MSPOT: Multi-Criteria Sequential Parameter Optimization

Martin Zaefferer, Thomas Bartz-Beielstein,
Martina Friese, Boris Naujoks, and Oliver Flasch

Schriftenreihe CIplus.
TR 2/2012. ISSN 2194-2870
Cologne University of Applied Sciences

January 30, 2012

Abstract

Many relevant industrial optimization tasks feature more than just one quality criterion. State-of-the-art multi-criteria optimization algorithms require a relatively large number of function evaluations (usually more than 10^5) to approximate Pareto fronts. Due to high cost or time consumption this large amount of function evaluations is not always available. Therefore, it is obvious to combine techniques such as Sequential Parameter Optimization (SPO), which need a very small number of function evaluations only, with techniques from evolutionary multi-criteria optimization (EMO). In this paper, we show how EMO techniques can be efficiently integrated into the framework of the SPO Toolbox (SPOT). We discuss advantages of this approach in comparison to state-of-the-art optimizers. Moreover, with the resulting capability to allow competing objectives, the opportunity arises to not only aim for the best, but also for the most robust solution. Herein we present an approach to optimize not only the quality of the solution, but also its robustness, taking these two goals as objectives for multi-criteria optimization into account.

1 Introduction

In many fields of industrial optimization, the duration of a process feedback plays a dominant role in the optimization process. Large evaluation times, caused by slow real-world processes or large computational processing times, restrict optimization processes to only a very limited number of such evaluations. Moreover, almost every industrial optimization task features more than one quality criterion. Techniques from multi-criteria decision making, evolutionary multi-criteria optimization in particular, were developed during the last decade to solve such tasks. The necessity to combine EMO techniques and optimization methods such as SPO [2], which require a very small number of function evaluations only, should be self-evident.

A typical real-world application is the optimization of structural parameters and process parameters of industrial process plants. In a recent research project, preparations are made to apply Multi-criteria SPOT (MSPOT) to the optimization of dust separators for coal power plants. Such cyclones are large steel structures used to filter dust from the exhaust gas flow of coal power plants. An optimal cyclone combines the conflicting goals of high filtration efficiency with low pressure loss. The behavior of cyclones can be simulated with a high degree of accuracy by using computational intensive numerical methods, such as multiphase computational fluid dynamics. Even if executed on a compute cluster, a single computational fluid dynamics run will require from several minutes to hours of processing time, severely limiting the number of possible evaluations.

Another well-known application of SPOT is the tuning of algorithm parameters. Here, not only one, but two or more objectives can be of interest as well. This study therefore presents Multi-Criteria Optimization (MCO) with SPOT. The three main topics of this paper are:

1. Outlining an approach to apply SPOT to MCO problems.
2. Comparing MSPOT to state-of-the-art algorithms in the field. Test situations should reflect aspects of industrial tuning problems such as costly function evaluations.
3. Testing how well the robustness of tuning algorithm parameters with SPOT can be increased by considering quality and standard deviation as objectives.

Research in combining MCO and surrogate model optimization is not a new topic. Various research in this topic has been performed (cf. [13]). In particular, Voutchkov and Keane [16] introduced a multi-criteria approach for sequentially improving on surrogate models, and tested it on simple multi-criteria functions with very few function evaluations. Their approach is similar to MSPOT and uses a subset of the test functions considered in our study, however, restricted to low dimensional decision spaces. Also, SPOT has previously been applied to multi objective optimization, but not as an MCO algorithm itself, but as a tuner for such an algorithm. TODO CITE?

The research goals are presented in more detail in Sec. 2. Section 3 gives an introduction to the topic of MCO. The following Sec. 4 describes how MCO can be integrated into SPOT. The settings for an experimental study on this approach are described in Sec. 5. Results of these experiments are reported and analyzed in Sec. 6 and Sec. 7. The paper concludes with a summary and an outlook.

2 Research Goals and Questions

The goals of this experimental study can be formulated as two research questions.

Research Question 1 *Is there a competitive advantage of MSPOT over state-of-the-art MCO algorithms?*

It is of interest if MSPOT can compete with or even outperform state-of-the-art MCO algorithms. Outperform means to reach higher hypervolumes on multi-criteria test problems. As stated above, target functions for real-world optimization problems can be very costly to evaluate. Therefore, the test problems are restricted to few function evaluations, i.e., a few ten or few hundred function evaluations are feasible. To show development beyond these limits we chose to test up to a maximum budget of 1000 function evaluations. The state-of-the-art MCO algorithms in this comparison are chosen to be NSGA2 and SMS-EMOA. The experimental analysis related to Research Question 1 will be referred to as Case Study I in the following.

Research Question 2 *Is* MSPOT*'s multi-criteria model optimization approach advantageous for robust parameter optimization?*

Robustness can be defined by using the following goals (we consider minimization problems in this study): (i) to minimize the mean $\overline{Y}$ (or expectation) of the objective function values, and (ii) to minimize the standard deviation $\mathrm{sd}(Y)$ of these objective function values.

If optimization of $\overline{Y}$ is chosen as the only objective, an undesirable solution might be considered to be optimal due to noise. On the other hand, if minimization of $\mathrm{sd}(Y)$ is chosen as the only objective, a solution might be found, which may not be optimal. It is insufficient to optimize the expectation or to minimize the standard deviation only in the search for robust optimal solutions. The trade-off between optimality and standard deviation plays a central role in nearly every optimization problem.

This paper considers the search for robust solutions as a multi-criteria optimization problem. To this end, a measure for robustness based on $\mathrm{sd}(Y)$ of the objective function in the presence of noise has been introduced. Jin and Sendhoff [14] use a variance-based robustness measure to deal with the search for robust solutions as a multi-criteria problem. The main advantage of multi-criteria approaches to the search for robust optimal solutions over the existing ones is that the user is able to make a choice among a set of solutions and select those that can deal best with the problem at hand. The experimental analysis related to Research Question 2 will be referred to as Case Study II in the following.

