Multi-Objective Optimization for Hard Real-World Problems under Limited Budgets

Martin Zaefferer
martin.zaefferer@smail.fh-koeln.de

Thomas Bartz-Beielstein
thomas.bartz-beielstein@fh-koeln.de

Martina Friese
martina.friese@fh-koeln.de

Boris Naujoks
boris.naujoks@fh-koeln.de

Oliver Flasch
oliver.flasch@fh-koeln.de

ABSTRACT

Sequential parameter optimization (SPO) is a heuristic that combines classical and modern statistical techniques to improve the performance of search algorithms and optimization tasks. Since only a fraction of the interesting industrial Problems are single objective it is fundamental to provide algorithms with the SPO-Toolbox (SPOT) to optimize multi objective problems.

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.

In this Paper we show that multi objective optimization can be efficiently done with SPOT. Furthermore we present an approach to optimize not only the quality of the solution, but also its robustness, taken these two goals as criteria for multi objective optimization. The results that we are presenting show...

Keywords

Sequential Parameter Optimization, Multi Criteria Optimization, Robustness

1. INTRODUCTION

In many fields of industrial optimization, the duration of a process feedback plays a determining role for the optimization process. Large evaluation times, caused by long time real-world processes or large computational processing times, restrict optimization processes to only a very limited number of such evaluations.

Moreover, most of industrial optimization tasks feature more than one quality criterion. Techniques from multi criteria decision making, evolutionary multi objective optimization (EMO) in particular are considered the last decade to solve such tasks. The necessity to combine EMO techniques and optimization methods requiring only a very small number of function evaluations should be self-evident.

A typical real-world application is the optimization of structural parameters and process parameters of industrial process plants. Currently, preparations are made to apply multi objective SPOT (MSPOT) to the optimization of dust seperators for coal power plants. 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 behaviour of cyclones can be simulated with a high degree of accuracy by using compute intensive numerical methods, such as multi-phase computational fluid dynamics (CFD). Even if executed on a compute cluster, a single CFD run will require from several minutes to hours of processing time, severely limiting the number of possible evaluations.

Typical parameter tuning problems as solved with Sequential Parameter Optimization Toolbox (SPOT) can include the optimization of industrial problems, or the tuning of algorithm parameters. In both cases, it can occur that not one but two or more objectives are of interest. 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 multi criteria problems.
2. Compare the multi objective optimization approach to state of the art algorithms in the field. Test situations should reflect aspects of industrial tuning problems like high cost for function evaluations.
3. Test how well the robustness of tuning algorithm parameters with SPO can be increased by considering quality and variance as objectives.

The research goals are presented in more detail in Sec. 2. Section 3 gives an introduction to the topic of MCO. The following section 4 describes how MCO can be done with SPO. Experimental settings for the purpose of testing that approach are described in section 5. The results of these experiments are reported and analyzed in section 6 and 7. Finally the conclusions of this work as well as an outlook on future research is given in section 8.

2. RESEARCH GOALS (TBB)

Case 1: Is there a competitive advantage of MSPOT against state of the art MCO algorithms?

It is of interest if MSPOT can compete with or outperform state of the art MCO algorithms. Outperform means

GECCO'12, July 7-11, 2012, Philadelphia, Pennsylvania, USA.
Copyright 2012 ACM 978-1-4503-1177-9/12/07 ...$10.00.

to reach higher hypervolumes on multi objective test problems. As stated above real target functions can be costly to evaluate. Therefore the test problems are restricted to few function evaluations. A low budget for the depicted area of application would be a few tenth or few hundreds of function evaluation. To show development beyond these limits we chose to test up to a maximum budget of 1000. The state of the art MCO algorithms in this comparison are chosen to be NSGA2 and SMS-EMOA.

Case 2: Is Multi-objective Model Optimization advantageous for robust parameter optimization?

Robustness can be defined by using the following goals:

- To optimize the expectation of the objective function in a neighborhood of a target point.
- To minimize the second order moment (variance) or higher order moments of the objective function.

