North-Rhine Westphalia (MSWWF) for financial support during that time.

Jürgen Jakumeit, Michael Emmerich: **Optimization of a gas turbine blade casting using evolution strategies and kriging**, *B. Filipic, J. Silc, Proc. Int'l Conf. Bioinspired Optimization Methods and Their Applications (BIOMA'04), 95-104, Jozef Stefan Institute, Ljubljana, Slovenia, 2004*

A follow-up of the aforementioned paper. This time, a more realistic industrial testcase was optimized and an enhanced version of the constraint and objective function formulation was used. The main innovation in this formulation was the weighting of

local constraint violations with respect to their control volume (see chapter 8, the idea of which was due to J. Jakumeit. I contributed the parts that were related to the MAES.

Jürgen Jakumeit, Michael Emmerich: **Inverse modeling and numerical optimization of heater temperatures in a Bridgman process** *MCWASP Conference Modeling of Casting, Welding and Advanced Solidification Processes XI 2005, France, (accepted for)*

This paper stands in line with the two previously mentioned publications. New test runs were presented. This time the optimization temperature profile was the focus and the similarity between simulated results and results from the actual production process were compared. Accordingly, for the first time a result obtained with the metamodel-assisted evolution strategy for the improvement of a production process in industry. My main contribution to this paper was the delivery of the problem specific optimization algorithm and its parallelization.

Chapter 5 and chapter 8 partly cover results from this paper.

Collaborative research center 'Computational Intelligence' (2003-2005)

In the year 2003 I started to work in the 'Collaborative Research Center Computational Intelligence' at the University of Dortmund at the Chair of Systems Analysis financed by the German DFG in a research project led by H.-P. Schwefel. There I worked on the operationalization of optimization methods within a multidisciplinary setting. Many of the application problems we were working on there were multi-objective problems, including problems with time consuming function evaluations. Most of the work on that topic was developed in cooperation with my co-workers N. Beume and B. Naujoks. During that time, I took the initiative to extend the MAES framework to multi-objective optimization. For that purpose I cooperated with my co-worker B. Naujoks, who to that time had already a profound expertise on the subject of multi-objective evolutionary optimization and multi-point airfoil design.

Michael Emmerich, Boris Naujoks: **Metamodel-assisted multiobjective optimisation strategies and their application in airfoil design**, *I. C. Parmee, Adaptive Computing in Design and Manufacture VI, 249-260, Springer, London, 2004*

The results of this chapter are partly published in chapter 5, 7 and 8. The idea and initiative to use the Kriging metamodels (including confidence information) also in multi-objective optimization with metamodel-assisted evolution strategies was due to me. The confidence interval based comparison methods in the pre-selection were also proposed by me. A discussion of multi-objective evolutionary algorithms (NSGA-II) and different ways of how to integrate metamodels were discussed with B. Naujoks. The evaluation of results on the airfoil test-case was done by B. Naujoks, who also contributed a method for averaging approximations of the Pareto fronts.

Michael Emmerich, Boris Naujoks: **Metamodel-assisted multi-objective optimisation with implicit constraints and their application in airfoil design**, *Proc. Int'l Conf. ERCOFTAC'04, Athens (CD-ROM), CIMNE, Barcelona, 2004*

In this paper, an extension of the metamodel-assisted NSGA-II is suggested that can deal also with (black-box) constraint functions. The results of this publications are partly reproduced in chapter 5,7 and 8.

My idea was to treat approximate constraint functions also by means of confidence interval boxes. The application problem and the interface to it was contributed by B. Naujoks. All other parts were developed in equal parts by the authors.

M. Emmerich, K. Giannakoglou, B. Naujoks: **Single- and multi-objective evolutionary optimisation assisted by gaussian random field metamodels.** *IEEE Transactions on Evolutionary Computation (TEC), 2005 (accepted for)*

Chapters 2 and chapters 4 - 8 include essentially all results of this journal article. Also the main theme of this paper, namely to emphasize on the use of confidence information and to generalize the metamodel-assisted ES from single to constrained and multi-objective optimization, corresponds to the main thread that runs through my thesis.

The paper was intended to provide both - an overview of our past-work and the presentation of new studies on the MAES. The overview on existing work was provided in equal parts by K. Giannakoglou and me. K. Giannakoglou proposed the initial idea of using approximate function evaluations in the pre-selection of single- and multi-objective optimization. K. Giannakoglou did pioneering on EA working with approximate evaluations, and he derived a basic architecture of the metamodel-assisted evolutionary algorithms (the idea of a pre-screening phase with approximate function evaluations).

I contributed the following results:

- The generalization of the filters to constrained and multi-objective optimization
- The study of the algorithms on test functions (excepting the study on the application problem)
- The idea to distinguish between precision and recall measures when assessing the quality of the subset selection by means of the metamodel and the formulation of these measures
- The test problem generator for multi-objective test problems (generalized Schaffer problem)
- The conceptual comparison of pre-screening criteria
- The derivation of multi-objective generalizations of the filters and their performance-assessment on the generalized Schaffer problem

Also, I conducted all test runs and implemented the optimization and metamodeling software.

B. Naujoks contributed a study on a very representative test-problem (the RAE2822 airfoil design) that exhausted all features of the developed metamodel-assisted evolutionary algorithm, namely the various types of constraint-handling and the handling of multiple objectives.

Michael Emmerich, Nicola Beume, Boris Naujoks: **An EMO algorithm using the hypervolume measure as selection criterion**, *C. A. Coello Coello, A. Hernández Aguirre, E. Zitzler, Proc. Evolutionary Multi-Criterion Optimization: Third Int'l Conference (EMO 2005), 3410, Lecture Notes in Computer Science, 62-76, Springer, Berlin, 2005*

This paper has been reproduced in large parts in chapter 7. Also one of the applications described in chapter 8 was first published in this paper.

The work on this paper started with my idea to develop an EMOA with a selection that is mainly based on the hypervolume measure, in order to enable a more elegant generalization of improvement-based pre-screening criteria.

