Investigating Uncertainty Propagation in Surrogate-Assisted Evolutionary Algorithms

Vanessa Volz
TU Dortmund University
D-44221 Dortmund
vanessa.volz@udo.edu

Günter Rudolph
TU Dortmund University
D-44221 Dortmund
guenter.rudolph@udo.edu

Boris Naujoks
TH Köln - Univ. of Applied Sciences
D-51643 Gummersbach
boris.naujoks@th-koeln.de

ABSTRACT

Uncertainty propagation is a technique to incorporate individuals with uncertain fitness estimates in evolutionary algorithms. The Surrogate-Assisted Partial Order-Based Evolutionary Optimisation Algorithm (SAPEO) uses uncertainty propagation of fitness predictions from a Kriging model to reduce the number of function evaluations. The fitness predictions are ranked with partial orders and the corresponding individuals are only evaluated if they are indistinguishable otherwise or the risk of uncertainty propagation exceeds a steadily decreasing error tolerance threshold. In this paper, we investigate the effects of using uncertainty propagation according to SAPEO on single-objective problems. To this end, we present and apply different ways of measuring the deviations of SAPEO from the underlying CMA-ES. We benchmark the algorithms on the BBOB testbed to assess the effects of uncertainty propagation on their performance throughout the runtime of the algorithm on a variety of problems. Additionally, we examine thoroughly the differences per iteration between the evolution paths of SAPEO and CMA-ES based on a model for the rank-one update. The BBOB results suggest that the success of SAPEO generally improves the performance but depends heavily on function and dimension, which is supported by the analysis of the evolution paths.

CCS CONCEPTS

•Theory of computation → Approximation algorithms analysis; Evolutionary algorithms; •Computing methodologies → Genetic algorithms; *Uncertainty quantification;* Gaussian processes;

KEYWORDS

Surrogate-Assisted Optimisation, Evolutionary Optimisation, Performance Analysis

ACM Reference format:
Vanessa Volz, Günter Rudolph, and Boris Naujoks. 2017. Investigating Uncertainty Propagation in Surrogate-Assisted Evolutionary Algorithms. In *Proceedings of GECCO '17, Berlin, Germany, July 15-19, 2017,* 8 pages.
DOI: http://dx.doi.org/10.1145/3071178.3071249

GECCO '17, Berlin, Germany
© 2017 Copyright held by the owner/author(s). Publication rights licensed to ACM. 978-1-4503-4920-8/17/07…$15.00
DOI: http://dx.doi.org/10.1145/3071178.3071249

1 INTRODUCTION

Expensive optimisation problems are difficult for algorithms that rely on the exploration of the search space, such as evolutionary algorithms. Surrogate-assisted evolutionary algorithms [7–9] aim to make expensive problems tractable by replacing function evaluations of strategically selected individuals with estimates from a machine learning model trained on previously evaluated points.

The SAPEO algorithm[1] proposed recently for multi-objective optimisation problems [18] uses different dominance relations to rank individuals based on their estimated function value and applies the concept of uncertainty propagation (often found in algorithms designed for noisy problems [10]) to minimise the number of function evaluations needed. Under certain conditions, individuals are not evaluated and their fitness is only estimated via a surrogate model, thus introducing an uncertainty that is propagated throughout the algorithm. Because the next generation of an evolutionary algorithm usually only depends on the genetic representation of its parents and their relative fitness (i.e. ranks), uncertainty propagation should have no effect on the evolution path, provided that the ranking of individuals is still correct. We previously tested SAPEO on the bi-objective BBOB benchmarking suite. Although we received underwhelming results, we still found the SAPEO concept promising and thus are now investigating its performance in more detail and on single-objective problems for better observability.

In this paper, we first examine the results of a single-objective benchmark to identify patterns where different SAPEO versions deviate from the underlying CMA-ES. We base this investigation on observable differences in the anytime performance of the algorithms which also suggest differing evolution paths. We discuss the observed patterns using selected functions from the testbed as examples[2].

Based on these observations, we develop research questions that we address with an examination of the effects of uncertainty propagation on a single iteration of a CMA-ES with SAPEO. We analyse thoroughly the deviations of SAPEO from the CMA-ES evolution path (e.g. regarding population mean and covariance matrix) using a model for the rank-one update. We identify the possible and worst-case deviations and analyse the expected effects empirically. We specifically analyse the error frequency of some of the partial orders proposed in [18] under different conditions, such as dimension and certainty of the surrogate model. We also discuss how the results of this paper pertain to the multi-objective context

[1] Acknowledgements: The SAPEO concept was developed during the SAMCO Workshop in March 2016 at the Lorentz Center (Leiden, NL). https://www.lorentzcenter.nl This work is part of a project that has received funding from the European Unions Horizon 2020 research and innovation program under grant agreement No 692286.

[2] Complete benchmarking results available on http://url.tu-dortmund.de/volz

in previous work.

In the following section 2, we briefly describe the SAPEO algorithm and its adaptation to single-objective problems. We also discuss related work on the CMA-ES, uncertainty propagation and surrogate-assisted optimisation in general. In section 3 we present the BBOB benchmarking results and identify behaviour patterns of the SAPEO algorithm when compared to a regular CMA-ES. We analyse the different aspects of deviations per iteration in section 4 to answer the questions raised in the preceding section. We summarise and discuss the findings in section 5 and outline directions for future work. Without loss of generality we assume that all problems are minimisation problems throughout this paper.

2 BACKGROUND

2.1 SAPEO

The SAPEO (**S**urrogate-**A**ssisted **P**artial Order-Based **E**volutionary **O**ptimisation) algorithm (first introduced in [18]) is aimed at finding an optimal balance between the number of function evaluations and the uncertainty propagated throughout the runtime of the algorithm introduced by fitness estimates. In [18] it was applied to multi-objective optimisation problems only. However, with only some minor adaptations detailed in the following, SAPEO can be applied to single-objective optimisation problems.