3 Multi-Criteria Optimization

The field of MCO can be separated in different parts. One separation is based on the fact whether preferences of a decision maker are incorporated in the optimization process before or after the application of optimization algorithms (a-priori vs. a-posteriori approaches). The most influential methods in the last decade for the a-posteriori approach (decision making after application of algorithms), which are also incorporated in this paper, are EMO Algorithms (EMOA). [8, 7] The space, which is covered by a front of non-dominated solutions (Pareto front), turned out to be a standard quality indicator for EMOA results. Moreover, it was incorporated for selection within EMOA in recent years. One of such implementations is SMS-EMOA [4], which, next to other implementations, is on the way to replace the former standard NSGA2 [8]. Therefore, the MSPOT approach, which is developed in this paper, is compared to both of these algorithms, i.e., NSGA2 as well as SMS-EMOA.

4 Multi-Criteria Optimization with SPOT

To understand how MSPOT extends the single-criteria SPOT approach, the latter will be introduced first. As a first step, SPOT generates an initial design of several points and evaluates it on the optimization problem function. Based on these evaluation results, SPOT builds a surrogate model (e.g., linear, Kriging, or tree-based models). Two approaches can be used to exploit that model. (i) The naive approach samples a large number of points in the decision space. Those points are then evaluated on the surrogate models and the best will be suggested for evaluation on the problem function. (ii) More sophisticated approaches use well-known optimization techniques to find the optimum of the surrogate model.
Related design points are evaluated on the costly problem function. This process of building and exploiting the surrogate model is repeated sequentially until a termination criterion is fulfilled. Algorithm 1 presents a formal description of the MSPOT scheme. This scheme discriminates the two phases of SPOT, namely the generation of an initial design (lines 1–5) and its sequential model based improvement (lines 6–14).

In Phase 1, SPOT determines a design of initial points in the decision space and evaluates each point k_0 times on the problem function. Here, in contrast to the single-criteria SPOT, evaluation yields not a scalar result for each design point, but a vector. Each element is one of n objective values to be optimized. Phase 2 consists of the following loop:

1. Build a model M for each of the n objectives from the evaluated design points.
2. Generate a (large) set of l design points $\vec{X}'$ by Latin hypercube sampling and compute their utilities $\vec{Y}'$ by evaluating them on the model(s).
3. Select $a = |\vec{X}''|$ (which has to be smaller than l) seemingly Pareto optimal design points from the large design. This cannot be achieved by sorting the results $\vec{Y}'$, because two or more objective values are known for each sampled design point. Instead, the design points are first sorted by their non-dominated sorting rank, i.e., all values on the current Pareto front get the lowest rank, then are removed. The Pareto front of the remaining points gets the next higher rank. This is repeated until all points in the objective space are ranked. This will assign the same value to all points on an individual Pareto front. If the settings of MSPOT require less

Algorithm 1: MSPOT

```
// phase 1, building the first design:
let O be the optimization problem with $d$ decision variables and $n$ objectives;
generate an initial design $\vec{X} := \{\vec{x}^1, \ldots, \vec{x}^m\}$ with $\vec{x} \in \mathbb{R}^d$ of $m$ design points;
fix $k := k_0$ to be the number of tests for determining estimated utilities;
foreach $\vec{x} \in \vec{X}$ do
    evaluate O with $\vec{x}$ $k$ times to determine the estimated utilities $\vec{y}$ of $\vec{x}$, where $\vec{y} \in \mathbb{R}^n$;
// phase 2, build, use and improve models:
while termination criterion not true do
    build n surrogate models $M_i, i \in \{1, \ldots, n\}$, based on $\vec{X}$ and $\vec{Y} := \{\vec{y}_1, \ldots, \vec{y}_m\}$;
    generate a set $\vec{X}'$ of $l$ new design points by latin hypercube sampling;
    foreach $\vec{x}' \in \vec{X}'$ do
        determine the predicted utility for each model with $\vec{y}' := M(\vec{x}')$;
    select best set $\vec{X}''$ of $a$ design points from $\vec{X}'$ based on non-dominated sorting rank and hypervolume contribution ($a \ll l$);
    foreach $\vec{x}'' \in \vec{X}''$ do
        evaluate O with $\vec{x}''$ $k$ times to determine the estimated utility $\vec{y}''$ of $\vec{x}''$;
    extend the design by $\vec{X} := \vec{X} \cup \vec{X}''$ and $\vec{Y} := \vec{Y} \cup \vec{Y}''$;
with $\vec{Y}$ create Pareto front $\vec{P}$ which contains all non-dominated points;
```

values than are available for the current non-dominated sorting rank, a tie breaker has to be introduced. This is solved by calculating the hypervolume contribution of each point, and removing those points with the smallest contribution. The contribution is recalculated after each removal, since hypervolume contribution might change for different subsets of the Pareto front.

4. The selected points $\vec{X}''$ are added to the design and evaluated on the problem. The loop starts over if the termination criterion is not reached.

Once the termination criterion is reached, the final Pareto front can be computed from the archive of design values. That is, all non-dominated points are reported to the user. This whole scheme represents the naive sampling approach. In case the naive approach is not to be used, well known MCO techniques such as NSGA2 or SMS-EMOA can be applied. These search the surrogate models M for Pareto optimal points. To avoid further sorting problems the population size of these MCO methods is set to the desired number of design points, say a, to be used in the next evaluations on the optimization target function. A budget for the model optimization has to be specified by the user.

More complex features of SPOT could be used, like increasing the repeats k on each design point to improve estimation on noisy problems. Consequently, this means that the best design points so far are also run again to obtain a comparable number of repeats. For simplicity the MCO approach only uses a

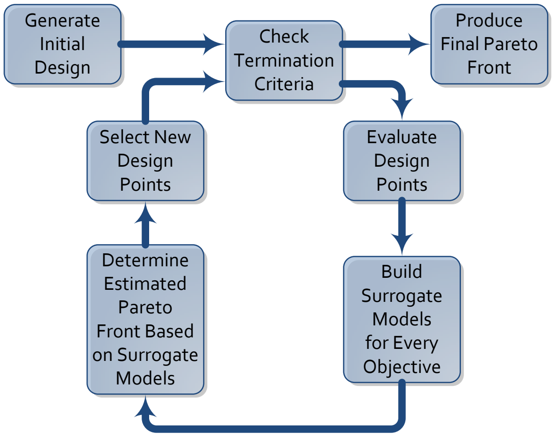
Generate Initial Design
Check Termination Criteria
Produce Final Pareto Front
Select New Design Points
Evaluate Design Points
Determine Estimated Pareto Front Based on Surrogate Models
Build Surrogate Models for Every Objective

Figure 1: The general MSPOT loop

fixed number of k_0 repeats. The general MSPOT loop is illustrated in Fig. 1.