If optimization of expectation is chosen as the only objective, it can happen that positive and negative deviations of the function cancel each other in the neighborhood of a target point. Thus, an undesirable solution will be considered to be robust. On the other hand, if optimization of the variance is chosen as the only objective, the optimizer may find a plateau of the original objective function, which may not be optimal. Therefore, it is sometimes insufficient to optimize the expectation or to minimize the variance only in the search of robust optimal solutions. In other words, the search for robust optimal solutions should be basically addressed as a trade-off between optimality and robustness.

In practice it is desirable to present a human user with a set of solutions trading of between the robustness and the optimality, from which the user has to make a choice according to the need of the application.

This paper considers the search of robust solutions as a multiobjective optimization problem. To this end, a measure for robustness based on the variance of the objective function in the presence of noises has been introduced, which is used as a separate objective in evolutionary optimization. [10] uses a variance-based robustness measure to deal with the search of robust solutions as a multiobjective problem. They suggest two methods for estimating the robustness measure using the information in the current design solely.

With the robustness measures defined above, it is then possible to explicitly treat the search of robust optimal solutions as a multiobjective optimization problem. The main advantage of the proposed multiobjective approach to the search of 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 best deal with the problem at hand.

3. MULTIOBJECTIVE OPTIMIZATION

The field of multiobjective optimization 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) that is also covered in this paper, were evolutionary multiobjective optimization algorithms (EMOA,[7, 6]).

The space that 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 the SMS-EMOA ([4]), which, next to other implementations, is on the way to replace the former standard NSGA-II [7]. To this end, the methods proposed for MOO with a very limited amount of function evaluations are compared to both of these algorithms, i.e. NSGA-II as well as SMS-EMOA.

Algorithm 1: MSPOT

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

4. MULTI OBJECTIVE SEQUENTIAL PARAMETER OPTIMIZATION (MZ)

To understand how the multi objective SPOT (MSPOT) approach differs, the single objective SPOT has to be introduced briefly. 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, tree based). Two approaches can be used to exploit that model: The naive approach samples a large number of points in the decision space. Those points are evaluated on the surrogate models and the best will be suggested for evaluation on the problem function. More sophisticated approaches use well known optimization techniques to find the optimum of the surrogate model. The optimum will then be suggested for evaluation on the 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 6–13).

In Phase 1 SPOT determines a design of initial points in the problems decision space and evaluates each point k_0 times on the problem function. Here, the only difference to single objective SPOT is that the evaluation yields not a scalar result for each design point, but a vector where each element is one of n objective values to be optimized.

Phase 2 consists of a loop with the following components:

1. Build a model M for each of the n objectives with the known design values. This can actually result into n univariate models or into one multivariate multiple regression model. Those kind of models consider both multiple response variables and multiple predictor variables. This allows to consider correlation between the response variables.
2. Randomly generate a (large) set of l design points $\vec{X}'$ and compute their utilities $\vec{Y}'$ by evaluating them on the build model(s).
3. Select a (which has to be smaller than l) seemingly optimal design points from the large design. This can not be achieved by simply sorting the results $\vec{Y}'$, since two or more objective values are available for each sampled design point. Instead the design points are first sorted by their non dominated sorting rank. That means, 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 value. This is repeated until all points in the objective space have a rank value. This works well for a first sorting, but it will assign the same value to all points on an individual Pareto front. So in case the settings of MSPOT require less values than are available for the current non dominated sorting rank there has to be a tie breaker. This is done by calculating the hypervolume contribution of each point, and removing those with the least contribution. This is recalculated sequentially after removing one point, 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 multi objective optimization techniques are employed. In the current implementation this can either be NSGA2 or SMS-EMOA. They 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 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 on each design point regularly. In that case a counter k is increased in each cycle and used to determine the number of repeats that are performed for each setting to be statistically sound in the obtained results. 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 fixed number of k_0 for repeats.

The general MSPOT loop is illustrated in Fig. 1.