The plan to develop a completely new EMO algorithm – which not necessarily works with metamodel-assistance – grew in discussions between all co-authors. The basic algorithm of the SMS-EMOA and efficient implementation was a joint work to which all co-authors contributed in equal parts, as well as the test of its performance on the ZDT and EBN functions.

Boris Naujoks, Nicola Beume, Michael Emmerich: **Multi-objective optimisation using S-metric selection: Application to three-dimensional solution spaces**, *G. W. Greenwood, Proc. 2005 Congress on Evolutionary Computation (CEC'05), Edinburgh, UK, IEEE Press, Piscataway NJ, 2005, (in print)*

This paper extends the SMS-EMOA to three-dimensional solution spaces. Part of the algorithms published in this work are described in a similar manner in chapter 6.

The test runs and analysis (both on the DTLZ benchmark problems and the application example) were carried out by N. Beume and B. Naujoks. Also they proposed specific adaptations of the SMS-EMOA for three dimensions. My contribution to this work was mainly a first version of the algorithm to compute the hypervolume measure in three dimensions (like it has been published in chapter 6). The version of this algorithm that has been published in this paper is an refinement of this initial algorithm credited to N. Beume and B. Naujoks. The credits for the development of the SMS-EMOA variants for three dimensional solution spaces and its performance analysis goes mainly to N. Beume and B. Naujoks.

Boris Naujoks, Nicola Beume, Michael Emmerich: **Metamodel-assisted SMS-EMOA applied to airfoil optimization tasks**, *accepted for EUROGEN 2005, Munich, International Conference on Design Optimization*

This paper describes the application of the metamodel-assisted SMS-EMOA to a problem in RAE airfoil optimization. Part of it is discussed in chapter 7 and 8. The integration of the metamodel-assistance into the SMS-EMOA was the main contribution by me. Application-problem specific adaptations of this approach as well as the design and analysis of experiments are credited to B. Naujoks and N. Beume.

Mihai-Christian Varcol Varcol and Michael Emmerich: **Metamodel-assisted Evolution Strategies applied in electromagnetics** *accepted for EUROGEN 2005, Munich, International Conference on Design Optimization*

This paper forms a part of the chapter on applications 8, and describes the application of the MAES in the domain of electromagnetic compatibility design.

The application problem was contributed by M.-C. Varcol, who also implemented the interface to the optimization algorithm, made problem specific adaptations of it, conducted test runs and interpreted the obtained results in the context of electromagnetics. My contribution was the disposal of the MAES. Also, I supervised M.-C. Varcol during the planning of the test runs and the knowledge integration phase. The work on the manuscript was shared by both authors in equal proportions.

Leiden Institute of Advanced Computer Science 2005

Michael Emmerich: **A rigorous analysis of two bi-criteria problem families with scalable curvature of the pareto fronts** *Leiden Institute on Advanced Computer Science, 2005, LIACS TR 2005-05*

The problem generators introduced in this paper were used for performance evaluation of multi-objective optimization algorithms in chapter 6 and 7.

Nomenclature

Abbreviations

ANN	Artificial neural network, page 22
BGO	Bayesian global optimization, page 37
CSA	Cumulative step-size adaptation algorithm, page 164
DES	Derandomized Evolution Strategy, page 164
DFPS	Davidon Fletcher Powell method , page 33
EA	Evolutionary algorithms, page 40
EBN	Multi-objective test problems by Emmerich, Beume, and Naujoks, page 133
EMC	Electromagnetic compatibility, page 155
EMOA	Evolutionary multi-objective optimization algorithm, page 120
ϵ-MOEA	Steady-state EMOA by Deb et al., page 134
ES	Evolution strategies, page 29
GPS	Generalized pattern search, page 35
GRF	Gaussian random fields, page 13
GRFM	Gaussian random field models, page 13
IPE filter	Imprecise evaluation filters, page 47
MA-DES	Metamodel-assisted DES, page 164
MAEA	Metamodel-assisted evolutionary algorithms , page 11
MAES	Metamodel-assisted evolution strategy, page 47
NACA	Test case for airfoil re-design, page 165
NSGA	Non-dominated sorting algorithm , page 118
PESA	Pareto evolution strategy, page 121

RAE2822	Test-problem defined by the Royal Airforce Establishment, page 168
RBF	Radial basis function, page 22
RBFN	Radial basis function networks, page 22
SGO	Statistical global optimization, page 37
SMS-EMOA	Evolutionary multi-objective optimization algorithm using $\mathcal{S}$ metric selection, page 118
SPEA	Strength Pareto evolutionary algorithm, page 121
SX	Single crystal, page 161
ZDT	Test problems by Zitzler, Deb and Thiele, page 134
DS	Directional solidified, page 161

Procedures and Operators

(μ, κ, λ)-ES	Multi-membered evolution strategy with population size μ, offspring population size λ and maximal life span κ, page 41
$(\mu, \kappa, \nu < \lambda)$-MAES	Metamodel assisted evolution strategy, with ν being the output size of its filter, page 48
(μ, λ)-ES	Multi-membered evolution strategy with population size μ, offspring population size λ and comma selection ($\kappa = 1$), page 41
(μ, λ)-ES	Multi-membered evolution strategy with population size μ, offspring population size λ and plus selection ($\kappa = \infty$), page 41
$rand()$	Random function generator, page 79
$\text{AA}_{\Delta\mathcal{S}}$	Archiving strategy by Knowles et al., page 124
ϵ-MOEA	Steady-state EMOA by Deb et al., page 134
evaluate	Evaluation procedure, page 41
ExI filter	Expected improvement filter, page 53
filter	Procedure that filters promising solutions, page 48
generate	Generation of offspring population via mutation and recombination, page 42
increase_age	Procedure that increments age of individuals, page 41
init	Initialization procedure, page 41
lb_ω filter	Filter using the lower confidence bound criterion with confidence factor $\omega = 2$, page 50