SAPEO defines multiple strategies to combine a surrogate model with an evolutionary optimiser, regardless of the specific problem, model and optimisation algorithm used for an experiment. These strategies are based on different dominance relations that induce a partial order of the individuals in a population by incorporating information about the estimated fitness and the expected uncertainty of the prediction by a surrogate model. Some dominance relations akin to the ones used in SAPEO were previously applied in a similar context [12, 14].

Provided that a partial order already clearly identifies the μ best individuals, the evolutionary algorithm can continue with these individuals as parents for the next generation. If this is not the case, SAPEO either

(1) refines the partial order by ranking the individuals that can not be selected confidently with another dominance relation,
(2) uses approaches from multi-objective evolutionary algorithms to distinguish solutions (e.g. hypervolume) or
(3) evaluates additional individuals until the μ best can be identified confidently.

This way, SAPEO has the potential to reduce the number of function evaluations of an evolutionary algorithm, while only allowing a controllable risk (via an uncertainty threshold ε and the selected partial orders). For more detailed information, see the algorithm description in [18].

In this study, we will be using a Kriging model and a CMA-ES, the state-of-the-art algorithm for single-objective problems. The only modification made to SAPEO for the application to single-objective optimisation problems is regarding the adaptation strategy of the uncertainty threshold ε. In the single-objective version, the threshold at generation $g+1$, ε_{g+1} is the minimum of ε_g and the standard deviation of the function values of the population at generation g. This is to reduce the number of indistinguishable individuals with overlapping confidence intervals.

The formal description of SAPEO can be simplified in the single-objective case as follows: Let $\tilde{f}(X_i) \in \mathbb{R}$ be the predicted fitness for individual X_i as computed by a local surrogate model with uncertainty $\tilde{\sigma_i}$ modelled by

$$\tilde{f}(X_i) = f(X_i) + e_i, \quad e_i \sim \mathcal{N}(0, \tilde{\sigma_i}) \text{ and,}$$
$$u_i = \tilde{\sigma}_i z\left(1 - \frac{\alpha}{2}\right).$$

Here, z denotes the quantile function of the standard normal distribution and α the confidence level. Thus, as per definition, $f(X_i)$ lies within the interval $[\tilde{f}(X_i) - u_i, \tilde{f}(X_i) + u_i]$ with probability $(1-\alpha)$ provided that $\tilde{\sigma_i}$ is estimated correctly (assumption **A1**).

We list the dominance relations and total orders used in this paper below. The relations assume that the actual function values lie in the confidence interval (**A2**). The definitions are adapted for the single-objective case from [18], for details on their justification and assumptions please refer to the original paper.

$<_f$: *Total order on function values.*

$$X_i <_f X_j := f(X_i) < f(X_j)$$

$<_p$: *Total order on predicted values.*

$$X_i <_p X_j := \tilde{f}(X_i) < \tilde{f}(X_j)$$

$\preceq_u$: *Confidence interval dominance.* (cf. [12, 14])

$$X_i \preceq_u X_j := \tilde{f}(X_i) + u_i < \tilde{f}(X_j) - u_j$$

$\preceq_c$: *Confidence interval bounds as objectives.*

$$X_i \preceq_c X_j := \begin{pmatrix} \tilde{f}(X_i) - u_i \\ \tilde{f}(X_i) + u_i \end{pmatrix} \preceq \begin{pmatrix} \tilde{f}(X_j) - u_j \\ \tilde{f}(X_j) + u_j \end{pmatrix},$$

with $a \preceq b \iff \forall k \in \{1 \dots d\} : a_k \leq b_k \wedge \exists k \in \{1 \dots d\} : a_k < b_k$ for $a, b \in \mathbb{R}^d$ akin to the commonly known Pareto dominance relation.

2.2 CMA-ES

The **C**ovariance **M**atrix **A**daptation **E**volution **S**trategy (CMA-ES) is a very popular optimisation algorithm that uses an iterative covariance matrix estimate to guide mutations into the direction of the negative gradient of the given problem. Many variants of the algorithm exist, we use [2] in this paper, along with its python implementation[3]. Moreover, we use a simplified model of a SAPEO iteration based on a rank-one update of the CMA-ES according to [1] in order to investigate the effects of uncertainty propagation in more detail. Throughout the paper, we will use parameters according to table 1. With the rank-one update, the mean m and covariance matrix C are adapted in a single iteration as follows.

$$x_i = m + \sigma y_i, \quad y_i \sim N_i(0, C)$$
$$m = m + \sigma y_w \quad \text{where } y_w = \sum_{i=1}^{\mu} w_i y_{i:\lambda}$$
$$C = (1 - ccov)C + ccov\ \mu_w\ y_w y_w^T \quad \text{where } \mu_w = \frac{1}{\sum_{i=1}^{\mu} w_i^2} \geq 1$$

[3] https://pypi.python.org/pypi/cma, version 1.1.7

symbol	value	explanation
λ	$4 + \lfloor(3\log n\rfloor)$	number of offspring [5]
μ	$\lfloor\frac{\lambda}{2}\rfloor$	number of parents[5]
w_i	$\frac{1}{\mu}$	individual weights
$ccov$	$\frac{2}{n^2}$	covariance matrix update weight[1]
σ	1	step size[1]

Table 1: CMA-ES parametrisation

With the propagation of uncertainty, the CMA-ES in our experiments will essentially be operating on a noisy problem. The most prominent representatives for such CMA-ES on noisy fitness functions are IPOP-CMA-ES and BIPOP-CMA-ES, where IPOP (**I**ncreasing **POP**ulation-size) and BIPOP (**BI**-**POP**ulation) incorporate different restart strategies and stopping criteria. These CMA-ES variants were compared on the BBOB noisy testbed by Ros [13] and found to perform similarly.

