When Parameter Tuning Actually is Parameter Control

Simon Wessing Mike Preuss Günter Rudolph

Computational Intelligence Group, Chair of Algorithm Engineering
Technische Universität Dortmund, Germany
{simon.wessing,mike.preuss,guenter.rudolph}@tu-dortmund.de

ABSTRACT

In this paper, we show that sequential parameter optimization (SPO), a method that was designed for (offline) parameter tuning, can be successfully used as a controller for multistart approaches of evolutionary algorithms (EA). We demonstrate this by replacing the restart heuristic of the IPOP-CMA-ES with the SPO algorithm. Experiments on the BBOB 2010 test cases suggest that the performance is at least competitive while the approach provides more options, e.g. setting more than one parameter at once. Essentially, we argue that SPO is a generalization of the IPOP heuristic and that the distinction between tuning and control is—although often useful—an artificial one.

Categories and Subject Descriptors

G.1.6 [**Numerical Analysis**]: Optimization—*global optimization, unconstrained optimization*; F.2.1 [**Analysis of Algorithms and Problem Complexity**]: Numerical Algorithms and Problems

General Terms

Performance, Experimentation

Keywords

parameter setting, parameter tuning, parameter control, SPO, CMA-ES

1. INTRODUCTION

A straightforward approach to improve optimization performance on multimodal problems are multistart strategies. These approaches work by splitting up the available budget of objective function evaluations into several independent runs of the optimization algorithm. Their advantage lies in the randomly chosen starting points that greatly influence in which region the optimization concentrates. A special case of multistart approaches are restart approaches, which usually contain heuristics to detect e. g. the algorithm's stagnation in a local optimum. It is then possible to trigger a restart with a new, random initial solution and modified strategy parameters. The variation of strategy parameters may follow simple heuristics, e.g. doubling the population size in order to better cope with the highly multimodal but regular global structure of the problem, as done in the covariance matrix adaption evolution strategy with increasing population size (IPOP-CMA-ES) by Auger and Hansen [1]. It may or may not take feedback from the previous starts into account. An example for an (although very indirect) feedback based mechanism is the somewhat more complex heuristic employed in the BI-population (BIPOP) CMA-ES by Hansen [6]. It uses IPOP and another heuristic interleaved, depending on which one has consumed a smaller budget of function evaluations. However, it may be argued that these mechanisms have been designed for tackling the special cases of some otherwise unsolvable problems and would have to be adapted to cope well with other difficult problems. Effectively, one is trying to solve a low dimensional (1 in this case) meta-optimization or parameter setting problem here which could be handled in a very different way.

The currently predominant nomenclature divides parameter setting into parameter tuning and parameter control [3]. In this context, control means the dynamic setting of parameters during the run, while tuning means parameter setting in a separate stage before the actual optimization. The latter one is often criticized as extremely time-consuming [3, 14], because it implies several complete runs of the optimizer to test different configurations. However, multistart approaches provide us with the same simple interface and are actually similarly expensive if the number of restarts is high enough.

As our approach utilizes restarts which are performed after somehow recognizing that the search becomes unproductive, it may make sense to ask for criteria that provide such information. Most current EA employ movement criteria for this purpose as already proposed by Schwefel [20]. This concept is e. g. applied by Sastry [19] or Zielinski and Laur [23]. Qualified run-time distributions by Hoos and Stützle [10] may be an alternative but have only been investigated for combinatorial/discrete optimization. We also disregard theoretical concepts here as they rather have a global view on the search space that is inappropriate if restarts shall be done rapidly.

The CMA-ES in its different variants employs the best known developed set of movement criteria relating to pop-

GECCO'11, July 12–16, 2011, Dublin, Ireland.

Algorithm 1 Sequential Parameter Optimization

Input:
$\mathbf{l}, \mathbf{u}$ // box constraints
N_{init} // size of the initial design set
$\mathbf{D} \leftarrow \text{lhs}(\mathbf{l}, \mathbf{u}, N_{\text{init}})$ // generate initial design
$\mathbf{Y} \leftarrow \text{runDesign}(\mathbf{D})$ // perform experiments
while budget not exhausted **do**
  // calculate performance indices
  $\mathbf{y} \leftarrow \text{aggregateRuns}(\mathbf{Y})$
  // fit empirical model of the response
  $\mathcal{M} \leftarrow \text{fitModel}(\mathbf{D}, \mathbf{y})$
  // find promising design point
  $\mathbf{d}_{\text{new}} \leftarrow \text{modelOptimization}(\mathcal{M})$
  // perform experiments and add results
  $\mathbf{Y} \leftarrow \mathbf{Y} \cup \text{runDesign}(\mathbf{d}_{\text{new}})$
  $\mathbf{D} \leftarrow \mathbf{D} \cup \mathbf{d}_{\text{new}}$
end while
return $\mathcal{M}$, $\mathbf{d}^*$ // return final model and best design

ulation variance, step sizes and numerical issues, mostly tasked at detecting situations where no further movement of the population can be expected [9, 5]. This is another motivation to rely on the CMA-ES for setting up a restart-based parameter variation method.

2. METHOD

In contrast to the given examples, we suggest to carry out the parameter setting by means of a tuning method, namely the sequential parameter optimization (SPO) by Bartz-Beielstein et al. [2]. The benefit of SPO is that it is a general purpose optimizer designed for extremely noisy real-valued problems and small budgets. Thus, it can be easily plugged into existing systems as a restart heuristic. It adopts ideas from design of experiments (DoE) [13] and design and analysis of computer experiments (DACE) [17, 18] to tackle the noisy optimization problem that parameter setting is.