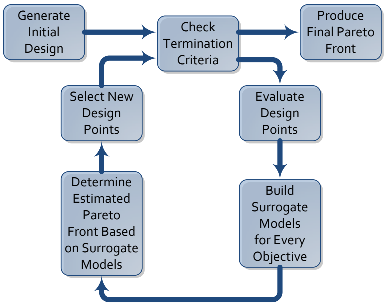
Generate Initial Design
Check Termination Criteria
Produce Final Pareto Front
Select New Design Points
Evaluate Design Points
Determine Estimated Pareto Front Based on Surrogate Models
Build Surrogate Models for Every Objective

Figure 1: The general MSPOT loop

Table 1: Multiobjective test functions used here. Dimensions d and n are 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. Lower and upper are the bounds of the decision space.

Function	Dimension (d,n)	lower	upper	ref. point
ZDT1	(30,2)	0	1	(11,11)
ZDT2	(30,2)	0	1	(11,11)
ZDT3	(30,2)	0	1	(11,11)
DTLZ1	(7,3)	0	1	(1000, 500,500)
DTLZ2	(12,3)	0	1	(11, 11,11)

5. EXPERIMENTAL SETTINGS

The multiobjective version of SPOT as established above is tested in two different scenarios. In the first case typical multi objective 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 minimalistic available budget. In the second case, the approach is tested with problems that might represent an interesting use case, as described in the motivation section. That use case is robust tuning of algorithm parameters by considering both expectation and variance.

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 multiobjective test functions, namely ZDT1 to ZDT3 [13], DTLZ1, and DTLZ2 [11]. The multiobjective test functions and their relevant settings are summarized in Table 1.

The settings used for the MSPOT runs are listed in Table 2. The number of repeats k is fixed to a value of one, since 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. [8] The model uses CMAES [9] to find its parameters.
2. spotPredictRandomForest (sRF): Random Forest method from the R package ”randomForest” which is based on

Table 2: Additional setup of MSPOT

Parameter	setting in MSPOT
auto.loop.nevals	1000
spot.ocba	F (false)
init.design.size m	100
init.design.repeats k_0	1
seq.design.maxRepeats k	1
seq.design.new.size a	32
seq.design.oldBest.size	0
seq.predictionModel.func M	”spotPredictForrester” ”spotPredictRandomForest” ”spotPredictEarth”
seq.predictionOpt.budget	1000
seq.predictionOpt.func	”spotParetoOptMulti”
seq.predict.subMethod	”nsga2”

Breiman and Cutler’s original Fortran code for classification and regression [5].

3. spotPredictEarth (sEA): Multivariate adaptive regression splines (MARS) provided by the ”earth” package Here, multivariate does not refer to multivariate multiple regression. MARS considers each variable seperately.

For all models default settings are used. For the MARS model that means that only linear terms are considered, 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 because they 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.

For SMS-EMOA, a straight forward implementation from the R package ”emoa” was used, the used implementation of NSGA2 is part of the ”mco” package[1]. The population size of NSGA2 and SMS-EMOA was chosen to be in line with the seq.design.new.size parameter a. Besides the population size and the budget (or number of generations), all parameter settings of SMS-EMOA and NSGA2 were left at default values.

Another important point is, as outlined in the research goals, that the function evaluation budget should be rather low. Testfunctions like mentioned in this section are usually optimized invoking several thousand of function evaluations by NSGA2 or SMS-EMOA. This will be lowered to 100, 500, and 1000 function evaluations. It is the scope of this paper to identify EMOA that cope with the very small number of fitness function evaluations (FFE) from the industrial applications of interest.

Since low budgets would limit the number of generations, the population size (or 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).

5.2 Case Study 2. Multi-Objective SPOT

The second case study tries to use the above established MSPOT to tune algorithms, namely ES and SANN. The first objective for tuning is the quality of ES or SANN (minimum function value y reached on a test function from layer 1) 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 $\vec{x}$. 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. 2. 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.