LBI_ω filter	Lower confidence bound improvement filter with confidence factor ω, page 55
MLI filter	Most likely improvement filter, page 55
mutate	Mutation procedure, page 41
P_ω-filter	High precision filter, page 58
PoI_τ filter	Filter accepting only individuals with probability of improvement exceeding a threshold τ, page 53
recombine	Recombination procedure, page 41
$\mathbf{reduce}_{\Delta\mathcal{S}}$	Elimination of worst element due to $\Delta\mathcal{S}$, page 125
replace	Replacement procedure, page 41
$\mathbf{replace}_{\Delta\mathcal{S}}$	replacement in SMS-EMOA, page 126
R_ω-filter	High recall filter, page 58
terminate	Procedure that checks termination criterion, page 41
$\hat{y}$ filter	Filter using the mean value criterion, page 50

Symbols

$(.)^T$	Transpose of a vector or matrix
$\mathbf{a}$	Individual ($\mathbf{a} \in \mathbb{I}$), page 41
$\beta, \beta_1, \ldots, \beta_{n_r}$	Regression function parameters for Kriging, page 17
$\mathbf{C}$	Correlation matrix, page 18
$\mathbf{X}$	Collection of evaluated search points, page 16
c	Correlation function, page 15
c'	Equivalent expression for $c(\mathbf{x})$, page 15
$\Delta\mathcal{S}(\mathbf{x}, R)$	Exclusive hypervolume of $\mathcal{S}(R)$ covered by $\mathbf{x}$, page 126
d	Input vector or search space dimension, page 10
$D^{\mathbf{r}}$	Partial derivative in the notation of Schwartz, page 25
D_t	Database of evaluated individuals, page 48
$\mathbf{d}_t$	Direction vector, page 32
E(.)	Mean value, page 15
ℓ	The index of the set containing elements with lowest rank of non-domination, page 121

erf	Gaussian error function, page 114
ExI	Expected improvement , page 39
$\mathcal{F}$	Gaussian Random Field, page 15
$\mathcal{F}_{\mathbf{x}}$	1-D random function of $\mathcal{F}$ at position $\mathbf{x}$, page 15
$f, f_1, \ldots, f_{n_f}$	Objective functions, page 10
f^t_{best}	Currently best individual, page 41
f_{sc}	Optimization criterion in bayesian global optimization, page 38
$g, g_1, \ldots, g_{n_g}$	Constraint functions, page 10
G_t	Sampled (generated) offspring population (not yet evaluated), page 41
$\mathbf{H}$	Matrix of activation function values, page 24
$\mathcal{H}_f$	Dominated hypervolume, page 112
h	Activation function for radial basis function networks, page 24
$\mathbb{I}$	Individual space, page 41
$\mathrm{Inv}_n(\pi)$	Number of inversions in permutation π, page 72
I(.)	Improvement function, page 39
$<_\kappa$	Comparison of individuals using maximum life span κ, page 45
κ	Maximal life span, page 41
k	Age of an individual (in the context of ES), page 41
λ	Number of offspring, page 41
$\Lambda(M)$	Lebesgue Measure of a set M, page 124
$\lambda_1, \ldots, \lambda_m$	Weights for linear predictor in GRFM, page 19
lb_ω	Lower confidence bound , page 38
LBI_ω	Lower bound improvement, page 55
μ	Number of parents, page 41
m	Number of sampled points, page 13
M_∞	Mach number, page 165
$M_\mu(A)$	The μ best solutions in A, page 71
MLI	Most likely improvement, page 54
∇f	Gradient of f, page 32

$\nabla^2 f$	Hessian matrix of f, page 32
ν	(Maximal) number of pre-selected individuals , page 48
n_f	Number of objective functions, page 10
n_g	Number of constraint functions, page 10
n_y	Response vector dimension, page 9
Ω	Borel algebra over $\mathbb{R}$
ω	Confidence factor , page 38
Ω_ω	Probability space, page 41
$\mathbf{1}$	Vector of ones, i. e. $(1, \ldots, 1)^T$, page 18
$\oslash$	Empty set, page 41
O_t	Evaluated offspring population, page 41
$\mathrm{nd}(R)$	Non-dominated subset of R, page 119
Ψ	Matrix of Radial basis function weights, page 24
ψ	RBF weight, page 24
$\Psi_\mu(M)$	Subset selection, page 59
$\wp(M)$	The set of all subsets (power set) of M ., page 63
p_α	Confidence level, page 57
P_t	Parent population, page 41
Φ	Cumulative gaussian distribution function
φ	Density of the gaussian distribution
PoI	Probability of improvement, page 52
Pr(A)	Probability of event A
Q_t	Offspring individuals that passed the filter, page 48
$\mathbb{R}$	Space of real numbers , page 9
$\mathcal{R}$	Mean zero gaussian random field, page 17
ρ	Number of individuals involved in recombination, page 41
$R_1, \ldots, R_\ell$	Sets of decreasing rank of non-domination, page 121
Re_c	Reynolds number (c=chord length), page 165
$\hat{s}$	Conditional standard deviation, page 16

$\hat{s}_{-i}$	Conditional standard deviation for cross-validation, page 27
$\mathbf{s}$	Vector of ES strategy parameters, page 41
$\mathbb{S}$	Search space, page 9
$\mathcal{S}(R)$	Hypervolume metric, page 124
σ	Step-size or (in evolution strategies) standard deviation of step-size , page 32
σ_{min}	Minimal step-size, page 67
s^2	Global variance of gaussian random field metamodel, page 15
τ_{global}	Global learning rate, page 44
τ_{local}	Local learning rate, page 44
$\theta, \theta_1, \ldots, \theta_d$	Parameters of correlation function, page 15
$\Upsilon_\mu(M)$	, page 58
Var(.)	Variance, page 15
Ξ_n	Standardized random variable ξ_n , page 72
ξ_n	Random variable describing the number of inversions in a random permutation of length n, page 72
$\mathbf{x}$	Object or input variables, page 41
$\mathbf{x}_{\max}$	Upper bounds of $\mathbb{S}$, page 10
$\mathbf{x}_{\min}$	Lower bounds of $\mathbb{S}$, page 10
$\mathbf{x}^t_{best}$	Currently best individual, page 41
$\mathbf{Y}$	Collection of result vectors, page 13
$\mathbb{Y}$	Response or solution space, page 9
$\mathbf{y}$	output vector: $\mathbf{y} = (f_1, \ldots, f_{n_f}, g_1, \ldots, g_{n_g})^T$, page 10
$\mathbf{y}^{\max}$	Reference point for the hypervolume, page 124
$\mathbf{y}_f$	Part of $\mathbf{y}$ that represents objective function values, page 108
$\mathbf{y}_g$	Part of $\mathbf{y}$ that represents constraint function values, page 108
$\hat{y}$	Estimated output value (conditional mean), page 16
$\hat{y}_{-i}$	Estimation for cross-validation, page 27
$y, y_1, \ldots, y_{n_y}$	Response or output values, page 10
$\zeta^A_\mu(M)$	Subset selection, page 58
$\zeta^B_\mu(M)$	Subset selection, page 59