Algorithm 1 outlines how SPO works. In the first exploratory phase, a random sample **D** (typically latin hypercube sample) of the search space is drawn and evaluated. To filter out noise, points may be sampled repeatedly and the results **Y** averaged. Then, a surrogate model $\mathcal{M}$ is created. In our case, DACE Kriging [11] is used as model. In an optimization loop, the model is then used to predict the next promising parameter configuration $\mathbf{d}_{\text{new}}$. When it is evaluated, the data is fed back into the model. If no new best configuration is found in a step, the number of repeats is increased by one. This means that the current best configuration is evaluated again, too. This adaptive mechanism ensures that wrong values that mislead the model get corrected over time.

Note that there is an important difference between our and the 'traditional' application of SPO. Usually, one would indentify the whole IPOP-CMA-ES as a monolithic block and try to optimize its parameters with SPO. Instead, we *replace* the IPOP heuristic with SPO and use it to control the remaining parameters. However, the SPO algorithm in itself remains completely unchanged. The only thing that has to be adapted to the new application are some of its parameters and the stopping criterion, which stays dependent on the function evaluations of the original problem. SPO is a true extension of IPOP as a restart heuristic, because

1. there are lower and upper bounds for the number of offspring λ, and
2. IPOP does not use any feedback from the optimization, except for the restart events themselves.

Thus, the possible configurations generated by IPOP can be included in SPO's initial sample and tried in order of increasing population size. So, the whole approach can be interpreted as a first exploration stage followed by a careful mix of exploitation and exploration.

Apart from possible performance gains, the real advantage is the flexibility of the approach. The user can smoothly adjust the trade-off between exploration and exploitation by determining the number of parameters, their boundaries, the number of configurations, repeats per configuration, and the order in which configurations are tried. The surrogate model then not only helps to exploit the information gained in this phase, but is easily visualizable afterwards. Thus, the user may employ it to choose a more exploitative set-up on the next problem instance, or generally look into parameter interactions on the algorithm/problem system. We will now go on to experimentally demonstrate both performance and practicability aspects in Section 3 before conclusions and outlook follow in Section 4.

3. EXPERIMENT

Research Question: How competitive is SPO as a restart heuristic in comparison with other specially designed heuristics?

Pre-experimental planning: The existing literature [1, 6, 12] and our preliminary experiments indicate that λ is the most important parameter. Thus, we initially tried out two SPO variants that use slightly different initial designs. Both generate a latin hypercube sample of five points in the region defined by Table 1. The number of offspring divided by the number of parents, λ/μ, is called selection pressure. It controls how greedy the algorithm is. The initial step size σ_{init} determines the size of the neighborhood at the beginning of the search. We shall add that the CMA-ES employs a weighting scheme that usually emphasizes the importance of the best individuals. This of course also influences the selection pressure. However, we stay with the default scheme here.

The smallest λ found in the initial design is replaced by $\lambda_{\text{def}} = 4 + \lfloor 3 \ln D \rfloor$. One of our two approaches arranges the configurations according to λ in ascending order (monotonically increasing, MOIN), while the other uses a random order except for the first configuration (RAND). Both begin with λ_{def}. These two heuristics are loosely inspired by IPOP [1] and BIPOP [6], respectively. However, preliminary experiments revealed that the RAND variant usually performed slightly worse. Thus, it is ignored in the remaining analysis.

To obtain more reliable results, we decided to increase the number of problem instances from 15 to 30 for the functions that appear interesting to us (see Table 2). These are generally the ones that force IPOP-CMA-ES to do a lot of restarts.

Task: The new appoach is tested on the BBOB 2010 problems [4]. The performance is assessed by keeping the desired quality fixed and estimating the expected running time (ERT). ERT depends on a given target function value, $f_t = f_{\text{opt}} + \Delta f$, and is computed over all relevant trials as the number of function evaluations executed during each trial

Table 1: **Considered parameters including their box constraints, default value, and additional transformation.**

Parameter	λ	λ/μ	σ_{init}
ROI	$\{\lambda_{\text{def}}, \ldots, 1000\}$	$[1.5, 2.5]$	$[1, 5]$
Default	$4 + \lfloor 3 \ln D \rfloor$	2	2
Transformation	$\log_{10}$	none	none

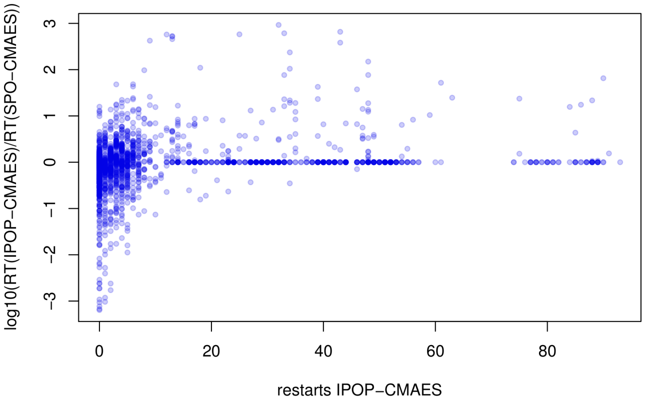


Figure 1: The RT ratio of IPOP-CMA-ES divided by SPO-CMA-ES is plotted against the number of restarts carried out by IPOP-CMA-ES. In this figure, larger values indicate an advantage for SPO-CMA-ES (contrary to Figure 3).

while the best function value did not reach f_t, summed over all trials and divided by the number of trials that actually reached f_t [8, 15]. Statistical significance is tested with the rank-sum test for a given target Δf_t using, for each trial, either the number of needed function evaluations to reach Δf_t (inverted and multiplied by -1), or, if the target was not reached, the best Δf-value achieved, measured only up to the smallest number of overall function evaluations for any unsuccessful trial under consideration.