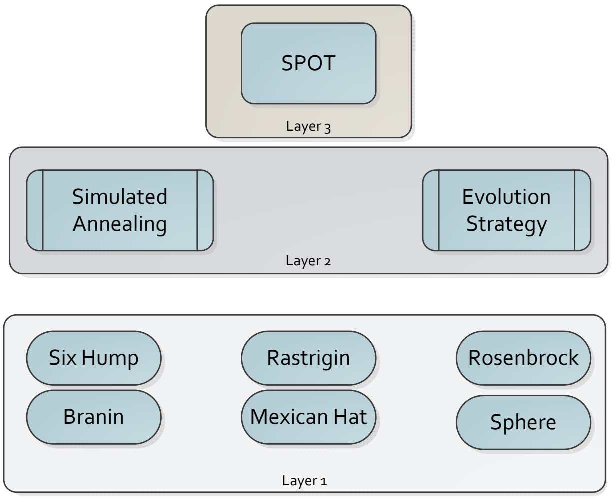

Figure 2: Experimental setup. The first layer comprehends the set of test functions. The optimization algorithms (O) with related designs ($\vec{X}$) belong to the second layer, whereas SPOT **is in the third layer.**

5.2.1 *Pre-experimental Planning: Problem Complexity and Structure*

To get some 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 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 in Tab. 5, we are considering minimization problems, i.e., smaller y values are better.

5.2.2 *Experimental Setup*

Experimental Setup on Layer 3.

The multi-objective MSPOT approach is compared to the standard, single-objective 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, and thus also consider the variance in a way. For OCBA, variance will play a role in deciding how often a certain setting is to be evaluated.[2] However, for selection of new points the variance is not con-

[1] All R packages can be received from the CRAN homepage, i.e. http://cran.r-project.org

Table 3: Layer 3. Setup of multi and single objective 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	2
seq.design.size l	2000	2000
seq.design.new.size a	8	6
seq.design.oldBest.size	0	2
seq.design.maxRepeats k	5	Inf

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

sidered with single objective SPOT and OCBA. Thus it will be interesting 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 objective SPOT runs are listed in Table 3, the surrogate models are the same as described in the previous section. The sequentially created 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-objective 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 $k = 5$ repeats. For OCBA k is increased as needed by SPOT. Therefore, the best design point of the single objective 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 on the numeric 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 target function evaluations (e.g., on Branin or SixHump) for each temperature.

When ES is tuned, no start points are used as the ES internally creates a random starting population based on the bounds 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.

Table 5: Layer 1. Description of two dimensional single objective testfunctions 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)

Experimental Setup on Layer 1.

The single objective numeric test functions and their relevant settings are summarized in Table 5. All of them are either defined or chosen to have a two-dimensional input space. Their start point is chosen with a deterministic value, to make sure that less additional noise distorts the results of the comparison. More et. al. [12] list a number of deterministic start points for different test functions. Therefore, the start point for the Rosenbrock function was chosen accordingly. The other 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 objective test functions is used as described in detail by [1]. Only a simple sphere function with a global optimum at zero was added to this set.

5.2.3 Evaluation

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

Figure 3 shows box plots of mean hypervolumes after 10 repeats for each run with a budget of 100 and 500 FFE. sLM is missing for the cases with a budget of 100 FFE, since it needs a rather large initial design.

NSGA2 and SMS-EMOA clearly outperform most of the SPOT approaches after 500 FFE. Earth is still better than NSGA2 and SMS-EMOA on the ZDT1-3 functions.

The results after a budget of one thousand FFE are not reported here. These show that earth is not better than NSGA2 or SMS-EMOA after this amount of FFE, but still competitive.