IPOP and BIPOP concepts are also incorporated in the most successful algorithms coupling CMA-ES with surrogate models. Loshchilov et al. [11] proposed them as extensions of ^{s*}ACM-ES, an earlier variant of a surrogate-assisted CMA-ES. ^{s*}ACM-ES as well as its variant and predecessors apply approximate ranking by Runarsson [15] to distinguish individuals in a noisy environment instead of the partial-order based ranking in SAPEO. In this paper, we will not be using these concepts as to not dilute the interpretability of the results, although they offer potential for further improvements.

2.3 Uncertainty Propagation

Surrogate-assisted optimisation algorithms tend to avoid the propagation of uncertainty in order to not run the risk of cumulative errors (see surveys in [7–9]). Instead, the featured algorithms often evaluate all individuals that are selected into the parent population. The selection is often based on an aggregated metric such as the expected fitness value given an assumption for the distribution of the error. An exception is the work in [12], where confidence intervals are used to select individuals without further evaluations for a differential evolution algorithm (DEMO).

For noisy optimisation problems, however, it is not possible to remove the uncertainty from the fitness evaluations. Although the expected error can be reduced through repeated evaluations, it can never be removed completely. One approach to dealing with this problem is to increase the robustness of the optimisation (see IPOP and BIPOP as described in section 2.2). There are also approaches that use uncertainty-based ranking on dominance relations such as [14] (c.f. $\preceq_c$ used in this paper and described in section 2.1) and [6, 17]. Uncertainty propagation for evolutionary algorithms (especially in multi-objective contexts) are discussed in [10].

3 UNCERTAINTY PROPAGATION ON BENCHMARK SUITE

For our experiments on anytime performance, we run different SAPEO variants as well as a regular CMA-ES on the BBOB single-objective benchmarking testbed (24 function, 15 instances). We test the algorithms on search spaces of 2,3,5,10 and 20 dimensions on each problem with a budget of 10^3 per dimension n. The experiment runs in batch mode with 360 jobs on a parallel computing cluster, most of which took around 30 minutes to complete and all finished within 5 hours. An increase of computation time with higher sample sizes for the surrogate model could be observed but the runtime of the experiments is still practicable.

We test selected SAPEO variants (strategies *uf*, *ucp*, *up* and surrogate sample sizes $\lambda, 2\lambda$) against the regular CMA-ES. We use different sample sizes to analyse the effect of a reduction of the uncertainty in the surrogate model. We choose strategies that differ in terms of their approach; *uf* only relies on the confidence intervals dominance relation and evaluates all remaining uncertain individuals while *ucp* postpones evaluation as much as possible by applying three increasingly risky dominance relations. *up* behaves like *ucp* in cases where the confidence interval widths of a population are approximately equal. The results of four functions representative for some patterns we observed are visualised in figure 1 generated with the BBOB postprocessing tools[4].

Figure 1a shows the empirical cumulative distribution function for all algorithms on four functions (dimension 2) with different characteristics (legend in caption). The graphs show the percentage of precision targets[5] reached on the corresponding function at each number of function evaluations (in log-scale). In our experiments, all algorithms only have a budget of up to $10^3 \cdot n$ function evaluations which is represented by the large crosses at $x = 3$. The remaining values are then bootstrapped as described in [3].

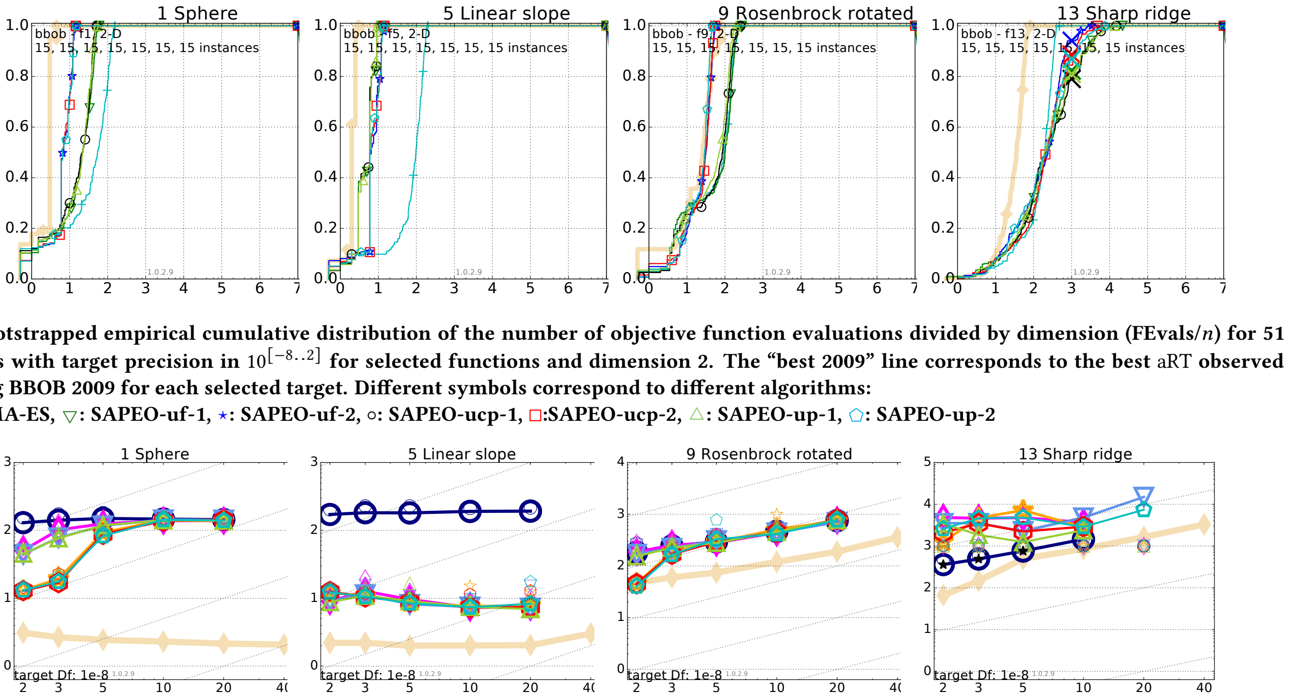
Generally, we can observe from the benchmark that on most functions in dimension 2, SAPEO versions with a larger sample size perform better than the remaining algorithms starting after about 10^1 function evaluations. This is true for the sphere and Rosenbrock functions in this dimension as pictured. Although there are some functions where all SAPEO variants clearly improve the performance of the CMA-ES (e.g., sphere and linear slope), for the majority of functions there are only small differences throughout the measured runtimes (c.f. sharp ridge until 10^2 function evaluations). Sometimes, there are small areas where one algorithm outperforms the others as is the case for the CMA-ES on the sharp ridge function after $10^2 \cdot n$ evaluations as pictured. All SAPEO variants generally perform at least as well as the CMA-ES at any time in dimension 2, but the sharp ridge function proves that there are exceptions to this rule.