Setup: As much as possible of the experimental setup is taken over from the BBOB 2010 guidelines [8]. Namely, the initial solution is chosen from $[-4, 4]^D$ and the final target precision is $\Delta f = 10^{-8}$. The maximal budget per instance is $10^6 \cdot \frac{D}{2}$ function evaluations. To be competitive, we reduce the SPO budget to the minimal possible values, namely an initial design of five configurations (default $10D$) and initially only one repeat per configuration (default 4). The experiments are carried out with the CMA-ES implementation in Python, version 0.9.51 [7]. The used BBOB version is 10.2.

Results: Figure 3 shows the ERT ratios between IPOP-CMA-ES and SPO-CMA-ES only on multimodal functions, but in all dimensions, while Table 2 shows the ERT and success rates in 5, 10, and 20 dimensions on all problems. Figure 1 reveals that SPO-CMA-ES can mainly yield lower running times when IPOP-CMA-ES makes more than ten restarts. Here, the x-axis represents the number of restarts of IPOP-CMA-ES. On the y-axis, we plotted the running time ratio of the two algorithms for each problem instance

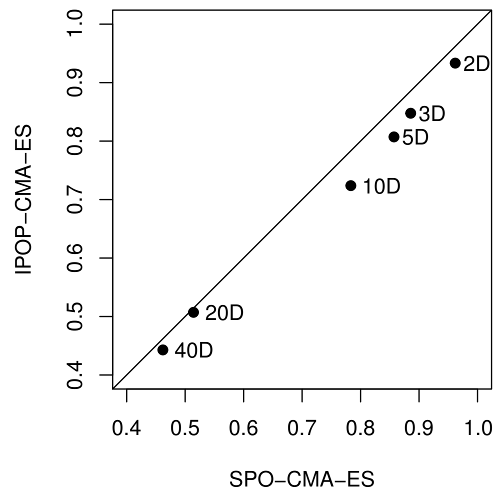


Figure 2: The success probabilities of both algorithms plotted against each other. The values are averaged over all problems.

and random seed. The points with a ratio of exactly one refer to problem instances that could not be solved by both. On the other hand, Figure 2 shows that the success rate of SPO-CMA-ES is generally higher than for IPOP-CMA-ES. Figure 4 finally illustrates some selected Kriging models which represent the main strength of the SPO approach. The models predict the best objective value to be achieved with each configuration.

Discussion: As Figure 1 suggests, it would have been advantageous for the SPO-CMA-ES to be even more conservative and try the exact default configuration at first. This way, the cases where IPOP-CMA-ES made no restarts and SPO-CMA-ES was worse could have been avoided. Apparently, the default values are indeed quite competitive on many problems.

We are convinced that the small difference between the success rates of the algorithms in $20D$ and $40D$ (see Figure 2) is due to a floor effect: The problems become too hard to be solved by any algorithm with the limited number of function evaluations.

The models in Figure 4 were chosen according to the number of points they were built from and their coefficient of determination (R^2). R^2 is an indicator for the proportion of sample variance that is explained by the model [21]. A model that perfectly fits the data would achieve a value of $R^2 = 1$. From this indicator, there still seems to be room for improvements of the models, most likely due to the low number of points and repeats. Each panel in the upper triangles of each subfigure shows an interaction effect between two parameters, while the lower triangles show the prediction uncertainty. The panels in each triangle share a common color map covering the range between the overall min and max values. This enables us to distinguish different strengths of effects. Naturally, the uncertainty is low (blue) around sampled points and high (red) far away from them. Note that this uncertainty estimation only accounts for the error possibly made by the model, not the random noise that comes from the stochastic process. Regarding the interaction effects, we can see from the absence of deep red or