Bibliography

[AB03] D. V. Arnold and H.-G. Beyer. A comparison of evolution strategies with other direct search methods in the presence of noise. *Computational Optimization and Applications*, 24(1):135–159, 2003.

[Ack87] D. H. Ackley. *A connectionist machine for genetic hillclimbing*. Kluwer Academic Publishers, Boston, MA, 1987.

[AD00] C. Audet and J. E. Dennis. Analysis of generalised pattern searches. Technical Report TR00-07, Rice university, Houston, 7 2000.

[Adl81] R. Adler. *The Geometry of Random Fields*. Wiley, NY, 1981.

[AP96] T.W. Athan and P.Y. Papalambros. A nore on weighted criteria methods for compromise solutions in multi-objective optimization. *Engineering Optimization*, 27:155–176, 1996.

[Bäc96] Th. Bäck. *Evolutionary Algorithms in Theory and Practice*. Oxford University Press, New York, 1996.

[BB05] Th. Bartz-Beielstein. *New Experimentalism Applied to Evolutionary Computation*. Natural Computing Series. Springer, Berlin, 2005. (in print).

[BBPM05] Th. B.-Beielstein, M. Preuß, and S. Markon. Validation and optimization of an elevator simulation model with modern search heuristics. In T. Ibaraki, K. Nonobe, and M. Yagiura, editors, *Metaheuristics: Progress as Real Problem Solvers*, chapter 5, pages 109–128. Kluwer, Boston MA, 2005. (im Druck).

[Bey01] H.-G. Beyer. *The Theory of Evolution Strategies*. Natural Computing Series. Springer, Berlin, 1 edition, 2001.

[BFM97] Th. Bäck, D. B. Fogel, and Z. Michalewicz, editors. *Handbook of Evolutionary Computation*. IoP Press, Bristol, UK, 1997.

[BL88] D. S. Broomhead and D. Lowe. Multivariate functional interpolation and adaptive networks. *Complex Systems*, 2:321–355, 1988.

[Bow76] V.J. Bowman. On the relation of the Chebycheff norm and the efficient frontier of multiple criteria objectives. *Lecture Notes in Economics and Mathematical Systems*, pages 76–85, 1976.

[BS02] H.-G. Beyer and H.-P. Schwefel. Evolution strategies - a comprehensive introduction. *Natural Computing*, 1(1):3–52, 2002.

[BSK05] D. Büche, N. N. Schraudolph, and P. Koumoutsakos. Accelerating evolutionary algorithms with gaussian process fitness function models. *IEEE Transactions on Systems, Man and Cybernetics, Special Issue on Knowledge Extraction and Incorporation in Evolutionary Computation (Part C)*, 35(2):183–194, 2005.

[BT05] P. Buchholz and A. Thümmler. Enhancing evolutionary algorithms with statistical selection procedures for simulation optimization. In *Proc. ACM Winter Simulation Conference (WSC)*, Orlando, Florida, Dec. 2005. in print.

[Car94] E. F. Carter. The generation and application of random numbers. *Forth Dimensions*, 14:12–24, August 1994.

[CC77] A. Charnes and W. Cooper. Global programming using multiple objective optimization - part I. *European Journal of Operations Research*, 1:39–54, 1977.

[CDG99] D. Corne, M. Dorigo, and F. Glover, editors. *New Ideas in Optimization*. David Hatter, University Press, Cambridge, 1999.

[CGTT00] A. R. Conn, N. Gould, P. L. Toint, and L. Toint. *Trust-Region Methods*. Series on Optimization. MPS-SIAM, 2000.

[CJ97] D. D. Cox and S. John. SDO: a statistical method for global optimization. In V. Hampton, editor, *Multidisciplinary design optimization*, volume 2, pages 315–329. SIAM, Philadelphia, PA, 1997.

[Coe99] C. A. Coello Coello. A survey of constraint handling techniques used with evolutionary algorithms. Technical Report 99, Laboratorio Nacional de Informática Avanzada, Xalapa, Veracruz, México, 4 1999.

[CVL02] C. A. Coello Coello, D. A. Van Veldhuizen, and G. B. Lamont. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Kluwer Academic Publishers, New York, May 2002. ISBN 0-3064-6762-3.

[CW90] D. Coppersmith and S. Winograd. Matrix multiplication via arithmetic programming. *Journal on Symbolical Computation*, pages 251–280, 9 1990.

[DAPM00a] K. Deb, S. Agrawal, A. Pratab, and T. Meyarivan. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In M. Schoenauer et al., editor, *Parallel Problem Solving from Nature - PPSN VI, Paris*, LNCS, pages 849–858, Berlin, 2000. Springer.

[DAPM00b] K. Deb, S. Agrawal, A. Pratab, and T. Meyarivan. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. KanGAL report 200001, Indian Institute of Technology, Kanpur, India, 2000.

[Dav81] H. A. David. *Order statistics*. Wiley, 2), address = NY edition, 1981.

[Deb01] K. Deb. *Multi-objective optimization using evolutionary algorithms*. John Wiley & Sons, Chichester, UK, 2001.

[DF04] T. T. Do and L. Fourment. Forging process optimization. Technical report, CEMEF, Ecole de Mines de Paris, Sophia Antipolis, 2004. APOMAT Cost 526 - Half yearly report, 8. 2004.