On dimension 2, SAPEO generally has an edge over the CMA-ES, but surrogate models increase in uncertainty with the number of dimensions in search space. The behaviour of the algorithms on the previously discussed functions in larger dimensions is visualised in figure 1b. These figures show the average runtime of each algorithm to reach the highest target of the respective function. The aRT values represent the expected average runtime of a restart algorithm. They are bootstrapped from 1 000 trials according to [3] and can therefore reach values higher than the allocated budget.

We generally observe quadratic scaling of the performance with dimension. The linear slope and sphere are exceptions with only linear scaling. The SAPEO variants improve the performance of the

[4] see https://github.com/numbbo/coco

[5] In the BBOB framework, there are 51 targets for precision between 10^2 and 10^{-8}.



(a) Bootstrapped empirical cumulative distribution of the number of objective function evaluations divided by dimension (FEvals/n) for 51 targets with target precision in $10^{[-8..2]}$ for selected functions and dimension 2. The "best 2009" line corresponds to the best aRT observed during BBOB 2009 for each selected target. Different symbols correspond to different algorithms:
+: CMA-ES, ▽: SAPEO-uf-1, ⋆: SAPEO-uf-2, ∘: SAPEO-ucp-1, □:SAPEO-ucp-2, △: SAPEO-up-1, ⬠: SAPEO-up-2

(b) Average running time (aRT in number of f-evaluations as $\log_{10}$ value), divided by dimension for target function value 10^{-8} versus dimension. Slanted grid lines indicate quadratic scaling with the dimension. Light symbols give the maximum number of function evaluations from the longest trial divided by dimension. Black stars indicate a statistically better result compared to all other algorithms with $p < 0.01$ and Bonferroni correction number of dimensions. Different symbols correspond to different algorithms:
∘: CMA-ES, ◇: SAPEO-uf-1, ⋆: SAPEO-uf-2, ▽: SAPEO-ucp-1, ⬡:SAPEO-ucp-2, △: SAPEO-up-1, ⬠: SAPEO-up-2

Figure 1: ECDFs for dimension 2 (1a) and aRT plots (1b) for selected BBOB results

CMA-ES exceptionally well on the linear slope, with even the maximum runtime of the SAPEO variants much lower than the CMA-ES averages. However, as expected, the SAPEO variants generally do not scale as well as the CMA-ES with increasing dimension.

As is the case for the sphere and Rosenbrock functions, some SAPEO variants (mostly with larger sample size) beat the CMA-ES in smaller dimensions but only perform as well as the CMA-ES for higher dimensions. This can be explained by the fact that SAPEO will behave like the CMA-ES if the certainty of its ranking is below a certain threshold, which is more often the case in higher dimensions. However, there are also some cases where the CMA-ES outperforms SAPEO (cf. sharp ridge function). This mostly happens on functions with narrow valleys or ridges that contain the optimum as uncertainties are punished harsher. Other examples are functions that are not approximated well with the surrogate model and selected kernel.

There are multiple functions in the benchmark (mainly multimodal functions with adequate or weak global structure) where none of the algorithms manages to reach the highest target. One possible explanation is a general weakness of the CMA-ES for this type of problem. There are still some cases where the surrogate model enables the CMA-ES to reach the highest target (cf. sharp ridge function, dimension 20).

There are some rare cases where one or more of the algorithms perform better in terms of aRT value than the best2009 comparison, but mostly, the tested algorithm do not reach the best2009 performance or only do so for smaller target values. However, the intention of this benchmark was to investigate the concept of uncertainty propagation and how much it can improve an evolutionary algorithm (in this case the CMA-ES). With more tuning, restarts and especially an even larger sample size, we suspect that the algorithms could compete with the best performances.

From the benchmark, we can conclude that uncertainty propagation generally has a notable and mostly positive effect on the anytime performance of the CMA-ES. However, it is not possible to assess the changes in terms of the evolution path and further details from the performance data. Based on our observations on the benchmark we identify the following questions for further analysis.

Q1 What is the worst-case deviation from the evolution path?
Q2 How often does the offspring selection differ and what is the effect on the evolution path?
Q3 What is the cause for the scaling behaviour?
Q4 What is the cause of the function-specific performance?
  Q4a How does the certainty of the surrogate model affect the performance? Can performance problems in SAPEO on some functions be explained by the model?
  Q4b Are there any problems with the SAPEO assumptions (see section 2.1 and [18])?