5 Experimental Settings

The multi-criteria version of SPOT, as established above, is tested in two different scenarios. In the first case, typical MCO test functions are optimized by MSPOT, and the results are compared to established MCO algorithms. This is done to evaluate the quality of the approach in general, under the assumption of a minimal available budget. The second case is a new application of MCO. Here, the MSPOT approach is used for robust tuning of algorithm parameters by considering both expectation and standard deviation as quality criteria.

5.1 Case Study I. Comparison

To compare MSPOT with state-of-the-art MCO algorithms, a comparison with NSGA2 and SMS-EMOA was performed on MCO test functions, namely ZDT1 to ZDT3, DTLZ1, and DTLZ2. [17, 9] Test functions and their relevant settings are summarized in Table 1. As mentioned earlier, Voutchkov and Keane [16] proposed an approach, which is somehow related to MSPOT. Amongst other they also tested on ZDT2 , although with a lower decision-space dimension. Figure 2 exemplifies that MSPOT can approximate Pareto fronts with very little evaluations on their use case. The experiments described here however, consider a higher dimensional decision space to see if MSPOT can also improve on significantly harder problems with less evaluations. Though the used functions are not explicitly hard problems, increasing the dimensionality raises the challenge for the employed algorithms. With the limited budget, it is unlikely for any method to approximate the real Pareto front adequately.

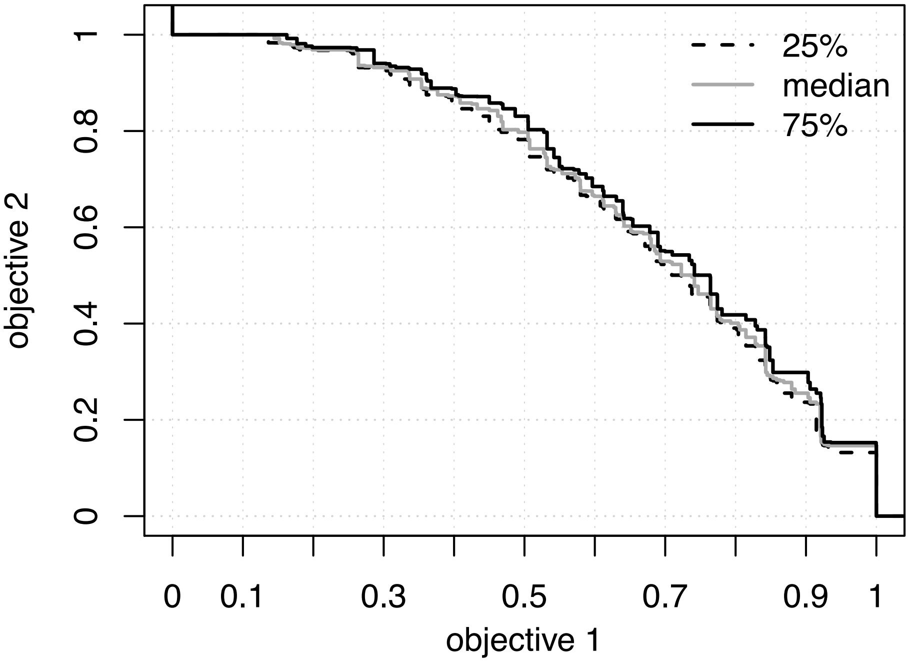

Figure 2: Attainment surfaces plotted with "eaf" package in R. Based on 10 runs with MSPOT on two dimensional ZDT2, with exactly 40 function evaluations.

The settings used for the MSPOT runs are listed in Table 2. The number of repeats k is fixed to a value of one, because the test functions are not noisy.

Three surrogate models are used by MSPOT for this experiment, i.e.,

1. spotPredictForrester (sFo): Kriging model for R based on the Matlab code by Forrester et al. [10]. The model uses CMAES [12] to find its parameters.
2. spotPredictRandomForest (sRF): Random Forest model from the R package[1] "randomForest" which is based on Breiman and Cutler's original Fortran code for classification and regression [5].
3. spotPredictEarth (sEA): Multivariate adaptive regression splines (MARS) [11] provided by the "earth" R-package.

All models are used with default settings. Consequently, the MARS model only considers linear terms without interactions. It has to be noted that SPOT keeps an archive of non-dominated solutions. It often yields more points than NSGA2 and SMS-EMOA since these are limited by their population size. Therefore, the Pareto front based on this archive has to be reduced. Reducing the number of points in the front avoids bias towards the MSPOT approach.

[1] SPOT and all other used R packages can be retrieved from the CRAN homepage, i.e. `http://cran.r-project.org`

Table 1: Multi-criteria test functions in use. Dimensions d and n name the dimensions of the decision space and the objective space. The reference point is used for calculating the hypervolume indicator at the end of each optimization run. The decision space is bounded by 0 and 1.

Function	Dimension (d,n)	Reference Point
ZDT1	(30,2)	(11,11)
ZDT2	(30,2)	(11,11)
ZDT3	(30,2)	(11,11)
DTLZ1	(7,3)	(1000,500,500)
DTLZ2	(12,3)	(11,11,11)

Table 2: Additional setup of MSPOT

Parameter	setting in MSPOT
init.design.size m	100
init.design.repeats k_0	1
seq.design.maxRepeats k	1
seq.design.new.size a	32
seq.predictionModel.func M	”spotPredictForrester” ”spotPredictRandomForest” ”spotPredictEarth”
seq.predictionOpt.budget	1000
seq.predictionOpt.method	”nsga2”

For SMS-EMOA, an implementation by Mersmann[2] was used. For NSGA2, the implementation from the ”mco” package was used in our study. Population sizes of NSGA2 and SMS-EMOA were chosen to be in line with the design size a. Besides that, all settings of SMS-EMOA and NSGA2 were left at default values.