[DFL04] T. T. Do, L. Fourment, and M. Laroussi. Sensitivity analysis and optimization algorithms for 3d forging process design. In *NUMIFORM'04, Columbus, USA*, 2004. in press.

[Din98] M. Dinolfo. Matrix Inversion in C++, 1998. Internet Web-Site (checked 2005): users.erols.com/mdinolfo/matrix.htm.

[DMC99] M. Dorigo, V. Maniezzo, and A. Colorni. The ant system: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics*, 26(1):29–41, 1999.

[DMM03a] K. Deb, M. Mohan, and S. Mishra. A Fast Multi-objective Evolutionary Algorithm for Finding Well-Spread Pareto-Optimal Solutions. KanGAL report 2003002, Indian Institute of Technology, Kanpur, India, 2003.

[DMM03b] K. Deb, M. Mohan, and S. Mishra. A fast multiobjective evolutionary algorithm for finding well-spread pareto-optimal solutions. Technical Report 2003002, KanGAL, Kanpur, India, 2003.

[Dob37] T. Dobzhansky. *Genetics and the Origin of Species*. Columbia University Press, 1937.

[DPAM02] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA–II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, April 2002.

[DS78] L. C. W Dixon and G. P. Szegö. The global optimization problem: An introduction. In L. C. W Dixon and G. P. Szegö, editors, *Towards Global Optimization*. North-Holland Publishing Company, 1978.

[DV97] J. E. Dennis and V.Torczon. Managing approximation models in optimisation. In N. M. Alexandrov and N. Y. Hussaini, editors, *Multidisciplinary Design Optimisation: State-of-the-art*, pages 330–347. SIAM, Philadelphia, 1997.

[EBN05] M. Emmerich, N. Beume, and B. Naujoks. An emo algorithm using the hypervolume measure as selection criterion. In C. Coello Coello et al., editor, *EMO 2005 Int'l Conference, March 2005, Guanajuato Mexico*, 2005. accepted for.

[EBNK99] M. A El-Beltagy, P.B. Nair, and A.J. Keane. Metamodelling Techniques for Evolutionary Optimisation of Computationally Expensive Problems: Promises and Limitations. In W. Banzhaf, J. Daida, A.E. Eiben, M.H. Garzon, V. Honavar, M. Jakiela, and R.E. Smith, editors, *Proc. of GECCO, Int'l Conf. on Genetic and Evolutionary Computation, Orlando 1999*, pages 196–203. Morgan Kaufman, 1999.

[EGG+00] M. Emmerich, M. Grötzner, B. Groß, F. Henrich, P. Roosen, and M. Schütz. Strukturoptimierung verfahrenstechnischer Anlagen mit evolutionären Algorithmen. In S. Hafner, H. Kiendl, R. Kruse, and H.-P. Schwefel, editors, *Computational Intelligence im industriellen Einsatz: Fuzzy Systeme, Neuronale Netze, Evolutionäre Algorithmen, Data Mining*, pages 277–282, Düsseldorf, 2000. VDI-Verlag.

[EGH+00] M. Emmerich, B. Groß, F. Henrich, P. Roosen, and M. Schütz. Global optimization of chemical engineering plants by means of evolutionary algorithms. In *Proc. Aspen World 2000: Optimizing the Manufacturing Enterprise*, Cambridge MA, 2000. Aspen Technology. *(CD-ROM).*

[EGN05] M. Emmerich, K. Giannakoglou, and B. Naujoks. Single- and multi-objective evolutionary optimisation assisted by gaussian random field metamodels. *IEEE-Transactions on Evolutionary Computation (in print)*, 2005.

[EGÖ+02] M. Emmerich, A. Giotis, M. Özdemir, Th. Bäck, and K. Giannakoglou. Metamodel-assisted evolution strategies. In J. J. Merelo Guervós et al., editor, *Parallel Problem Solving from Nature - PPSN VII, Granada, Spain*, LNCS, pages 361–370, Berlin, 2002. Springer.

[EGS01] M. Emmerich, M. Grötzner, and M. Schütz. Design of graph-based evolutionary algorithms: A case study for chemical process networks. *Evolutionary Computation*, 9(3), 2001.

[EH01] M. Emmerich and R. Hosenberg. Tea - a C++ library for the design of evolutionary algorithms. Technical report, SFB 531, Universität Dortmund, 2001. Reihe CI 106/04.

[EJ03] M. Emmerich and J. Jakumeit. Metamodel-assisted optimisation with constraints: A case study in material process design. In G. Bugeda et al., editor, *Evolutionary Methods for Design, Optimisation and Control with Applications to Industrial and Societal Problems - EUROGEN'2003*. CIMNE, Barcelona, 2003. CDROM.

[EJ04] M. Emmerich and J. Jakumeit. Optimization of a gas turbine blade casting using evolution strategies and kriging. In B. Filipic et al., editor, *Bioinspired Optimization methods and their applications - BIOMA'2004*, pages 95–104, Ljubljana, Sl, 2004. Jozef Stefan Institute.

[Emm99] M. Emmerich. Optimierung verfahrenstechnischer Prozeßstrukturen mit evolutionären Algorithmen. Technical report, Dept. of Computer Science, University of Dortmund, Februar 1999.

[Emm00] M. Emmerich. An interval constraint propagation technique for chemical process network synthesis. In M. Sasikumar, D. Rao, and P. R. Prakash, editors, *Proc. Knowledge Based Computer Systems (KBCS 2000), Mumbai*, pages 470–481, New Delhi, 2000. Allied Publishers.

[Emm05] M. Emmerich. A rigorous analysis of two bi-criteria problem families with scalable curvature of the pareto fronts. Technical Report LIACS TR 2005-05, Leiden Institute on Advanced Computer Science, Leiden, NL, May 2005.

[EN04a] M. Emmerich and B. Naujoks. Metamodel-assisted multiobjective optimisation strategies and their application in airfoil design. In I. Parmee, editor, *Proc of. Fifth Int'l. Conf. on Adaptive Design and Manufacture (ACDM), Bristol, UK, April 2004*, pages 249–260, Berlin, 2004. Springer.