4 UNCERTAINTY PROPAGATION IN A SINGLE ITERATION

In the following, we investigate the effects of uncertainty propagation on a single iteration under near-optimal conditions in order to answer the questions posed at the end of the previous section. To this end, we analyse the extent of possible errors in section 4.1. Additionally, in section 4.2, we simulate the prediction error of a surrogate model to investigate the likelihood of errors under different ranking strategies as well as the expected effect per iteration.

By default, the CMA-ES applies multinormally distributed mutations. Here, we replace the multinormal distribution by a uniform distribution on the surface of a hyperellipsoid. This choice is rooted in the fact that a uniform distribution on a hypersphere surface converges weakly to a multinormal distribution as $n \to \infty$ [16, p. 222f.] and is intended to eliminate possible misorderings caused by the random lengths of multinormal random vectors for this experiment.

For the initial population, we thus distribute the individuals equidistantly on an n-sphere with radius $r = 1$ and centre in the origin (not equal to the optimum). For problems with search space dimension $n = 2$, we obtain the initial population by placing the individuals on a circle with an angle of $\frac{360}{\lambda}$ deg between each of them. Achieving an equidistant (deterministic) distribution of λ points on the surface of an n-dimensional hypersphere analytically, however, is computationally laborious for large n.

For $n > 2$ we therefore apply an approximative solution: We first generate λ individuals uniformly on the n-sphere by drawing λ independent standard multinormally distributed random vectors and divide each of them by its realized length. Next, we maximise the sum of the Euclidean distances between each individual and its nearest neighbour. We take the best sampling obtained from a CMA-ES with 10 restarts, 1 000 iterations per run and standard parametrisation[6]. We keep the sample for the following experiments constant to avoid noise introduced by the sampling. The setup of this population automatically defines the initial covariance matrix $C = I_n$ where I_n is the n-dimensional unit matrix.

In the following experiments, we will be using the sphere model as a fitness function (cf. [4]) as it is the most clearly observable one out of the fitness functions in the SAPEO benchmark. Moreover, the SAPEO algorithms show interesting behaviour patterns on the sphere function in our experiment in section 3 that we intend to investigate further. Since the population in this experiment is generated around the origin, we set the minimum $x_{opt} = \arg\min_{x \in \mathbb{R}^n} f(x) = 1 \in \mathbb{R}^n$ and $f(x_{opt}) = 0$. This means that the optimum lies outside of the unit n-sphere described by the population and that, ideally, the mean of the offspring generation should observably move towards the optimum. To model a CMA-ES iteration, we use the rank-one update as described in section 2.2 with parameters according to table 1.

4.1 Possible and Worst-Case Effects of Uncertainty Propagation

We define a selection error as the number of individuals that are incorrectly selected into the parent population. More formally, let O be the offspring population $(x_{(i)}, i \in \{1, \ldots, \lambda\})$ sorted according to the individuals' function values in increasing order. Then, the correct parent population P^* can be expressed as:

$$P^* = \{x_{(i)} | i \le \mu\}$$

For another parent population P', the selection error then is the cardinality of the set-theoretic difference of P' and P^*, i.e.

$$e(P') = |P' \setminus P^*|$$

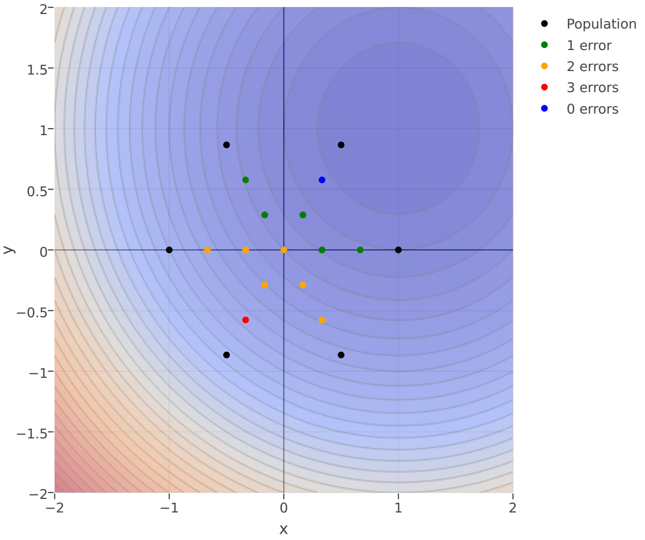


Figure 2: Population means of possible parent selections from the previous generation's offspring (black dots), colour coded according to the number of selection errors. The contour plot in the background visualises the 2-dimensional sphere function with an optimum at $1 \in \mathbb{R}^2$.

For the experiments, we list all possible parent populations P' with at least one selection error. As per the definition of e we can have a maximum of $\min(\mu, \lambda - \mu)$ selection errors. Thus, let $i \in \{1, .., \min(\mu, \lambda - \mu)\}$ be the number of selection errors. We then have $\binom{\mu}{i}$ possibilities to draw individuals from P^* to replace them with $\binom{\lambda-\mu}{i}$ possible combinations of incorrectly selected individuals. For dimension $n = 2$ and parameters according to table 1 ($\lambda = 6$, $\mu = 3$), this results in 19 erroneous parent populations. Of these populations, 9 have one selection error, another 9 have two and the remaining one has 3.

The original population as generated per the model described in the previous section is visualised in figure 2 as black dots, along with the means of the different parent populations. Green dots represent the means of selections with 1 error, yellow ones with 2. The red dot is the mean of the population without a correct individual and the blue one is the actual mean without errors. The background of the figure shows a contour plot of the fitness function with a colour scale from blue (small) to red (high). The optimum is visible at $1 \in \mathbb{R}^2$ in the dark blue region.