In many real-world problems, the number of available function evaluations is very restricted. This is in contrast to typical MCO test functions, which are usually optimized invoking several thousand of function evaluations. To reflect real-world settings, the number of function evaluations will be set to 100, 500, and 1000 in our experiments. It is the scope of this paper to identify EMOA that cope with very small numbers of fitness function evaluations available in the industrial applications of interest. Since low budgets would limit the number of generations, the population size (”seq.design.new.size” for SPOT) is changed along with the budget (size 4 with budget 100, size 16 with budget 500 and size 32 with budget 1000).

[2] Available at: `https://git.p-value.net/emoa.git/plain /examples/sms_emoa.r`

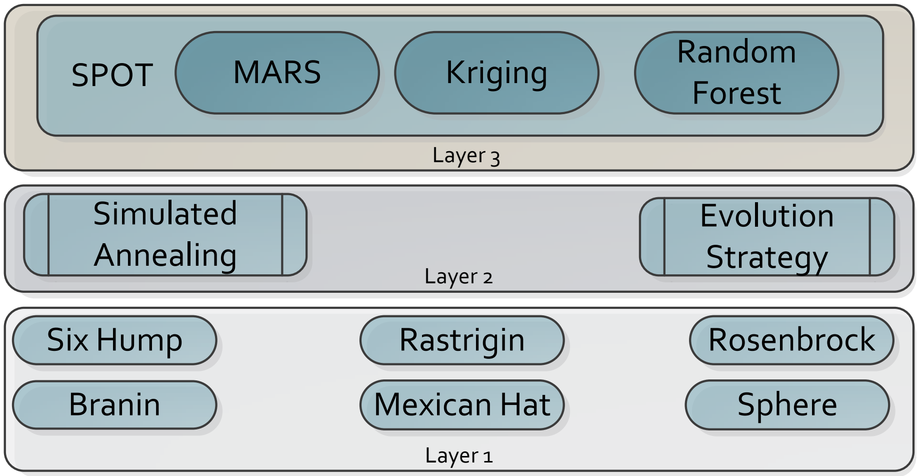

Figure 3: Experimental setup. The first layer comprehends the set of test functions. The optimization algorithms (O) with related designs ($\vec{X}$) belong to the second layer, whereas SPOT is in the third layer.

5.2 Case Study 2. Multi-Criteria SPOT

The second case study tries to use the MSPOT established above to tune algorithms, namely an Evolution Strategy (ES) and Simulated Annealing (SANN). The first objective for tuning is the quality of ES or SANN (minimal expected test function value) based on design points $\vec{x}$ (algorithm parameters) found by SPOT. The second objective is the standard deviation of the y's, to evaluate the robustness of the parameter setting $\vec{x}$. MSPOT considers both objectives. In this case, we consider a run of an algorithm, such as ES or SANN, as an optimization problem O. This situation is depicted in Fig. 3. The test functions (Layer 1) are optimized by SANN or ES (Layer 2) which again are tuned by SPOT or MSPOT (Layer 3). Starting with Layer 3, we will discuss the corresponding experimental setups.

5.2.1 Pre-experimental Planning: Problem Complexity and Structure

To gain insight into the problem complexity and structure, we performed a sweep over the search space. Ten thousand runs of the algorithms were executed, i.e., 1000 design points were evaluated ten times. For each design point, mean value and standard deviation were calculated. Here, $\overline{Y}$ denotes the mean value of k runs of the algorithms, and $\mathrm{sd}(Y)$ its standard deviation.

For example, if an ES with parameter set $\vec{x}$ is run ten times with different seeds on Rosenbrock's function, we obtain ten function values $y_1, \ldots, y_{10}$. As can been seen from Table 5, we are considering minimization problems, i.e., smaller y values are better.

Table 3: Layer 3. Setup of multi- and single-criteria SPOT for tuning ES and SANN

Parameter	MSPOT	SPOT
spot.ocba	FALSE	TRUE
auto.loop.nevals	200	200
init.design.size m	10	10
init.design.repeats k_0	5	5
seq.design.size l	1000	1000
seq.design.new.size a	5	3
seq.design.oldBest.size	0	2
seq.design.maxRepeats k	5	Inf

5.2.2 Experimental Setup

Experimental Setup on Layer 3. The multi-criteria MSPOT approach is compared to the standard, single-criteria SPOT approach. To evaluate the quality of this approach, the resulting Pareto fronts will be compared against the standard deviation and mean quality of a single objective run of SPOT. This run will use optimal computational budget allocation (OCBA) [6] and thus also consider standard deviation in a way. For OCBA, standard deviation will play a role in deciding how often a certain setting is to be evaluated [1]. However, the standard deviation is not considered with single-criteria SPOT and OCBA for the selection of new points. Thus, it is of interest to see how the one best setting found by SPOT and OCBA compares to the front of MSPOT. The settings used for the multi- and single-criteria SPOT runs are listed in Table 3, the surrogate models are the same as described beforehand. The sequentially generated surrogate models are optimized using the sampling approach as described in Algorithm 1. The surrogate models themselves are not optimized by means of MCO. This ensures a fair comparison of the multi- and single-criteria approaches, because the choice of completely different optimization techniques is avoided. The standard deviation recorded for each design point of the MSPOT runs is based on $k = k_0 = 5$ repeats. For OCBA k is increased as needed by SPOT. Therefore, the best design point of the single-criteria SPOT run will be reevaluated to ensure that its standard deviation is based on the same number of repeats.

Experimental Setup on Layer 2. ES and SANN are allowed to use 100 function evaluations of the test functions. For SANN, the `optim` function in R is used, which also includes a simulated annealing version as described by Belisle [3]. SANN requires a start point, which is chosen deterministically as mentioned above. All algorithm settings are left at default, besides TMAX and TEMP which are both tuned by SPOT with the region of interest as specified by Table 4. TEMP is the starting temperature and TMAX indicates the maximum number of function evaluations for each temperature.