[EN04b] M. Emmerich and B. Naujoks. Metamodel-assisted multiobjective optimisation strategies with implicit constraints and their application in airfoil design. In K. Giannakoglou et al., editor, *ERCOFTAC 2004*, page CDROM, Barcelona, 2004. CIMNE.

[ESB02] M. Emmerich, M. Schallmo, and Th. Bäck. Industrial applications of evolutionary algorithms: A comparison to traditional methods. In I. Parmee and P. Hajela, editors, *Optimisation in Industry, Proc. of Int'l Conf., Barga, Italy 2001*, pages 303–314, Berlin, 2002. Springer.

[ESGG00] M. Emmerich, M. Schütz, B. Groß, and M. Grötzner. Mixed-integer evolution strategy for chemical plant optimisation with simulators. In I. Parmee, editor, *Adaptive Design and Manufacture*, pages 55–67, Berlin, 2000. Springer.

[Fle03] M. Fleischer. The measure of pareto optima. applications to multi-objective metaheuristics. In Carlos M. Fonseca et al., editor, *Evolutionary Multi-Criterion Optimization - EMO'2003, Faro, Portugal*, 2632, pages 519–533, Berlin, 2003. Springer.

[FM90] A.V. Fiacco and G.P. McCormick. *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*. SIAM, Philadelphia, 1990.

[Fon95] C. M. M. Fonseca. *Multiobjective Genetic Algorithms with Applications to Control Engineering Problems*. PhD thesis, Department of Automatic Control and Systems Engineering, University of Sheffield, Sheffield, UK, September 1995.

[FP63] R. Fletcher and M. J. D. Powell. A rapidly convergent descent method for minimization. *Computer Journal*, 6, 1963.

[FSATV92] W. H. Flannery, S. A. S. A. Teukolsky, and W. T. Vetterling. *Interpolation and extrapolation*, chapter 3, pages 99–122. Cambridge University Press, 1992.

[GEN+01] A. Giotis, M. Emmerich, B. Naujoks, K. Giannakoglou, and Th. Bäck. Low cost stochastic optimisation for engineering applications. In *Proc. Int'l Conf. Industrial Applications of Evolutionary Algorithms, EUROGEN2001, Athens, Greece*, Barcelona, 2001. CIMNE (CD-ROM).

[GG99] A. P. Giotis and K.C. Giannakoglou. Single- and multi-objective airfoil design using genetic algorithms and artificial intelligence. In *EUROGEN 99, Evolutionary Algorithms in Engineering and Computer Science*, pages 65–72, 1999.

[GGK01] K. Giannakoglou, A. Giotis, and M. Karakasis. Low-cost genetic optimization based on inexact pre-evaluations and the sensitivity analysis of design parameters. *Inverse Problems in Engineering*, 9:389–412, 2001.

[GGP00] A. P. Giotis, K. C. Giannakoglou, and J. Périaux. A Reduced-Cost Multi-Objective Optimization Method Based On The Pareto Front Technique, Neural Networks And PVM. In *Proc. European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS'00),* (CD-ROM), Barcelona, 2000. Center for Numerical Methods in Engineering (CIMNE).

[Gia99] K.C. Giannakoglou. Designing turbomachinery blades using evolutionary methods. In *ASME Paper 99-GT-181, 44th ASME Gas Turbine and Aero-engine Congress*, Indianapolis, IN, USA, 1999.

[Gia02] K. C. Giannakoglou. Design of optimal aerodynamic shapes using stochastic optimization methods and computational intelligence. *International Review Journal Progress in Aerospace Sciences*, 38:43–76, 2002.

[GMW81] P.E. Gill, W. Murray, and M.H. Wright. *Practical Optimization.* Academic Press London, New York, 1981.

[Gol89] D. E. Goldberg. *Genetic Algorithms in Search, Optimisation and Machine Learning.* Addison Wesley, Reading, MA, 1989.

[HJ61] R. Hooke and T. A. Jeeves. Direct search solution of numerical and statistical problems. *JACM*, 8:212–229, 1961.

[HO01] N. Hansen and A. Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9(2):159–195, 2001.

[HP95] R. Horst and P.M. Pardalos, editors. *Handbook of Global Optimization.* Kluwer, 1995.

[HS96] F. Hoffmeister and J. Sprave. Problem independent handling of constraints by use of metric penalty functions. In L. J. Fogel, P. J. Angeline, and Th. Bäck, editors, *Evolutionary Programming V - Proc. Fifth Annual Conf. Evolutionary Programming (EP'96)*, pages 289–294. The MIT Press, 1996.

[JB05] Y. Jin and J. Branke. Evolutionary optimization in uncertain environments - a survey. *IEEE Transactions on Evolutionary Computation*, 9(3), 2005. in print.

[JGV03] C.-P. Jeannerod, P. Giorgi, and G. Villard. On the complexity of polynomial matrix computations. In *Proc. of the 2003 international symposium on Symbolic and algebraic computation (ISSAC), Philadelphia, PA*, pages 135–142. ACM Press, NY, August 2003.

[JHN05] J. Jakumeit, M. Herdy, and M. Nitsche. Parameter optimization of the sheet metal forming process using an iterative parallel kriging algorithm. *Structural and Multidisciplinary Optimization*, Januar 2005. in print.

[Jin05] Y. Jin. A comprehensive survey of fitness approximation in evolutionary computation. *Soft Computing Journal*, 9(1):3–12, 2005.

[Joh81] F. John. *Partial Differential Equations.* F. John, 4 edition, 1981.

[JSW98] D. Jones, M. Schonlau, and W. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4):433–492, 1998.

[KC02] J. Knowles and D. Corne. On metrics for comparing nondominated sets. In *Congress on Evolutionary Computation (CEC'2002), Honolulu, Hawaii*, pages 711–716, Piscataway, NY, 2002. IEEE Press.

[KC03] J. Knowles and D. Corne. Properties of an Adaptive Archiving Algorithm for Storing Nondominated Vectors. *IEEE Transactions on Evolutionary Computation*, 7(2):100–116, April 2003.