5-D

Δf	1e+1	1e+0	1e-1	1e-3	1e-5	1e-7	#succ
f1	11	12	12	12	12	12	15/15
0: I	3.4	9.8	15	29	42	54	15/15
1: S	3.5	8.7	15	26	**38**	49	15/15
f2	83	87	88	90	92	94	15/15
0: I	13	16	18	19	21	22	15/15
1: S	15	17	18	20	21	22	15/15
f3	720	1600	1600	1600	1700	1700	15/15
0: I	1.1	51	3.1e3	3.1e3	3.0e3	3.0e3	10/30
1: S	1.8	32	490	500	500	500	26/30
f4	810	1600	1700	1800	1900	1900	15/15
0: I	2.3	∞	∞	∞	∞	∞ *2.5e6*	0/30
1: S	2	∞	∞	∞	∞	∞ *2.5e6*	0/30
f5	10	10	10	10	10	10	15/15
0: I	5.9	17	21	22	22	22	15/15
1: S	7.6	18	25	26	26	26	15/15
f6	110	210	280	580	1000	1300	15/15
0: I	1.6	1.8	2	1.6	1.2	1.2	15/15
1: S	1.5	1.6	1.9	1.4	1.1	1.1	15/15
f7	24	320	1200	1600	1600	1600	15/15
0: I	4.5	2.3	1.5	1.2	1.2	1.3	15/15
1: S	4.4	1.4	1.4	1.2	1.2	1.3	15/15
f8	73	270	340	390	410	420	15/15
0: I	3.7	4.2	4.8	5.2	5.4	5.6	15/15
1: S	3	5.5	6.1	6.3	6.5	6.9	15/15
f9	35	130	210	300	340	370	15/15
0: I	6.4	7.3	6.6	6	6	5.9	15/15
1: S	6	10	8.4	7.4	7.2	7	15/15
f10	350	500	570	630	830	880	15/15
0: I	3.3	3.4	3.2	3.1	2.6	2.6	15/15
1: S	3.2	**2.7**	2.8	2.8	2.3	2.3	15/15
f11	140	200	760	1200	1500	1700	15/15
0: I	7.9	7.1	2.1	1.6	1.4	1.3	15/15
1: S	7.5	6.9	2.1	1.6	1.4	1.3	15/15
f12	110	270	370	460	1300	1500	15/15
0: I	6.5	6.3	7.6	8.1	3.6	3.5	15/15
1: S	8.4	6.4	6.6	7.1	3.2	3.2	15/15
f13	130	190	250	1300	1800	2300	15/15
0: I	4.2	**4.9**	5.9	1.6	1.6	1.5	15/15
1: S	5.4	7.5	6.9	1.9	2	1.9	15/15
f14	9.8	41	58	140	250	480	15/15
0: I	1.4	2.8	3.6	4.8	5.6	4.4	15/15
1: S	2.5	3.1	3.8	4.7	**5.2**	4.3	15/15
f15	510	9300	1.9e4	2.0e4	2.1e4	2.1e4	14/15
0: I	1.5	0.94	0.98	0.98	0.99	0.99	15/15
1: S	1.8	1.1	0.85	0.87	0.89	0.91	15/15
f16	120	610	2700	1.0e4	1.2e4	1.2e4	15/15
0: I	1.6	3.3	2.1	0.91	0.91	0.91	30/30
1: S	1.3	3.4	3	1.3	1.3	1.3	30/30
f17	5.2	210	900	3700	6400	7900	15/15
0: I	4.9	1.1	1.1	0.92	1	1.1	30/30
1: S	5	1	1.1	1.1	0.89	1	30/30
f18	100	380	4000	9300	1.1e4	1.2e4	15/15
0: I	5.5	3.7	1.1	1.1	1.1	1.2	30/30
1: S	1.1	2.7	0.98	**0.86**	**0.88**	**0.9**	30/30
f19	1	1	240	1.2e5	1.2e5	1.2e5	15/15
0: I	19	2.6e3	360	2	2	2	30/30
1: S	25	6.9e3	270	**1.3**	**1.3**	**1.3**	30/30
f20	16	850	3.8e4	5.4e4	5.5e4	5.5e4	14/15
0: I	**3.2**	9.9	1.7	1.3	1.4	1.4	30/30
1: S	4.4	9.9	1.5	1.1	1.2	1.2	30/30
f21	41	1200	1700	1700	1700	1800	14/15
0: I	4.6	16	30	32	32	33	30/30
1: S	3.1	7.7	31	31	31	31	30/30
f22	71	390	940	1000	1000	1100	14/15
0: I	6.3	33	310	630	610	600	28/30
1: S	8.8	69	570	550	530	520	28/30
f23	3	520	1.4e4	3.2e4	3.3e4	3.4e4	15/15
0: I	2.6	9.3	**3.4**	2.1	2	2	30/30
1: S	2.6	10	12	5.6	5.5	5.4	30/30
f24	1600	2.2e5	6.4e6	9.6e6	1.3e7	1.3e7	3/15
0: I	2.8	40	5.7	7.6	5.7	5.7	1/30
1: S	3	**8.5**	**0.85**	**1.1**	**0.86**	**0.86**	6/30