ES internally creates a random starting population based on the bounds

Table 4: Layer 2. ES and SANN Parameters, which are tuned

Algorithm	Parameter	lower bound	upper bound
ES	μ	2	10
ES	ν	2	10
ES	τ	1	1.5
SANN	temp	1	100
SANN	tmax	1	100

Table 5: Layer 1. Description of two dimensional single objective test functions used in experiments for tuning ES and SANN

Function	Start point	lower bound	upper bound
Branin	(6,10)	(-5,0)	(10,15)
SixHump	(1.5,0.8)	(-1.9,-1.1)	(1.9,1.1)
Rastrigin	(2,6)	(-5.12,-5.12)	(5.12,5.12)
Rosenbrock	(-1.2,1)	(-2,-2)	(2,2)
MexicanHat	(0.5,-1.5)	(-8,-8)	(8,8)
Sphere	(4,3)	(-5,-5)	(5,5)

given for the target function. The number of parents μ is tuned as an integer. The selection pressure ν as well as the learning parameter for self adaption τ are tuned as real valued numbers (float). They are varied as listed in Table 4. Each combination of the two algorithms and the six test functions is one optimization problem O to be tuned by SPOT and MSPOT, leading to 12 different problems.

Experimental Setup on Layer 1. The single objective numeric test functions and their relevant settings are summarized in Table 5. Their start point is chosen with a deterministic value to make sure that less additional noise distorts the results of the comparison. All start points are manually chosen to have a sufficient distance from the global optima, and not to have symmetric values. The implemented set of single criteria test functions is used as described in detail by [1]. In order to detect obvious effects and abnormalities, the well-known sphere function was added to our set of test functions.

5.2.3 Evaluation

We have chosen three surrogate models M as shown in Table 2, two algorithms and six test functions, altogether $3 \times 2 \times 6 = 36$ configurations are considered. In order to keep the complexity manageable, we decided for the following procedure:

1. First, both MSPOT and SPOT are run for each algorithm-surrogate-problem combination. For example, MSPOT is used with random forest to tune ES on Rosenbrock's function.
2. Second, we will determine the best surrogate model, say M^*, for MSPOT and SPOT independently.

3. The final evaluation comprehends $2 \times 6 = 12$ comparisons. Mean best function values and standard deviations of MSPOT and SPOT are compared.

6 Results

6.1 Case Study I. Comparison

Results from runs with limited budgets of function evaluations (100 and 500, respectively) are presented in Fig. 4. The contrast between these results is obvious. While the direct approaches SMS-EMOA and NSGA2 clearly outperform most of the MSPOT approaches after 500 function evaluations, their performance in comparison is much worse after only 100 function evaluations. After 100 function evaluations, almost all MSPOT approaches perform better than SMS-EMOA and NSGA2 do. This particularly holds for the DTLZ test functions. On the ZDT functions, the direct approaches perform at least better than MSPOT incorporating the sFO model. This is probably due to the high input dimension. Kriging is rarely reported to work well for high dimensions, i.e., greater than 20. [15] We feature 30.

Both remaining other models sEA and sRF perform better than the direct approaches on all test functions. Comparing these two, sEA beats sRF on all test function except for ZDT3. A possible correlation to the discontinuity of the Pareto front of this test function cannot be neglected, but is not investigated any further at the moment.

After 500 function evaluations, sEA still is the best MSPOT approach and it also outperforms SMS-EMOA and NSGA2 for $n = 2$ (ZDT functions). However, all other MSPOT based approaches yield worse results. This also holds for $n = 3$ (DTLZ functions), except for sEA performing worse than sFO and sRF here. The better performance of sFO might be due to the lower dimension of the decision space. Another interesting observation is that NSGA2 performs bad on DTLZ2, while all other approaches do much better. In particular, the other direct approach, SMS-EMOA, is the best algorithm on this function.

In Fig. 5 the hypervolume growth of MSPOT with sEA is compared to SMS-EMOA on DTLZ2. The course of the hypervolume and the corresponding standard deviation for each generation is plotted against the corresponding number of function evaluations.

The figure shows that MSPOT clearly outperforms the SMS-EMOA in the beginning, for the first 300 generations approximately. After around 300 function evaluations SMS-EMOA takes the lead. The exact crossing point varies for different settings, however SMS-EMOA always overtakes the MSPOT approach for the DTLZ functions. This behavior cannot be reported for the first 1000 function evaluations on the two dimensional test cases.