The new population mean in our experiment solely depends on the number and type of selection errors, the fitness values of the individuals do not affect it. We will therefore systematically analyse all possible errors and their effects on the next iteration. Some of

[6] R package cmaes, parameters according to table 1

the means overlap each other and there is a green as well as an orange point on the origin (cf. figure 2).

For this experiment, i.e. for the fitness function and original population in this case, all of the selections with 1 error resulted in a mean with a better fitness value than the previous one. This is not at all true for cases with more selection errors. However, the evolution path could take a considerable detour even with a single error since it would also be guided by a different covariance matrix that could unnecessarily spread orthogonally to the gradient. We therefore collected additional metrics to measure the effect of selection errors more accurately in terms of their deviation from the correct selection. In table 2, we list the maximum of selected metrics in order to express the worst-case effects in relation to the error count, answering research question **Q1** posed in section 3.

	# selection errors			
symbol	1	2	3	measure
δm	0.67	1.15	1.33	Euclidean dist. between means
$\delta\alpha_0$	60	60	0	abs. main axis angle [degrees]
δk	0.67	1	1.33	difference of condition numbers

Table 2: Maximum deviation (rounded) per number of errors

In the table, δm is the Euclidean distance between the correct new population mean and the one resulting from the erroneous parent population. As is already clear from figure 2, a single error only causes a smaller setback in terms of the population mean whereas a completely wrong selection moves the mean into the opposite direction. Additionally, due to the symmetric setup of the experiment, the population with 3 errors results in a covariance matrix with its main axis pointing exactly in the opposite direction as the intended one and thus an angle of 0 between the main axes. Therefore, its succeeding generation still has a chance of containing individuals with better function values. δk expresses the difference between the actual conditioning number and the conditioning number of the covariance matrix resulting from an erroneous update. This means that the ellipsoid is wider and the algorithm explores more. This is beneficial behaviour especially for the populations with 1 or 2 errors in order to counteract the incorrect angle of the main axis.

From this experiments, we observe that the CMA-ES is fairly robust against all possible errors. A small number of errors will not cause a large deviation from the evolution path but the optimisation process will be slowed down nevertheless since the algorithm has to counteract the error with more exploration. However, even if there is a large number of errors, the algorithm should be able to recover fast as long as these errors do not occur in each generation. This is due to the symmetric nature of the distribution used to generate the offspring population, that will also create offspring not in negative gradient direction which can counteract these errors. In order to assess the risk of propagating uncertainty, we will assess the number of expected errors in the next subsection.

4.2 Expected Effects of Uncertainty Propagation

After having observed all possible errors and their effects in the previous section, we study the same issues empirically in this section to answer the remaining research questions from section 3. This analysis aims at estimating the expected effects of uncertainty propagation in a single iteration under the conditions detailed in the experiment description above. For the following experiments, we therefore need to artificially insert noise into the system, which, in an actual application, would be produced by the Kriging model. If the model assumptions hold, the actual fitness value of an individual i is distributed according to a normal distribution with the predicted value as mean and a variance of $\tilde{\sigma_i}$ that the model estimates as well.

The allowed uncertainty $\tilde{\sigma_i}$ is limited to the standard deviation of the function values of a population through the decreasing error tolerance threshold ε, which is in turn limited by the standard deviation of the previous population. The standard deviation of the population visualised in figure 2 is $\varepsilon \approx 2.19$. We test noise levels σ_u of $0.1, 0.5, 1, 1.5, 2$. In our experiments, we use the same noise level for all individuals. We thus simulate the prediction error by drawing random numbers distributed according to the normal distribution described above to replace the correct fitness values. We compare the rankings according to the SAPEO variants used in section 3 with the correct ranking and evolution path according to the actual function values using metrics detailed in table 3. SAPEO-ucp and SAPEO-up would always make the same selection decisions since all confidence intervals in this setup have the same width. The results for SAPEO-up in the following therefore hold for SAPEO-uup as well. The results for SAPEO-uf are analysed via the observations for SAPEO-u, which describe the frequency of additional function evaluations in SAPEO-uf.

measure	explanation
Critical Individuals	Number of individuals on the same rank as the μ-best one divided by μ
CI Errors	Number of individuals outside the confidence intervals divided by λ
Selection Errors	Number of selection errors (see section 4.1) divided by μ
Distance of means	Distance between the means (cf. table 2)
Difference min EV	Difference between the minimal Eigenvalues of the covariance matrices
Absolute Main Axis Angle	Angle between the main axes of the distribution (cf. table 2)
Difference of condition no.	Difference between condition numbers of the covariance matrices (cf. table 2)
Difference max EV	Difference between the maximal Eigenvalues of the covariance matrices

Table 3: Measures used to express deviation from the evolution path

We run the experiments for dimension 2 to compare it with the results from the previous subsection. In order to investigate the influence of search space dimension n, we run a second set of experiments with $n = 10$.