10-D

Δf	1e+1	1e+0	1e-1	1e-3	1e-5	1e-7	#succ
f1	22	23	23	23	23	23	15/15
0: I	5.3	12	18	31	44	56	15/15
1: S	5.8	13	19	31	45	58	15/15
f2	190	190	190	190	190	200	15/15
0: I	21	24	25	27	29	30	15/15
1: S	21	25	26	28	30	31	15/15
f3	1700	3600	3600	3600	3600	3700	15/15
0: I	3.6	770	1.3e4	1.3e4	1.3e4	1.3e4	3/30
1: S	3.1	**170**	**760**	**750**	**750**	**750**	24/30
f4	2200	3600	3700	3700	3700	2.9e4	12/15
0: I	6.3	∞	∞	∞	∞	∞ *5.0e6*	0/30
1: S	5.4	∞	∞	∞	∞	∞ *5.0e6*	0/30
f5	20	20	20	20	20	20	15/15
0: I	17	36	38	39	39	39	15/15
1: S	14	28	31	33	33	33	15/15
f6	410	620	830	1300	1800	2400	15/15
0: I	1.7	1.8	1.8	1.8	1.7	1.6	15/15
1: S	1.6	1.8	1.9	1.8	1.7	1.7	15/15
f7	170	1600	4200	5100	5100	5400	15/15
0: I	**2**	1.4	1.4	1.5	1.5	1.4	15/15
1: S	3.5	1.5	1.1	1.3	1.3	1.2	15/15
f8	330	920	1100	1300	1300	1300	15/15
0: I	2.4	4.2	4.3	4.4	4.6	4.7	15/15
1: S	2.5	4.8	4.9	4.9	5	5.2	15/15
f9	200	650	860	1100	1100	1200	15/15
0: I	4.6	6.4	5.9	5.4	5.4	5.5	15/15
1: S	3.3	6.2	5.9	5.4	5.5	5.5	15/15
f10	1800	2200	2500	2800	4500	4700	15/15
0: I	2	2	2	1.9	1.3	1.3	15/15
1: S	2.1	2.1	2	1.9	1.3	1.3	15/15
f11	270	1000	2600	3300	4100	4800	15/15
0: I	12	3.8	1.7	1.5	1.3	1.2	15/15
1: S	12	3.7	1.6	1.5	1.3	1.2	15/15
f12	520	900	1200	1600	3700	5200	15/15
0: I	2.5	3	3.7	4.8	2.7	2.3	15/15
1: S	3.2	3.2	3.5	4.6	2.6	2.2	15/15
f13	390	600	800	4600	6200	7800	15/15
0: I	4	5.7	5.1	1.3	1.7	1.6	15/15
1: S	5.5	6.1	5.9	1.5	1.9	2	15/15
f14	37	98	130	390	690	4300	15/15
0: I	2.1	3	**3.7**	4	5.2	1.4	15/15
1: S	3	3.4	4.1	4.1	5.3	1.4	15/15
f15	4800	3.9e4	7.4e4	7.6e4	7.8e4	8.0e4	12/15
0: I	0.8	1.2	0.94	0.95	0.97	0.98	15/15
1: S	0.87	1.1	0.8	0.82	0.84	0.86	15/15
f16	430	7000	1.6e4	5.1e4	6.6e4	7.2e4	15/15
0: I	1.9	0.81↓	**1.3**	1	0.96	0.91	30/30
1: S	2.1	0.6↓	2.6	1.5	1.3	1.3	30/30
f17	26	430	2200	9900	2.0e4	2.7e4	15/15
0: I	1.8	1.2	0.87↓	1	0.84	1.1	30/30
1: S	2.4	1.1	0.76↓	0.96	0.82	**0.87**	30/30
f18	240	840	7000	2.8e4	3.7e4	4.3e4	15/15
0: I	1.1	3.1	1.3	1.1	1	1	30/30
1: S	1.3	1.8	1.1	0.87	**0.86↓**	**0.84↓**	30/30
f19	1	1	1.1e4	1.4e6	1.4e6	1.4e6	15/15
0: I	**59**	2.4e4	25	0.85↓	0.85↓	0.85↓	30/30
1: S	90	3.2e4	25	0.95↓	0.94↓	0.94↓	29/30
f20	32	1.5e4	5.5e5	5.7e5	5.8e5	5.9e5	15/15
0: I	**5.4**	2.8	1.1	1.1	1.1	1.1	30/30
1: S	6.4	2.6	1.8	1.7	1.7	1.7	29/30
f21	130	2200	4400	4600	5100	1.1e4	15/15
0: I	6.9	31	63	61	56	25	30/30
1: S	8.6	33	74	70	64	29	30/30
f22	98	2800	6400	6800	8300	1.0e4	15/15
0: I	**12**	360	4.0e3	3.8e3	3.1e3	2.5e3	5/30
1: S	50	350	2.5e3	2.3e3	1.9e3	1.7e3	7/30
f23	2.8	920	1.6e4	2.0e5	2.1e5	2.1e5	15/15
0: I	1.8	36	21	11	11	11	26/30
1: S	2.5	48	18	8.7	8.6	8.4	29/30
f24	9.9e4	1.0e6	7.5e7	7.5e7	7.5e7	7.5e7	1/15
0: I	32	6.9	1	∞	∞	∞ *5.0e6*	0/30
1: S	26	3.6	2	2	2	2	1/30