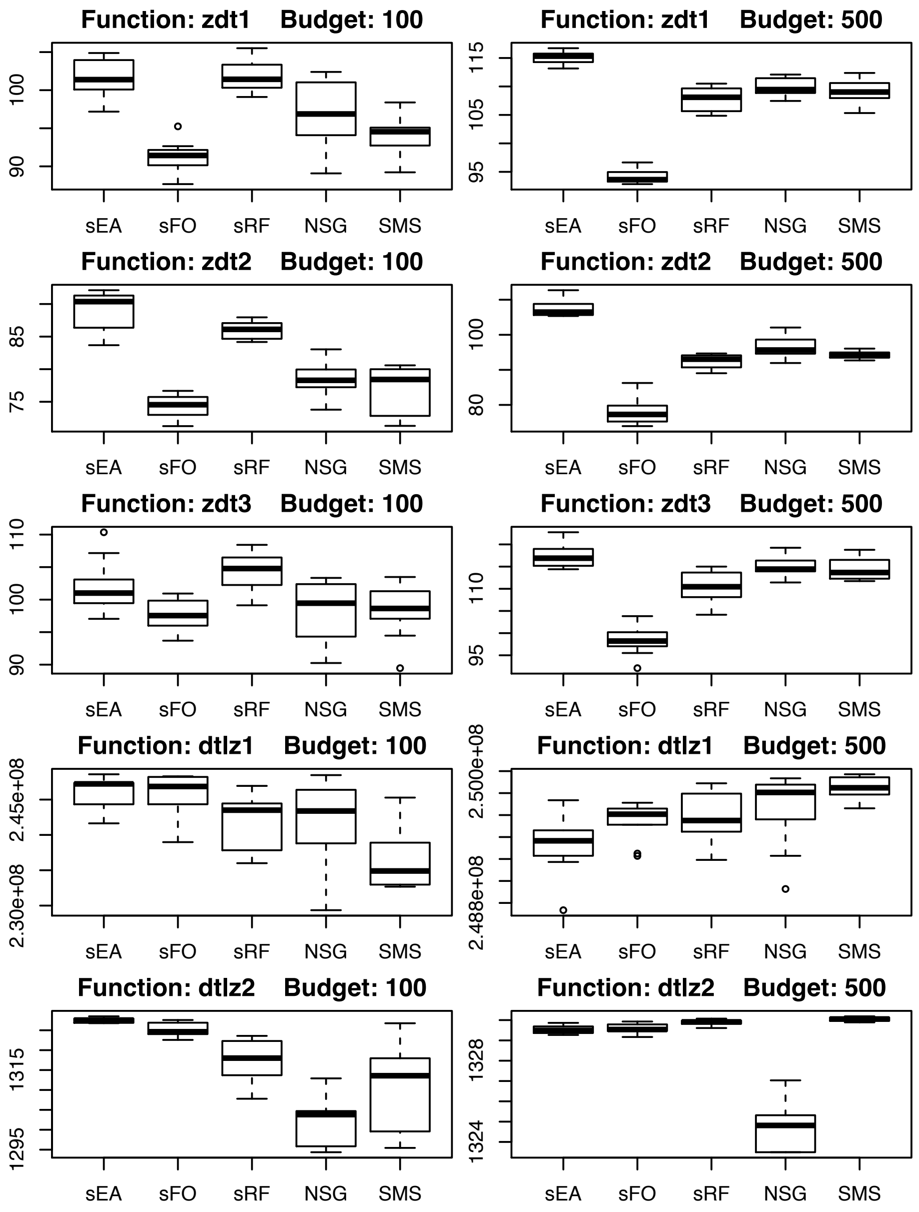

Figure 4: Boxplot of hypervolumes on all test functions for different budgets. The left column presents the results with a budget of 100, the right one with 500 function evaluations. Corresponding test functions are in one line.

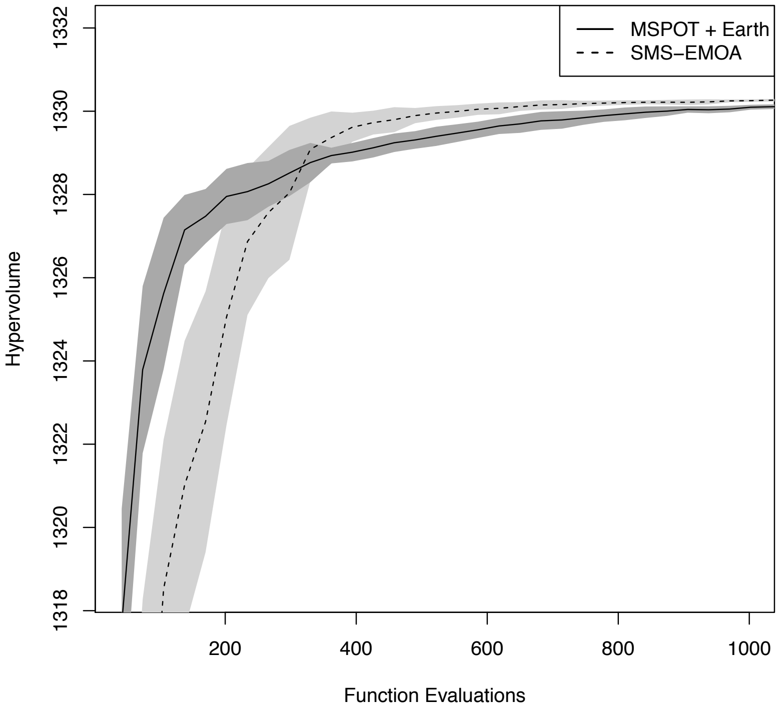

Figure 5: Mean hypervolume growth on DTLZ2 with a budget of 1000 function evaluations. The colored areas show +/- one standard deviation. The used design or population size is 32.

6.2 Case Study II: Multi-Criteria SPOT

6.2.1 Results from the Pre-experimental Study

In every experimental setup from our study, $\mathrm{sd}(Y)$ and $\overline{Y}$ are correlated. We observed that a parameter setting $\vec{x}$, which results in a good mean function value, has a low standard deviation. As shown in Fig. 7, contour plots based $\overline{Y}$ and $\mathrm{sd}(Y)$ present similar information. Moreover, the correlation between $\overline{Y}$ and $\mathrm{sd}(Y)$ increases for good parameter settings. Figure 6 clearly illustrates this correlation. Pearson's product-moment correlation of $\overline{Y}$ and $\mathrm{sd}(Y)$ is 0.9314042 for Rosenbrock's function. Similar values were obtained with Rastrigin (correlation: 0.8242029), Branin (0.9024674), SixHump (0.8976964), MexicanHat (0.9370178), and Sphere (0.9791648).

6.2.2 Results from the Comparison

First, we analyzed which model M performs best for each SPOT variant: MARS, Kriging, or Random Forest. In both cases, sFo, i.e., Kriging, performs best. Therefore, it was chosen for the final comparison of the SPOT variants.