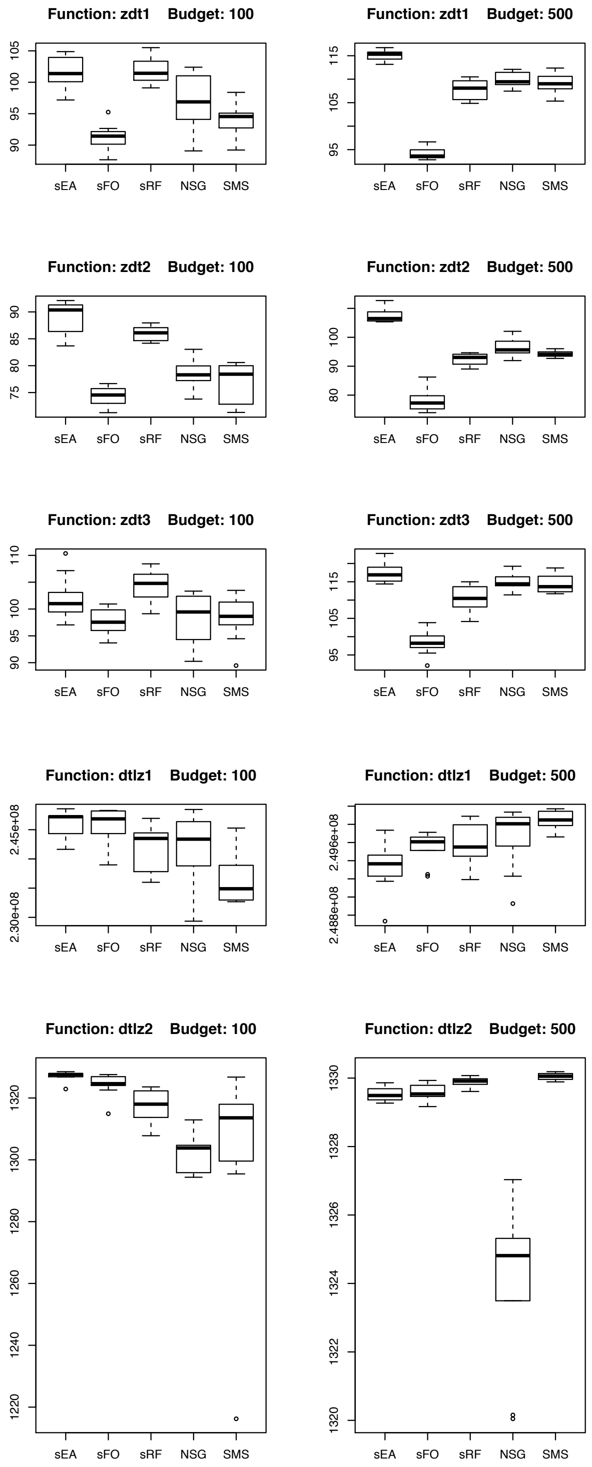

Figure 3: Boxplot of hypervolumes on all testfunctions for different budgets

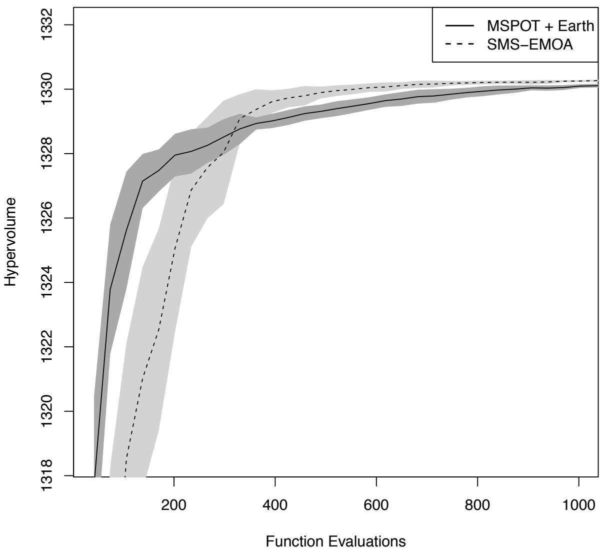

Figure 4: Mean hypervolumes of SMS-EMOA **and** MSPOT **on dtlz2 with a budget of 1000 function evaluations. The coloured areas show +/- one standard deviation. The used design or population size is 32.**

To show the development over the whole process figure 4 compares the hypervolume growth of MSPOT + earth with SMS-EMOA on dtlz2. This shows how MSPOT clearly outperforms the SMS-EMOA at the start, while after a few hundred evaluations SMS-EMOA takes the lead. The exact crossing point would vary for different settings. For some of the experiments MSPOT + earth would even keep the lead for the whole tested range.