20-D

Δf	1e+1	1e+0	1e-1	1e-3	1e-5	1e-7	#succ
f1	43	43	43	43	43	43	15/15
0: I	7.9	14	20	33	45	58	15/15
1: S	8.3	14	20	32	45	57	15/15
f2	380	390	390	390	390	390	15/15
0: I	34	40	44	47	48	49	15/15
1: S	32	40	43	47	48	49	15/15
f3	5100	7600	7600	7600	7600	7700	15/15
0: I	26	∞	∞	∞	∞	∞ *1.0e7*	0/30
1: S	25	**3.9e4**	∞	∞	∞	∞ *1.0e7*	0/30
f4	4700	7600	7700	7700	7800	1.4e5	9/15
0: I	∞	∞	∞	∞	∞	∞ *1.0e7*	0/30
1: S	**6.2e4**	∞	∞	∞	∞	∞ *1.0e7*	0/30
f5	41	41	41	41	41	41	15/15
0: I	19	30	34	34	34	34	15/15
1: S	**16**	28	31	32	32	32	15/15
f6	1300	2300	3400	5200	6700	8400	15/15
0: I	1.4	1.3	1.2	1.1	1.2	1.2	15/15
1: S	1.4	1.2	1.1	1.1	1.1	1.2	15/15
f7	1400	4300	9500	1.7e4	1.7e4	1.7e4	15/15
0: I	1.8	5	3.1	1.9	1.9	1.9	15/15
1: S	1.4	**3.5**	**2.4**	**1.5**	**1.5**	**1.5**	15/15
f8	2000	3900	4000	4200	4400	4500	15/15
0: I	3.6	3.9	4.2	4.4	4.5	4.5	15/15
1: S	3.8	5.6	5.9	6	6	6.1	15/15
f9	1700	3100	3300	3500	3600	3700	15/15
0: I	4.3	5.4	5.8	5.9	5.9	5.9	15/15
1: S	4.3	5.6	5.9	6.1	6.1	6.1	15/15
f10	7400	8700	1.1e4	1.5e4	1.7e4	1.7e4	15/15
0: I	1.7	1.8	1.6	1.2	1.1	1.1	15/15
1: S	1.7	1.8	1.6	1.2	1.1	1.1	15/15
f11	1000	2200	6300	9800	1.2e4	1.5e4	15/15
0: I	9.4	4.9	1.9	1.3	1.2	1	15/15
1: S	9.2	4.9	1.9	1.3	1.1	1	15/15
f12	1000	1900	2700	4100	1.2e4	1.4e4	15/15
0: I	1.9	1.6	**2.5**	2.9	1.4	1.5	15/15
1: S	2.2	3	3.8	3.8	1.7	1.8	15/15
f13	650	2000	2800	1.9e4	2.4e4	3.0e4	15/15
0: I	3.1	4.8	7.2	1.5	1.7	2.3	15/15
1: S	4.5	4.5	5	1.7	2	2.2	15/15
f14	75	240	300	930	1600	1.6e4	15/15
0: I	3.4	2.7	3.4	4	6.1	1.2	15/15
1: S	4.1	2.8	3.6	3.8	5.9	1.2	15/15
f15	3.0e4	1.5e5	3.1e5	3.2e5	4.5e5	4.6e5	15/15
0: I	1.1	1	0.68	0.69	0.51↓	0.52↓	15/15
1: S	1	1	0.6	0.62	0.46↓	0.48↓	15/15
f16	1400	2.7e4	7.7e4	1.9e5	2.0e5	2.2e5	15/15
0: I	**1.3**	**0.59**	1.1	1.9	1.9	1.8	30/30
1: S	1.6	1	1.6	2.5	2.8	2.5	30/30
f17	63	1000	4000	3.1e4	5.6e4	8.0e4	15/15
0: I	**2.1**	0.99	1.1	0.7↓	0.94	0.96	30/30
1: S	2.7	1.1	1.2	0.79↓	0.82	0.85↓	30/30
f18	620	4000	2.0e4	6.8e4	1.3e5	1.5e5	15/15
0: I	1	1.3	**0.83**	0.97	0.99	0.97	30/30
1: S	1.3	1.8	0.96	**0.79↓**	0.87	0.87	30/30
f19	1	1	3.4e5	6.2e6	6.7e6	6.7e6	15/15
0: I	190	5.2e4	2.9	**4.4**	**5.6**	**5.5**	7/30
1: S	220	1.6e5	**1.6**	9.3	11	11	4/30
f20	82	4.6e4	3.1e6	5.5e6	5.6e6	5.6e6	14/15
0: I	**4.9**	5.4	46	∞	∞	∞ *1.0e7*	0/30
1: S	5.5	5.7	18	53	52	52	1/30
f21	560	6500	1.4e4	1.5e4	1.6e4	1.8e4	15/15
0: I	8.4	240	240	230	220	200	26/30
1: S	23	160	190	180	170	150	26/30
f22	470	5600	2.3e4	2.5e4	2.7e4	1.3e5	12/15
0: I	38	720	∞	∞	∞	∞ *1.0e7*	0/30
1: S	42	750	6.1e3	5.8e3	5.3e3	1.1e3	2/30
f23	3.2	1600	6.7e4	4.9e5	8.1e5	8.4e5	15/15
0: I	2.1	8.6e3	380	∞	∞	∞ *1.0e7*	0/30
1: S	1.9	6.5e3	170	200	120	120	3/30
f24	1.3e6	7.5e6	5.2e7	5.2e7	5.2e7	5.2e7	3/15
0: I	4.4	2.3	∞	∞	∞	∞ *1.0e7*	0/30
1: S	5.9	3.7	∞	∞	∞	∞ *1.0e7*	0/30

Table 2: Expected running time (ERT in number of function evaluations) divided by the best ERT measured during BBOB-2009 (given in the respective first row) for different Δf values for functions f_1–f_{24}. The median number of conducted function evaluations is additionally given in *italics*, if $\mathrm{ERT}(10^{-7}) = \infty$. #succ is the number of trials that reached the final target $f_{\mathrm{opt}} + 10^{-8}$. 0: I is IPOP-CMA-ES and 1: S is SPO-CMA-ES. Bold entries are statistically significantly better compared to the other algorithm with $p \leq 0.05$. Up and down arrows indicate significant differences to the best BBOB-2009 algorithm.