[KCF03] J. D. Knowles, D. W. Corne, and M. Fleischer. Bounded archiving using the lebesgue measure. In *Congress on Evolutionary Computation (CEC'2003), Canberra, Australia*, pages 2490–2497, Piscataway, NY, 2003. IEEE Press.

[KES01] J. Kennedy, R. Eberhart, and Y. Shi. *Swarm intelligence*. Morgan Kaufmann Publishers, 2001.

[KL85] L. W. Koenig and A. M. Law. A procedure for selecting a subset of size m containing the ℓ best of k independent normal populations, with applications to simulation. *Commun. Statist.–Simulation and Computation*, 14:719–734, 1985.

[KO96] J. R. Koehler and A. B. Owen. Computer experiments. In S. Ghost and C. R. Rao, editors, *Handbook on Statistics*, volume 13, pages 239–245. Elsevier-Science, 1996.

[Kra03] O. Kramer. Restriktionsbehandlung bei Evolutionsstrategien mit Geschlechtern. Technical Report SYS-1/03, Dept. of Computer Science, University of Dortmund, August 2003.

[Kur99] F. Kursawe. *Grundlegende empirische Untersuchungen der Parameter von Evolutionsstrategien - Metastrategien*. Dissertation, Dept. of Computer Science, University of Dortmund, 1999.

[Kus62] H. J. Kushner. A versatile stochastic model of a function of unknown and time vrying form. *Journal of Math. Anal. Appl.*, pages 150–167, 5 1962.

[LT99] R. M. Lewis and V. Torczon. Pattern search algorithm for bound constrained minimisation. *SIAM Journal on Optimisation*, 9(4):1082 – 1099, 1999.

[Mac98] D. J. C. MacKay. Introduction to gaussian processes. In C. M. Bishop, editor, *Neural Networks and Machine Learning*, volume 168 of *NATO Advanced Study Institute*, pages 133–165. Springer, Berlin, 1998.

[Mar01] B. H. Margolius. Permutations with inversions. *Journal of Integer Sequences*, 4 2001. Article 01.1.4 (Electronic Journal).

[Mie99] K. Miettinen. *Nonlinear Multiobjective Optimization*. Kluwer Academic Publishers, NL, 1999.

[MTZ78] J. Mockus, V. Tiesis, and A. Zilinskas. The application of Bayesian methods for seeking the extremum. In L. C. W. Dixon and G. P. Szego, editors, *Towards Global Optimization*, volume 2, pages 117–129. North-Holland, 1978.

[Mye92] D. E. Myers. Kriging, cokriging, radial basis functions and the role of positive definiteness. *Computers Mathematics Applications*, 24(12):139–148, 1992.

[Nas98] S. G. Nash. Sumt (revisited). *Operations Research*, 46:763–775, 1998.

[NBE] B. Naujoks, N. Beume, and M. Emmerich. Multi-objective optimisation using s-metric selection:
application to three-dimensional solution spaces. In *CEC'2005, Edinburgh, UK*. submitted for.

[NBE05] Boris Naujoks, Nicola Beume, and Michael Emmerich. Metamodel-assisted SMS-EMOA applied to airfoil optimization tasks. In *Proceedings EUROGEN'05* (CD-ROM). 2005. (im Druck).

[ND03] P. K. Nain and K. Deb. Computationally effective search and optimization procedure using coarse to fine approximations. In *Congress on Evolutionary Computation (CEC'2003), Canberra, Australia*, pages 2081–2088, Piscataway, NY, 2003.

[Nov99] E. Novak. Numerische Verfahren für Hochdimensionale Probleme und der Fluch der Dimension. In *Jahresbericht der DMV 101*. DMV, 1999.

[NR96] E. Novak and K. Ritter. Global optimization using hyperbolic cross points. In C.A. Floudas and P.M. Pardalos, editors, *State of the Art in Global Optimization*, pages 19–33. Kluwer, Boston, 1996.

[NWTW02] B. Naujoks, L. Willmes, T.Bäck, and W.Haase. Evaluating multi-criteria evolutionary algorithms for airfoil optimisation. In J. J. Merelo Guervós et al., editor, *Parallel Problem Solving from Nature - PPSN VII, Granada, Spain*, LNCS, pages 841–850, Berlin, 2002. Springer.

[OBS98] A. I. Oyman, H.-G. Beyer, and H.-P. Schwefel. Where elitists start limping: Evolution strategies at ridge functions. In A. E. Eiben, Th. Bäck, M. Schoenauer, and H.-P. Schwefel, editors, *Proc. Parallel Problem Solving from Nature – PPSN V, Amsterdam*, pages 34–43, Berlin, 1998. Springer.

[OGH94] A. Ostermeier, A. Gawelczyk, and N. Hansen. Step-size adaptation based on non-local use of selection information. In Y. Davidor et al., editor, *Parallel Problem Solving from Nature - PPSN III, Jerusalem*, volume 866 of *LNCS*, pages 189–198, Berlin, 1994. Springer.

[Pad00] A. Padula. Interpolation and pseudorandom function generators. Senior honors thesis, Department of Mathematics, College of William and Mary, Williamsburg, VA, 2000.

[Pag02] M. D. Pagel. *Encyclopedia of Evolution.* Oxford University Press, 2002).

[Par95] J. Paredis. Coevolutionary computation. *Artificial Life*, 2(4):355–375, 1995.

[PSE05] M. Preuss, L. Schönemann, and M. Emmerich. Counteracting genetic drift and disruptive recombination in (μ, λ)-EA on multimodal fitness landscapes. In *Proc. Genetic and Evolutionary Computation Conf. (GECCO 2005), Washington D.C.*, 2005. (im Druck).

[Ran03] R. Ranz. On the complexity of matrix product. *SIAM Journal on Computing*, 32(5):1356–1369, 2003.

[Rat98] A. Ratle. Accelerating the convergence of evolutionary algorithms by fitness landscape approximations. In A.E. Eiben et al., editor, *Parallel Problem Solving from Nature - PPSN V, Amsterdam, NL*, LNCS, pages 87–96, Berlin, 1998. Springer.