The test is executed 10 000 times for each SAPEO variant, noise level and dimension. Each of these runs takes about 3 hours on a regular desktop PC. The metrics captured from all tests are visualised in boxplots in figure 3. Each of the tests is labelled on the x-axis with its noise level σ_u (top) and dimension n (bottom). The

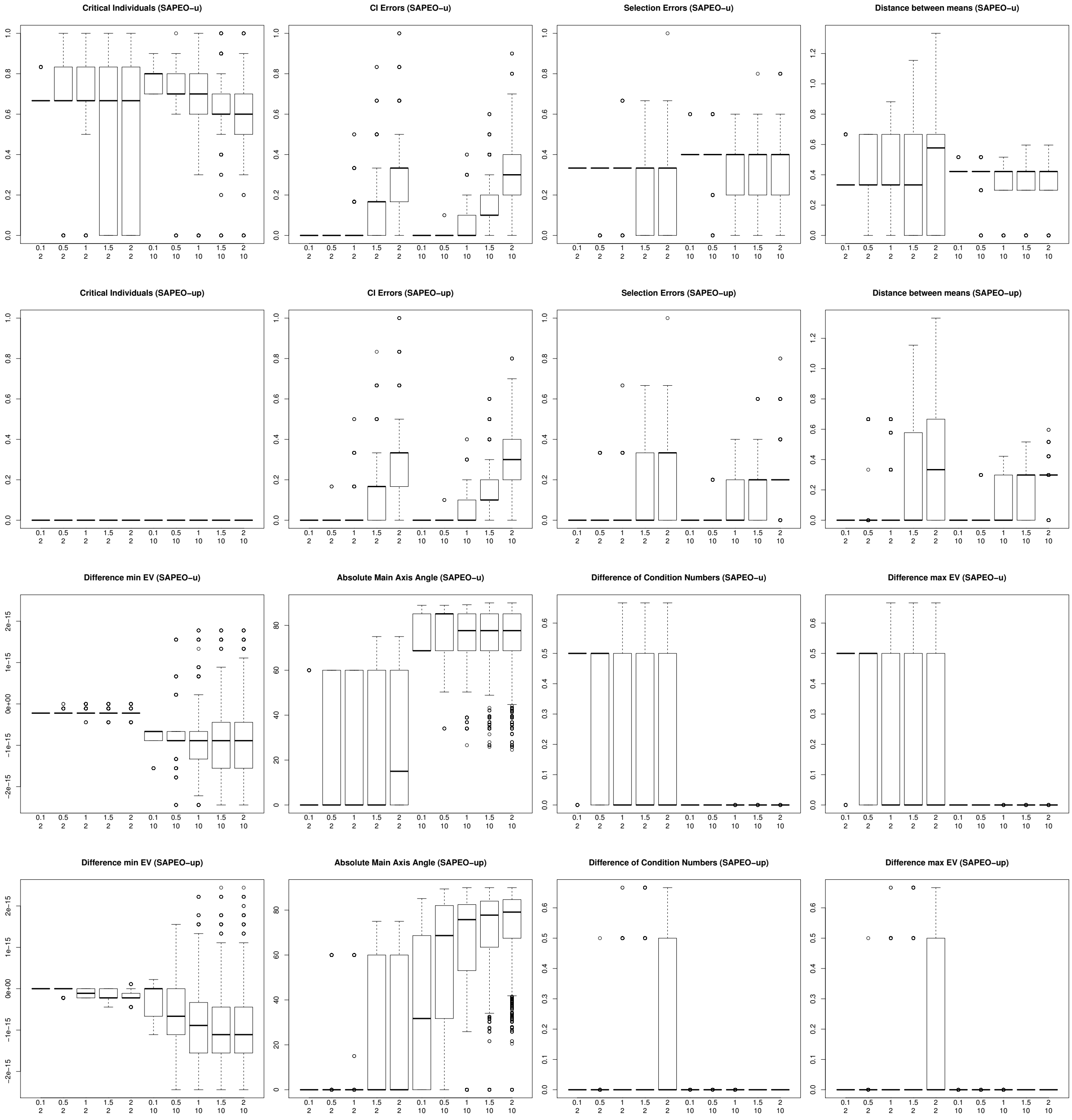


Figure 3: Empirical deviations from the evolution path according to metrics in table 3 for SAPEO-u and SAPEO-up/SAPEO-ucp for 10 000 tests with noise levels σ_u between 0.1 and 2 and for dimensions n of 2 and 10

results for the different SAPEO variants are in separate figures.

Based on the plots, we investigate the remaining research questions. The number of critical individuals is always 0 for SAPEO-up since it always introduces a total order by choosing the predicted fitness value of the critical individuals. How often this happens is visible in the corresponding plot for SAPEO-u. The median number of critical individuals seems to always be quite high for all noise levels and dimension with around 0.6μ. SAPEO-uf would evaluate all individuals that are critical, thus saving only about 30% percent of function evaluations when compared to the regular CMA-ES. The dimension of the search space seems to only be a minor influence on the ranking behaviour [**Q3**].

The boxplot on CI Errors gives some insight on research question **Q4a+b** showing that assumption **A2** in section 2.1 can be violated, especially for higher noise levels and higher search space dimension. These results might justify lowering the uncertainty threshold ε depending on dimensionality in order to reduce this type of error, since they might cause problems in the scaling behaviour of the algorithm [**Q3**]. The number of selection errors (see **Q2**) increases with noise and dimension, which of course could be partially influenced by the number of CI errors as described above. SAPEO-up/SAPEO-ucp seems to work exceptionally well for lower noise levels with almost no errors in all tests. The selection errors in SAPEO-u are caused due to the random selection of parents from the critical individuals. Their effect on the evolution path is investigated with the remaining plots.

The distance between the means obviously correlates with the number of selection errors (as was already suggested from the findings in section 4.1). However, considering the much higher amount of selection errors for SAPEO-u, the distance of the means is still relatively low, suggesting robust behaviour of the algorithm. We see a significantly declining performance with increasing noise level [**Q4a**]. The difference of the minimal Eigenvalue and the angle between the main axes are used to describe the main direction of exploitation starting from the new mean. The difference of the Eigenvalues and the resulting angles are consistently higher for dimension 10 which is explained by the higher possibility of error in more dimensions [**Q3**]. However, we also see that the angle can be significantly different from the correct one. We therefore also investigate the exploration behaviour expressed through the condition number and the minimal EV of the covariance matrix visualised in the remaining plots. In section 4.1 we observed more exploration (a higher difference of condition numbers) with more selection errors. This seems to be reflected directly only for dimension 2. However, this type of exploration is not fully grasped with the condition number for higher dimensions. A study of the population generated from the new mean and covariance matrix might help to investigate this issue in the future.