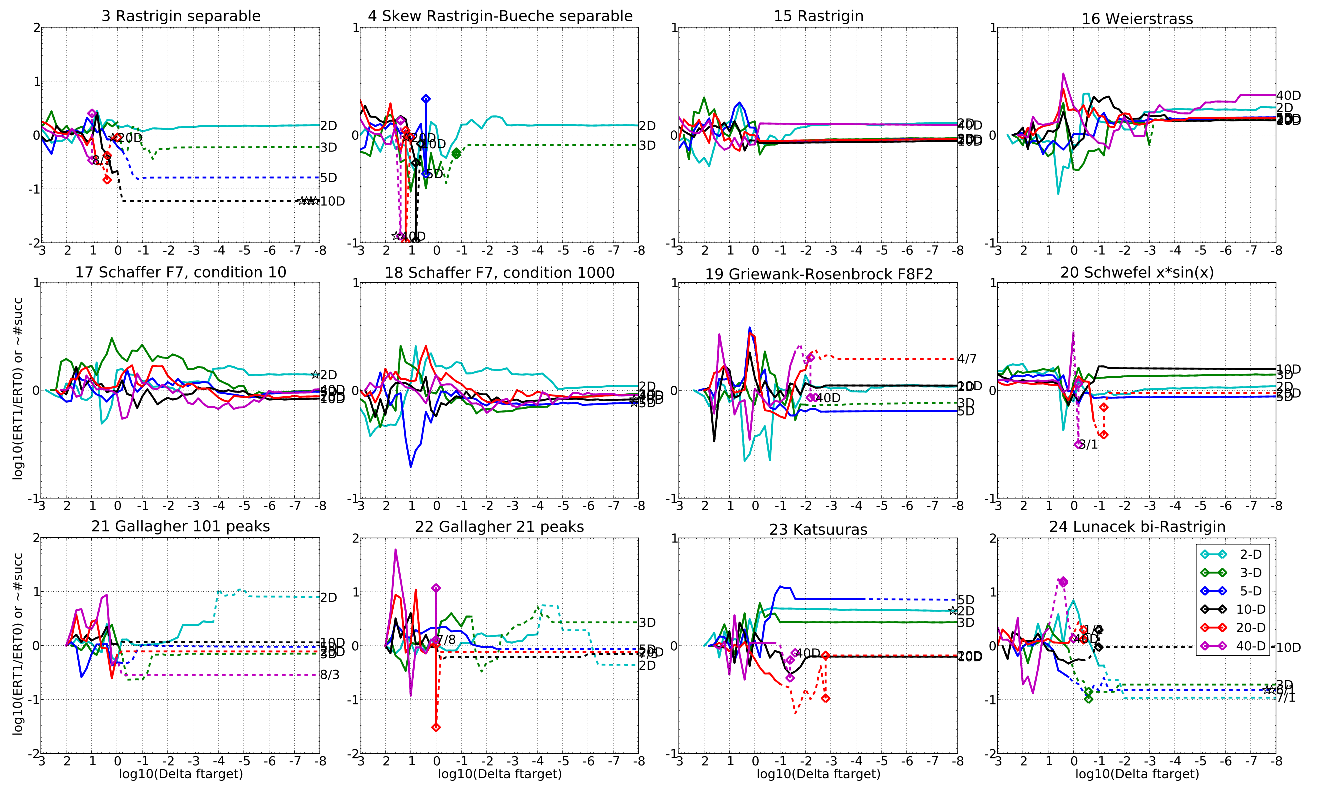


Figure 3: ERT ratio of SPO-CMA-ES divided by IPOP-CMA-ES versus $\log_{10}(\Delta f)$ for multimodal functions in 2, 3, 5, 10, 20, 40-D. Ratios $< 10^0$ indicate an advantage of SPO-CMA-ES, smaller values are always better. The line gets dashed when for any algorithm the ERT exceeds thrice the median of the trial-wise overall number of f-evaluations for the same algorithm on this function. Symbols indicate the best achieved Δf-value of one algorithm (ERT gets undefined to the right). The dashed line continues as the fraction of successful trials of the other algorithm, where 0 means 0% and the y-axis limits mean 100%, values below zero for SPO-CMA-ES. The line ends when no algorithm reaches Δf anymore. The number of successful trials is given, only if it was in $\{1 \ldots 9\}$ for SPO-CMA-ES (1st number) and non-zero for IPOP-CMA-ES (2nd number). Results are significant with $p \leq 0.05$ for one star and $p \leq 10^{-\#\star}$ otherwise, with Bonferroni correction within each figure.

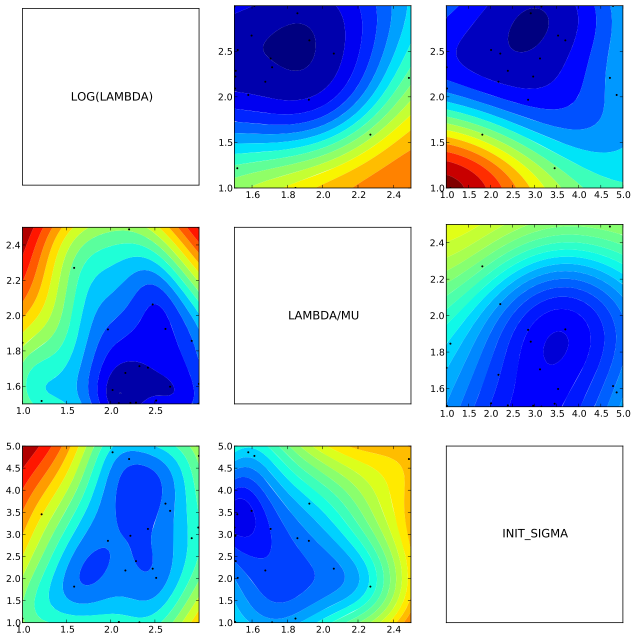
blue colors, that λ/μ and σ_{init} usually have a weaker influence than λ. Furthermore, high λ values yield better results (blue), as was already expected. Interestingly, there seem to be cases where a lower selection pressure λ/μ is promising (Fig. 4a, 4b), and cases where the opposite is true (Fig. 4c, 4d). This cannot be said about σ_{init}, so that it could be probably omitted from the optimization without significant loss. Note, however, that all problems in the BBOB test bed have a feasible region of $[-5, 5]^D$. On unbounded and/or real-world problems it might be more difficult to find an appropriate σ_{init}.