6.2 Case Study II: Multi-objective SPOT

6.2.1 *Results from the Pre-experimental Study*

In general, $\mathrm{sd}(Y)$ and $\overline{Y}$ are correlated. We observed that a parameter setting $\vec{x}$, which result in good mean function value, has a low standard deviation. That is, the correlation between $\overline{Y}$ and $\mathrm{sd}(Y)$ increases for good parameter settings. Figure 5 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), Six-Hump (TBD), MexicanHat (TBD), and Sphere (TBD).

6.2.2 *Results from the Comparison*

First, we analyzed which model M performs best for each SPOT variant: earth, forrester, linear regression, or random forest. In both cases, SFO, i.e., Kriging, performed best. Since forrester performed best, it was chosen for the final comparison of the SPOT variants.

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

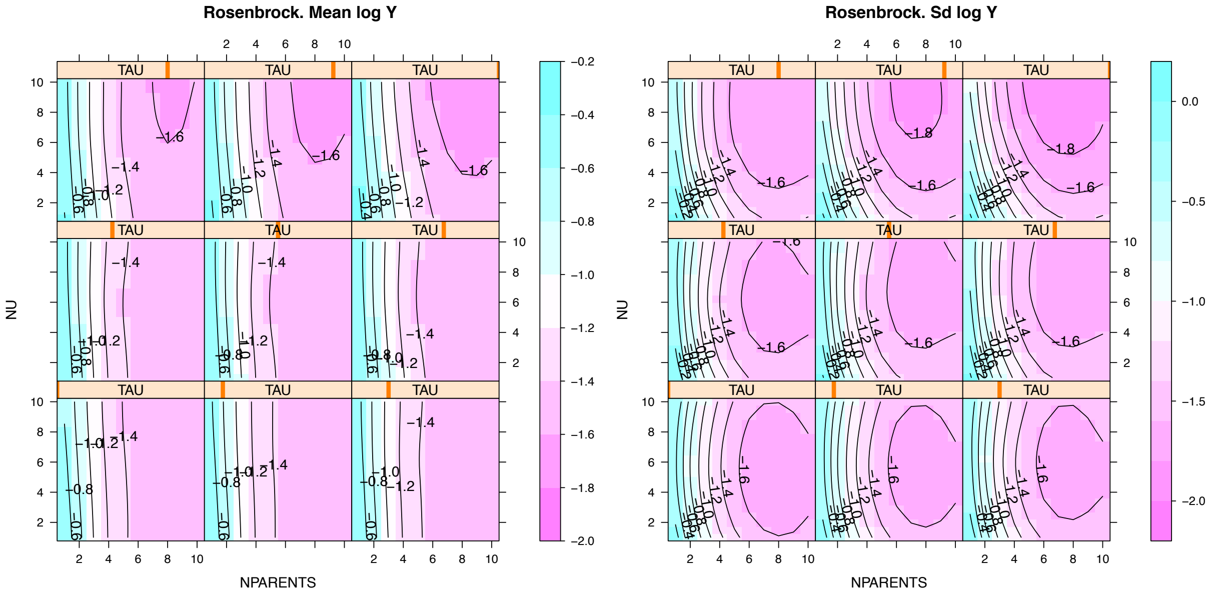

Figure 6: ES on Rosenbrock function: Sweep. Left: function value, right: standard deviation. The correlation between $\overline{Y}$ and sd(Y) is illustrated. Based on 10,000 ES algorithm runs