5 CONCLUSION AND FUTURE WORK

In this paper, we investigate the effects of uncertainty propagation on surrogate-assisted evolutionary algorithms in multiple ways using the SAPEO algorithm as an example. A comparison of the anytime performance of SAPEO with the underlying CMA-ES shows a general improvement through the surrogate-assisted approach. However, the amount of improvement varies greatly across functions and decreases with dimension. We identify multiple questions to investigate these issues further.

We address these questions by examining the deviations from SAPEO and CMA-ES on a single iteration in detail using a simplified model on the sphere function. We find that the deviations can be very high if the parent selections differ greatly. At the same time, the exploratory behaviour of the CMA-ES is able to counteract a small amount of errors. Selection errors still can have a measurable impact on the mean and main search direction for the new population. The quality of the surrogate model has a high impact on the satisfaction of some of the assumptions of SAPEO as it influences the algorithm in multiple ways. Dimensionality has a small impact on the ranking strategies, but high dimensions are especially problematic for the identification of the main search direction.

In the future, we intend to incorporate the findings into SAPEO by adapting the error tolerance to minimise the violations of the algorithm's assumptions. Using restart strategies and larger populations as well as larger and/or optimised samples for the surrogate model can potentially improve the performance of SAPEO further. We plan to compare SAPEO with $^{s*}ACM-ES$ in the future and investigate theoretical performance guarantees of SAPEO.

REFERENCES

[1] Y. Akimoto, A. Auger, and N. Hansen. 2015. Continuous Optimization and CMA-ES. In *Companion of Genetic and Evolutionary Computation Conference (GECCO)*. ACM, New York, NY, 313–344.

[2] N. Hansen. 2006. The CMA Evolution Strategy: A Comparing Review. *Studies in Fuzziness and Soft Computing* 192 (2006), 75–102.

[3] N. Hansen, A. Auger, O. Mersmann, T. Tušar, and D. Brockhoff. 2016. COCO: A Platform for Comparing Continuous Optimizers in a Black-Box Setting. *CoRR* abs/1603.08785 (2016). http://arxiv.org/abs/1603.08785

[4] N. Hansen, S. Finck, R. Ros, and A. Auger. 2010. *Real-Parameter Black-Box Optimization Benchmarking 2010: Noiseless Functions Definitions*. Technical Report RR-6829. INRIA Saclay-Île-de-France, Paris, France.

[5] N. Hansen and A. Ostermeier. 2001. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation* 9, 2 (2001), 159–195.

[6] E.J. Hughes. 2001. Evolutionary Multi-objective Ranking with Uncertainty and Noise. In *Evolutionary Multi-Criterion Optimization (EMO)*. Springer, Berlin, 329–243.

[7] Y. Jin. 2005. A Comprehensive Survey of Fitness Approximation in Evolutionary Computation. *Soft Computing* 9, 1 (2005), 3–12.

[8] Y. Jin. 2011. Surrogate-assisted evolutionary computation: Recent advances and future challenges. *Swarm and Evolutionary Computation* 1, 2 (2011), 61–70.

[9] J. Knowles and H. Nakayama. 2008. Meta-Modeling in Multiobjective Optimization. In *Multiobjective Optimization - Interactive and Evolutionary Approaches*, J. Branke and others (Eds.). Springer, Berlin, 245–284.

[10] P. Limbourg and D. E. Salazar Aponte. 2005. An Optimization Algorithm for Imprecise Multi-Objective Problem Functions. In *IEEE Congress on Evolutionary Computation (CEC 2005)*. IEEE Press, Piscataway, NJ, 459–466.

[11] I. Loshchilov, M. Schoenauer, and M. Sebag. 2013. Intensive Surrogate Model Exploitation in Self-adaptive Surrogate-assisted CMA-ES (saACM-ES). In *Genetic and Evolutionary Computation Conference (GECCO 2013)*. ACM Press, New York, pp.439–446.

[12] M. Mlakar, D. Petelin, T. Tušar, and B. Filipič. 2015. GP-DEMO: Differential Evolution for Multiobjective Optimization based on Gaussian Process models. *European Journal of Operational Research* 243, 2 (2015), 347–361.

[13] R. Ros. 2010. Black-box optimization benchmarking the IPOP-CMA-ES on the noisy testbed: comparison to the BIPOP-CMA-ES. In *Companion of Genetic and Evolutionary Computation Conference (GECCO 2010)*. ACM Press, New York, 1511–1518.

[14] G. Rudolph. 2001. A Partial Order Approach to Noisy Fitness Functions. In *IEEE Congress on Evolutionary Computation (CEC 2001)*. IEEE Press, Piscataway, NJ, 318–325.

[15] T. Runarsson. 2004. Constrained Evolutionary Optimization by Approximate Ranking and Surrogate Models. In *Parallel Problem Solving from Nature (PPSN VIII)*. Springer, Berlin, 401–410.

[16] A. J. Stam. 1982. Limit theorems for uniform distributions on spheres in high-dimensional Euclidean spaces. *Journal of Applied probability* 19, 1 (1982), 221–228.

[17] J. Teich. 2001. Pareto-Front Exploration with Uncertain Objectives. In *Evolutionary Multi-Criterion Optimization (EMO)*. Springer, Berlin, 314–328.

[18] V. Volz, G. Rudolph, and B. Naujoks. 2017. Surrogate-Assisted Partial Order-Based Evolutionary Optimisation. In *Evolutionary Multi-Criterion Optimization (EMO)*. Springer, Berlin, 639–653.