[Rec94] I. Rechenberg. *Evolutionsstrategie '94*. Frommann-Holzboog, Stuttgart, 1994.

[Rin01] H. Rinne. *Taschenbuch der Statistik*. Verlag Harri Deutsch, Heidelberg, 2001.

[Rit90] K. Ritter. Approximation and Optimisation on the Wiener Space. *Journal of Complexity*, 6:337–364, 1990.

[Rud01] G. Rudolph. A partial order approach to noisy fitness functions. In J.-H. Kim, B.-T. Zhang, G. Fogel, and I. Kuscu, editors, *Proc. 2001 Congress on Evolutionary Computation (CEC'01), Seoul*, pages 318–325, Piscataway NJ, 2001. IEEE Press.

[Sac97] V. N. Sachkov. *Probabilistic Methods in Combinatorical Analysis*. Cambridge University Press, 1997.

[Sch95] H.-P. Schwefel. *Evolution and Optimum Seeking*. Wiley, N.Y., 1995.

[Sch02] H.-P. Schwefel. Deep insight from simple models of evolution. *BioSystems*, 64(1–3):189–198, 2002.

[SEP04] L. Schönemann, M. Emmerich, and M. Preuß. On the extinction of evolutionary algorithm subpopulations on multimodal landscapes. *Informatica (Sl)*, 28(4):345–351, 2004.

[Sie00] C. Siefert. *Model-assisted Pattern Search*. Senior honor thesis, College of William and Mary, Williamsburg, VA, 2000.

[Str69] V. Strassen. Gaussian elimination is not optimal. *Numerical Mathematics*, 13(4):354–356, 1969.

[SWJ98] M. Schonlau, W. Welch, and D. Jones. Global versus local search in constrained optimization of computer models. In W. F. Rosenberger N. Flournoy and W.K. Wong, editors, *New Developments and Applications in Experimental Design*, volume 34, pages 11–25. Institute of Mathematical Statistics, Hayward, California, 1998.

[SWMW00] J. Sacks, W. J. Welch, W. J. Mitchell, and H.-P. Wynn. Design and analysis of computer experiments. *Statistical Science*, 4(4):409–435, 2000.

[SWN03] T. J. Santner, N. J. Williams, and W. I. Notz. *The Design and Analysis of Computer Experiments*. Springer, Berlin, 2003.

[SX93] M. Schoenauer and S. Xanthakis. Constrained GA optimization. In Stephanie Forrest, editor, *Proc. of the fifth international conference on genetic algorithms*, pages 573–580, San Mateo, CA, 1993. Morgan Kaufmann.

[Tor97] V. Torczon. On the convergence of pattern search algorithms. *SIAM Journal of Optimisation*, 7:1–25, 1997.

[TT97] M.W. Trosset and V. Torczon. Numerical optimization using computer experiments. Technical Report TR 9738, Institute for Computer Applications in Science and Engineering ICASE, NASA Langley Research Center, Hampton Virgina, 1997.

[Tve03] A. Tveit. On the complexity of matrix inversion. Technical report, Department of Computer and Information Science, Norwegian University of Science and Technology (IDI-NTNU)), Trondheim, Norway, 2003.

[TZ89] A. Törn and A. Zilinskas. *Global Optimization*, volume 350 of *Lecture Notes in Computer Science*. Springer, Berlin, 1989.

[USZ03] H. Ulmer, F. Streichert, and A. Zell. Evolution strategies assisted by gaussian processes with improved pre-selection criterion. In *Congress on Evolutionary Computation (CEC'2003), Canberra, Australia*, pages 692–699, Piscataway, NY, 2003. IEEE-Press.

[Var03] C.-M. Varcol. Einsatz von metamodell-gestützten Evolutionsstrategien in der elektromagnetischen Feldoptimierung. Technical report, Dept. of Computer Science, University of Dortmund, Dortmund, 3 2003. Diploma Thesis.

[vR79] C.J. van Rijsbergen. *Information retrieval*. Butterworths, London, 2 edition, 1979.

[VT00] M. W. Trosset V. Torczon. Direct search methods: Then and now. Technical Report NASA/CR-2000-210125, ICASE Report No. 2000-26, ICASE, Hampton, VA, 2000.

[Wei00] S. Weinzierl. Introduction to monte carlo methods. Technical Report NIKHEF-00-012, NIKHEF, Theory Group, Amsterdam, 2000.

[Wei04] E. Weisstein. Strassens formulas. From MathWorld–A Wolfram Web Resource., 2004. http://mathworld.wolfram.com/StrassenFormulas.html.

[Wie01] D. Wiesmann. *Anwendungsorientierter Entwurf evolutionärer Algorithmen*. Shaker Verlag, Aachen, 2001. Dissertation at Dept. of Computer Science, University of Dortmund.

[ZDT00] E. Zitzler, K. Deb, and L. Thiele. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evolutionary Computation*, 8(2):173–195, Summer 2000.

[Zho05] J. Zhou. Entwicklung einer Java/c++ Komponente zur Datenanalyse und -visualisierung von Daten aus der Optimierung. Technical report, Dept. of Computer Science, University of Dortmund, Dortmund, September 2005.

[Zit99] E. Zitzler. *Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications*. PhD thesis, Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, November 1999.

[ZLT01] E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the performance of the strength pareto evolutionary algorithm. Technical Report 103, Computer Engineering and Communication Networks Lab (TIK), Swiss Federal Institute of Technology (ETH), Zurich, 2001.

[ZM00] D. B. Fogel Z. Michalewicz. *How to Solve It: Modern Heuristics*. Springer, Berlin, 2000.

[ZT98] E. Zitzler and L. Thiele. Multiobjective optimization using evolutionary algorithms— a comparative study. In A. E. Eiben et al., editor, *Parallel Problem Solving from Nature - PPSN V, Amsterdam, NL*, LNCS, pages 292–301, Berlin, 1998. Springer.

[Zup04] C. Zuppa. Error estimates for modified local shepard's interpolation formula. *Applied Numerical Mathematics archive*, 49(2):245–259, 2004.