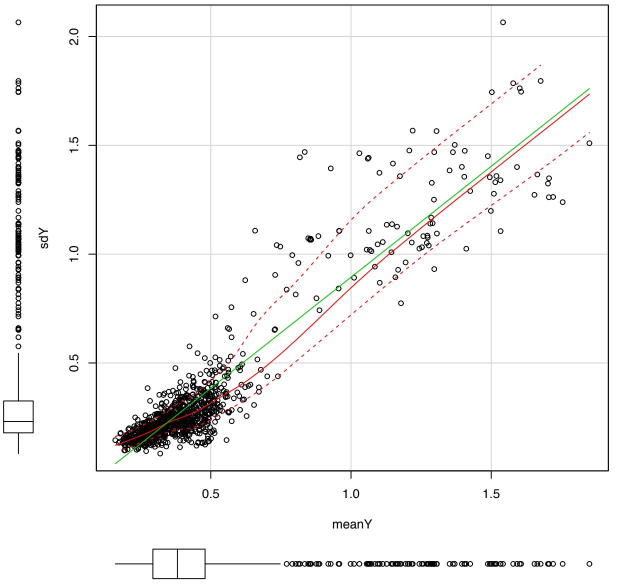

Figure 5: ES on Rosenbrock's function: standard deviation plotted against mean function value. Based on 10,000 algorithm runs. Note, SPOT **uses 200 algorithm runs (or even less) to determine good design points**

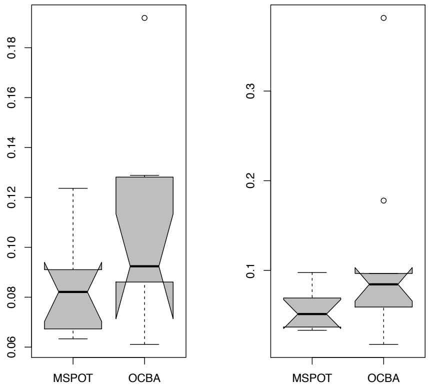

Figure 7: ES on Rosenbrock's function: MSPOT versus OCBA-SPOT. Left: function value, right: standard deviation

front for the final comparison with SPOT. Then pareto front large, otherwise pareto front consists of a few (or even one) point only. Therefore, comparison can be based on best ocba conf versus conf with best mean func conf from pareto front.

A typical result from these final comparisons is shown in Fig. 7. The multi-objective SPOT approach performed equally good or even slightly better than SPOT-OCBA on every test function.

ES performed better than SANN on every test function as can be seen from Fig. 8.

7. ANALYSIS AND INTERPRETATION OF RESULTS (TBB)

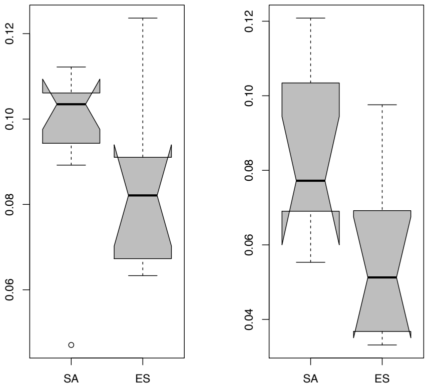

Figure 8: Rosenbrock function: ES **versus** SANN. **Left: function value, right: standard deviation. Both algorithms were tuned with** MSPOT. **TODO: Include default values from SANN and ES without tuning**

Analyzing results from the first case study lead to the following conclusion.

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.* □

Although SPOT usually works very well with Kriging models, it can be noticed that the Kriging approach used here performs not as good as Random Forest or Multivariate Adaptive Regression Splines. This is probably due to the high input dimension. Kriging is rarely reported to work well for high dimensions, i.e greater than 20. We feature 30.

As illustrated in Fig. 5, $\overline{Y}$ 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.

Conclusion 2. *If the algorithm improves during the optimization, the the final* $\overline{Y}$ *values have a small standard deviation. If the optimization of the algorithm fails, then the standard deviation is relatively high.* □

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

Conclusion 3. *Integrating the standard deviation of a solution can be beneficial for the search process.*

Although conclusion 3 requires further investigation, we are optimistic that enhanced MSPOT variants might result in a performance boost for MSPOT. However, MSPOT is not the first approach, which combines function values and variances. For example, chapter 3 in Forrester et al. [8] presents fundamental ideas in a very comprehensive manner.