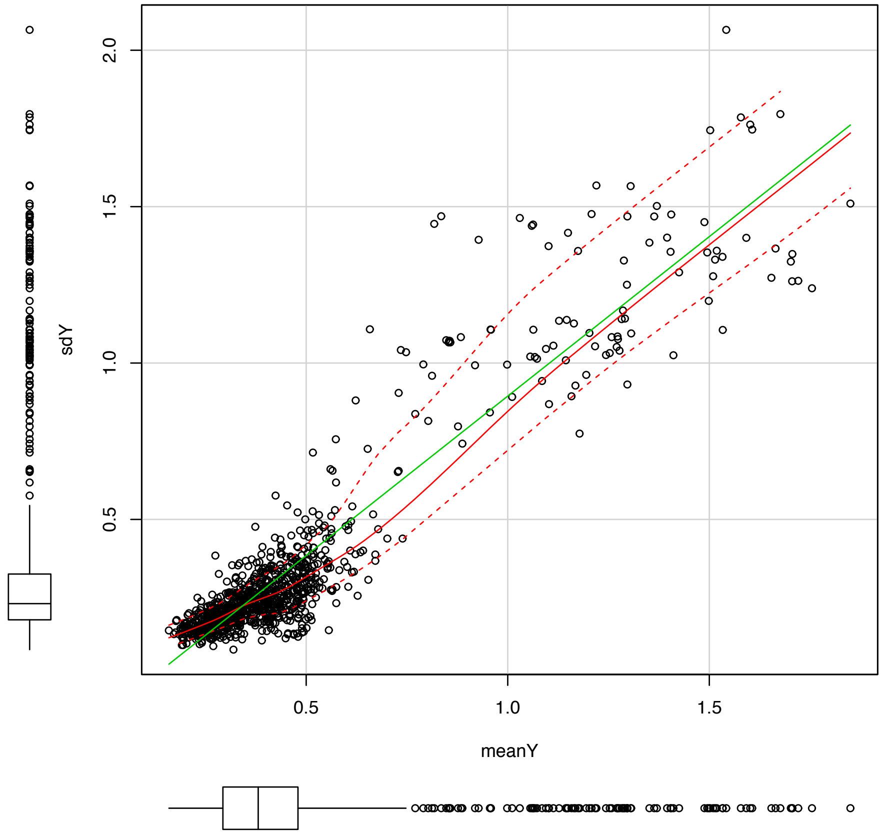

Figure 6: ES on Rosenbrock's function: standard deviation sd(Y) plotted against mean function value $\overline{Y}$. Similar values (correlation > 0.8) were calculated for remaining test functions. Results are based on 10,000 ES algorithm runs. Note that SPOT uses 200 algorithm runs (or even less) to determine good design points.

Standard SPOT determines a design point, whereas MSPOT generates a set of (Pareto optimal) design points. Since good design points, i.e., design points with a low $\overline{Y}$ value, are expected to have a low associated standard deviation, the design point with the best $\overline{Y}$ value was chosen from the Pareto front for the final comparison with SPOT.

A typical result from these final comparisons is shown in Fig. 8. It also shows by example how ES performed better than SANN on every test function.

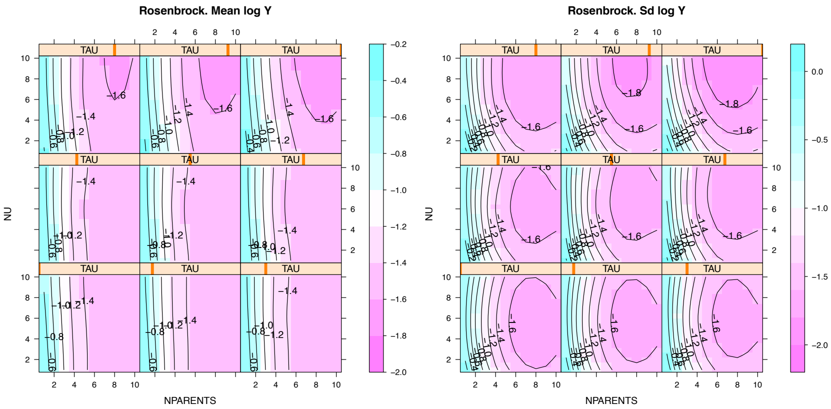

Figure 7: ES on Rosenbrock's function. These two figures were generated based on 10,000 ES algorithm runs to illustrate the correlation between mean $\overline{Y}$ and standard deviation sd(Y) of function values. Contour plots of function values (left) and standard deviations (right) show a very similar structure. Selection pressure is plotted against population size. The sliders on top of each panel shows the variation of the learning rate. NPARENTS: number of parents μ. NU: selection pressure ν. TAU: learning parameter for self adaption τ.

7 Analysis and Interpretation of Results

Analyzing results from the first case study lead to the following conclusion, which gives an answer to research question 1.

Conclusion 1 *On the lowest budget, which we stated earlier as being most promising for* MSPOT*, it is shown that* SPOT *methods outperform NSGA2 or SMS-EMOA on a majority of test functions.* □

Regarding the performance of MSPOT versus SPOT, our experiments show that MSPOT performs equally good or even slightly better than the standard approach.

Conclusion 2 *Integrating the standard deviation of a solution can be beneficial for the search process.* □

Although conclusion 2 requires further investigation, we are optimistic that enhanced MSPOT variants might result in a performance boost. However, MSPOT is not the first approach, which combines function values and standard deviations. For example, chapter 3 in Forrester et al. [10] presents fundamental ideas in a very comprehensive manner. Hence, analyzing and improving the MSPOT approach is of great interest, not only, because it performs slightly better than the single objective approach. Especially in uncertain real-world settings, it is desirable to provide a set of trade-off solutions between robustness

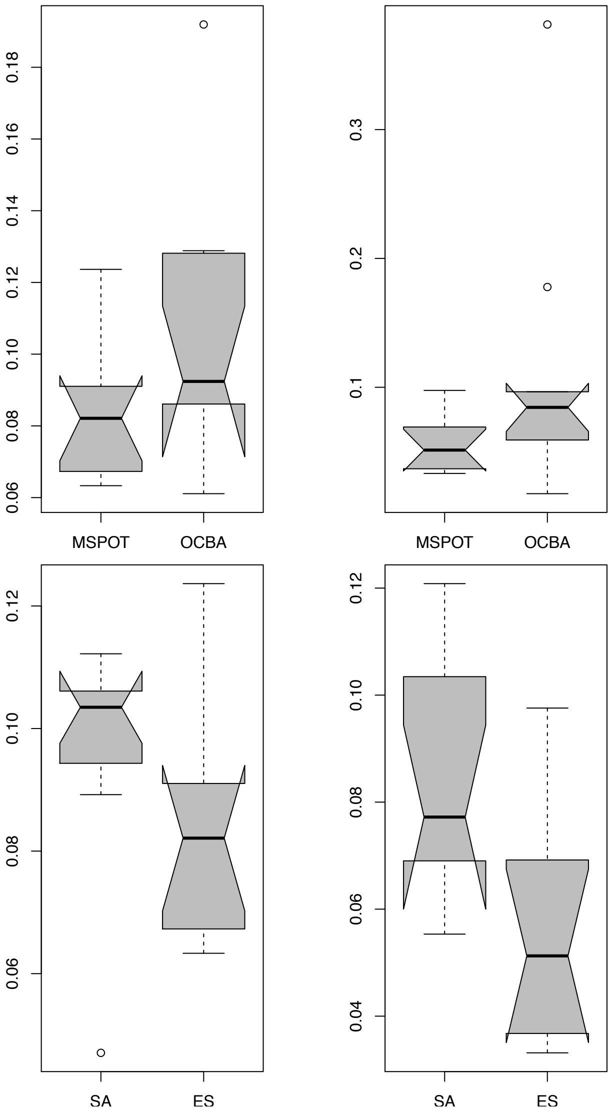

Figure 8: Results on Rosenbrock. Left: function value, right: standard deviation. Upper: MSPOT vs. OCBA-SPOT, both on ES. Lower: ES vs. SANN, both tuned with MSPOT.

and optimality to a human decision maker. Here, the decision maker has to make a choice according to the need of the particular application.

In addition to conclusions related to research questions 1 and 2, we made interesting observations about the correlation structure of $\overline{Y}$ and sd(Y). As illustrated in Fig. 6, $\overline{Y}$ and sd(Y) are positively correlated if minimization problems are considered. We can conclude from these observations that parameter designs, which were used during the final stage of an algorithm tuning, have both low $\overline{Y}$ and low sd(Y) values. Further analysis reveals interesting interactions between solution quality and standard deviation, which gives reason for additional research. However, based on the actual data from our experiments, we can state the following conclusion.

Conclusion 3 *If the algorithm improves during the optimization, the final* $\overline{Y}$ *values have a small standard deviation. If the optimization of the algorithm fails, then the standard deviation is relatively high.* □

8 Summary and Outlook

A multi-criteria approach to SPO was defined in this paper. It was shown that MSPOT can be applied successfully to solve hard MCO problems with a strictly limited budget. Moreover, MSPOT was applied to single objective algorithm tuning, by considering standard deviation of results as a second objective. Both SPOT and MSPOT were able to find good parameter settings. Results indicated that there is actually a high correlation between standard deviation and solution quality, if the solution is in the vicinity of the optimum. It could be observed however, that MSPOT can find better solutions than single objective SPOT on this problem type. The proposed MSPOT approach can be improved in many ways: Several SPOT features can be integrated into MSPOT, e.g., an adaptation rule for the number of repeats, k, on noisy problems or an optimization on the surrogate models. Moreover, the choice of population sizes for this internal optimization of the surrogates should be made independent of the used sequential budget a. And last, but not least, MARS can be used with more sophisticated parameterizations. Finally, a main focus of further research will be to test MSPOT on real industrial problems like the structural optimization of dust filtering cyclones. Such applications are the driving force behind this research.

References

[1] T. Bartz-Beielstein and M. Friese. Sequential parameter optimization and optimal computational budget allocation for noisy optimization problems. CIOP Technical Report 02/11, January 2011.

[2] T. Bartz-Beielstein, K. E. Parsopoulos, and M. N. Vrahatis. Design and analysis of optimization algorithms using computational statistics. *Applied Numerical Analysis and Computational Mathematics*, 1(2):413–433, 2004.

[3] C. J. P. Belisle. Convergence theorems for a class of simulated annealing algorithms. *Journal Applied Probability*, 29:885–895, 1992.

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

[5] L. Breiman. Random forests. *Machine Learning*, 45(1):5 –32, 2001.

[6] C. H. Chen. An effective approach to smartly allocate computing budget for discrete event simulation. In *Proceedings of the 34th IEEE Conference on Decision and Control*, pages 2598–2605, 1995.

[7] C. A. Coello Coello, D. A. Van Veldhuizen, and G. B. Lamont. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Springer, New York, 2007.

[8] K. Deb. *Multi-Objective Optimization using Evolutionary Algorithms*. Wiley, New York, 2001.

[9] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. Scalable Test Problems for Evolutionary Multi-Objective Optimization. Technical Report 112, Institut für Technische Informatik und Kommunikationsnetze, ETH Zürich, 2001.

[10] A. Forrester, A. Sobester, and A. Keane. *Engineering Design via Surrogate Modelling*. Wiley, 2008.

[11] J. H. Friedman. Multivariate adaptive regression splines. *Ann. Stat.*, 19(1):1–141, 1991.

[12] N. Hansen. The CMA evolution strategy: a comparing review. In J. Lozano, P. Larranaga, I. Inza, and E. Bengoetxea, editors, *Towards a new evolutionary computation. Advances on estimation of distribution algorithms*, pages 75–102. Springer, 2006.

[13] Y. Jin. A comprehensive survey of fitness approximation in evolutionary computation. *Soft Computing*, 9(1):3–12, 2005.

[14] Y. Jin and B. Sendhoff. Trade-off between performance and robustness: an evolutionary multiobjective approach. In *Evolutionary multi-criterion optimization (EMO 2)*, pages 237–251. Springer, Berlin, 2003.

[15] M. Preuss, T. Wagner, and D. Ginsbourger. High-Dimensional Model-Based Optimization Based on Noisy Evaluations of Computer Games. In *Learning and Intelligent Optimization Conference (LION 6)*. Springer (to appear), 2012.

[16] I. Voutchkov and A. Keane. Multiobjective optimization using surrogates. In *Adaptive Computing in Design and Manufacture ACDM*, 2006.

[17] E. Zitzler, K. Deb, and L. Thiele. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evolutionary Computation*, 8(2):173–195, 2000.