8. SUMMARY AND OUTLOOK (MF)

Case 1 summary:
A multi objective approach to SPO was defined in this paper. The first question as outlined in our research goals, was if MSPOT can be applied successfully to solve multi objective problems with a strictly limited budget. The answer found for this is, that MSPOT proved to perform well when optimizing with a low function evaluation budget on typical MCO test functions. While the time consumption for MSPOT is significantly higher than for state of the art algorithms in MCO, it will be advantagous when applied for costly problems in hard real word problems. It seemed to work especially well with a Multivariate Adaptive Regression Splines model (earth).

Case 2 summary:
The second case study showed that SPOT and MSPOT both are able to find good paramter settings. In the results it was noticed that there is actually a high correlation between standard deviation and mean quality, if the solution is in a good region of the parameter space. At a first look that disqualifies the MSPOT approach since both objectives show similar interaction with the decision variables. It could be observed however, that MSPOT can find better solutions than single objective SPOT on this problem type. This could indicate that the additional objective provides SPOT with the information to find good solutions more frequently. Since both approaches use the same budget, this can be an improvement on comparable applications of SPOT.

outlook:
For future work it might be of interest to improve the MSPOT approach in general as well as details of the used models. General improvement might be to make use of SPOT features which were not applied to MSPOT yet, like increasing the number of repeats k on noisy problems (linearily or with OCBA). MSPOT might also profit if the naive sampling approach was used together with optimization on the surrogate models. This could be achieved by using the results from sampling as an initial population for the surrogate optimization process. Also, the selection of different population sizes for this internal optimization of the surrogates could be made independent of the used sequential budget a.

With regard to detail changes, this could be to improve the used MARS model by using non default settings (e.g., with interactions, higher degree).

Real multivariate multiple regression analysis could be applicable to our problems, especially due to the noticed correlation of the objectives for the second case study. It should be researched if models that exploit such correlation are profitable.

While not of a main interest for our main research goals, it could be interesting to see if MSPOT can be applied to problems with larger budgets successfully. As observed in the experiment results MSPOT finds good solutions early on, then being outperformed by the MCO algorithms after several hundred function evaluations. This might give a hint how MSPOT could be a reasonable way of generating initial populations for other MCO algorithms.

If MSPOT itself was used for a large budget, limiting the number of design points for model building could be another interesting step. Since the time consumption for model building raises with the number of known points, MSPOT gets slower the longer it runs. The number of design points could be reduced with feature selection methods like PCA.

The main interest for our research would be to test MSPOT on real industrial problems like the cyclone application for filtering dust from power plant ovens. Other practical applications could include tuning of algorithm paramters for the detection of events in water quality data. This application can be considered multi objective, since there is always a trade off between false positive rate and true positve rate.

9. 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, Research Center CIOP (Computational Intelligence, Optimization and Data Mining), Cologne University of Applied Science, Faculty of Computer Science and Engineering Science, January 2011.

[2] T. Bartz-Beielstein, M. Friese, M. Zaefferer, B. Naujoks, O. Flasch, W. Konen, and P. Koch. Noisy optimization with sequential parameter optimization and optimal computational budget allocation. In *Proceedings of the 13th annual conference companion on Genetic and evolutionary computation*, GECCO '11, pages 119–120, New York, NY, USA, 2011. ACM.

[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. A. Coello Coello, D. A. Van Veldhuizen, and G. B. Lamont. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Springer, New York, 2nd edition, 2007.

[7] K. Deb. *Multi-Objective Optimization using Evolutionary Algorithms*. Wiley-Interscience Series in Systems and Optimization. Wiley, New York NY, 1 edition, 2001.

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

[9] 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.

[10] Y. Jin and B. Sendhoff. Trade-off between performance and robustness: an evolutionary multiobjective approach. In *Proceedings of the 2nd international conference on Evolutionary multi-criterion optimization*, EMO'03, pages 237–251, Berlin, Heidelberg, 2003. Springer-Verlag.

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

[12] J. J. Moré, B. S. Garbow, and K. E. Hillstrom. Testing unconstrained optimization software. *ACM Transactions on Mathematical Software*, 7(1):17–41, 1981.

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