4. CONCLUSIONS AND OUTLOOK

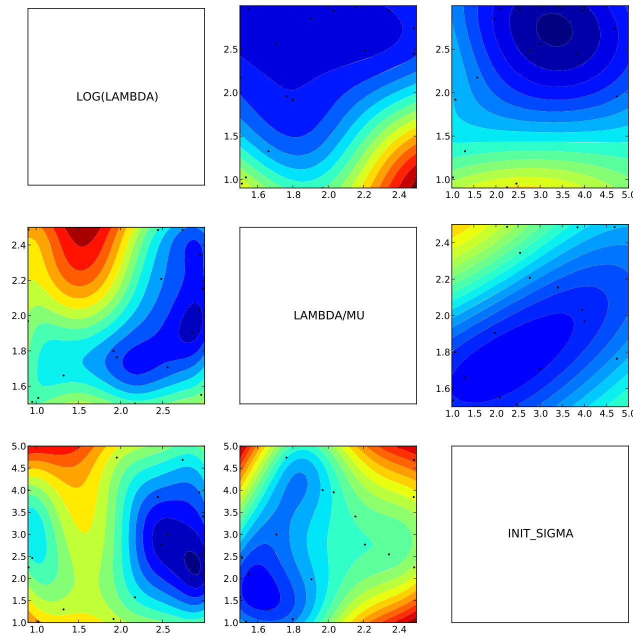
SPO can be regarded as an extension of the IPOP heuristic. So, we compared the newly developed SPO-CMA-ES to IPOP-CMA-ES, an algorithm that has proven to be still competitive on recent benchmarks [16]. The experiments showed that SPO-CMA-ES can achieve a better running time and success rate than IPOP-CMA-ES, especially on the harder problems of the BBOB test bed. Additionally, SPO-CMA-ES can also provide insight into the parameter influences, thanks to its roots in parameter tuning. The surrogate model yields information whether parameters interact with each other or have no influence. The user can then employ this information to refine the parameter setting for a possible next instance of the same problem class. Thus, the approach can blend smoothly between parameter tuning and parameter control. After all, parameter setting (tuning) is just some kind of noisy optimization. Thus, we would like to encourage researchers not to insist on a sharp distinction of tuning and control but to take methods from the respective other fields into account.

To make SPO-CMA-ES independent of the actual objective values, it might be promising to apply a rank transformation prior to modeling. This has already been demonstrated to be effective in a parameter tuning context by Wessing and Wagner [22]. In the future, it also would be interesting to add SPO as restart heuristic to other algorithms that might benefit even more from an improved parameter control, as CMA-ES is already one of the most successful local search algorithms with a very robust set of default parameters.

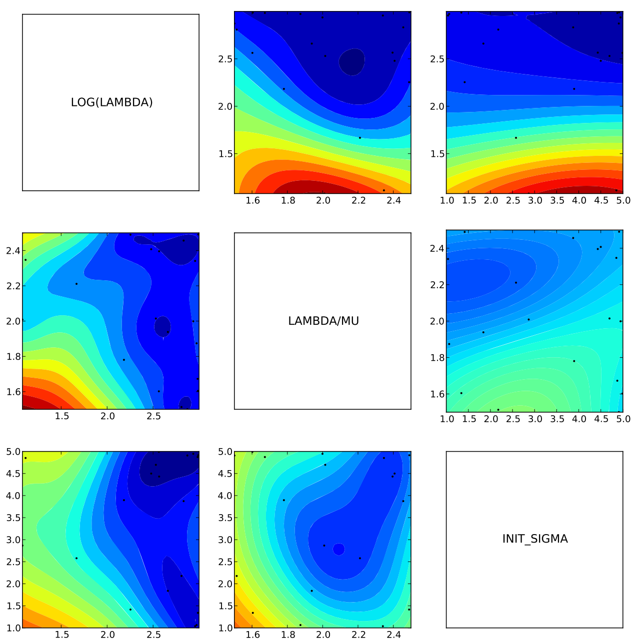
Another interesting question would be if it is possible to incorporate some kind of cost management into the restart heuristic. Such a heuristic would also account for the number of objective function evaluations carried out by each



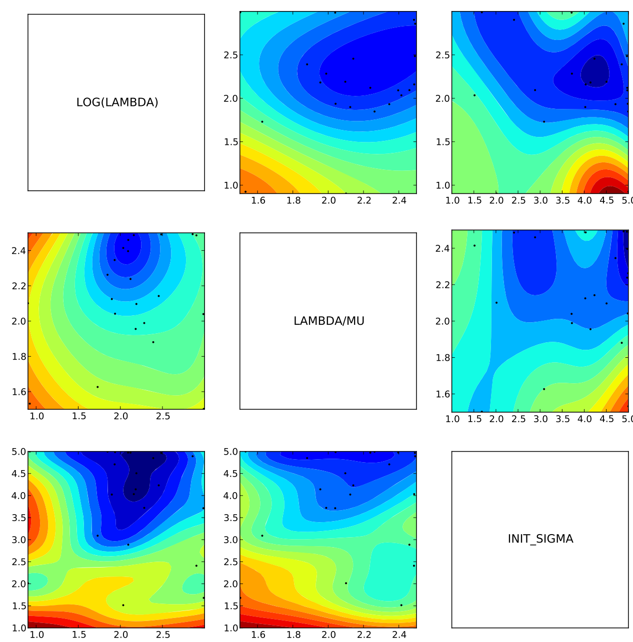
(a) 96 restarts, distributed over 19 configurations on f_3 in $10D$ ($R^2 = 0.41$). The run did not reach f_t.



(b) 67 restarts, distributed over 17 configurations on f_4 in $5D$ ($R^2 = 0.28$). The run did not reach f_t.



(c) 94 restarts, distributed over 19 configurations on f_{21} in $20D$ ($R^2 = 0.14$). The run did not reach f_t.



(d) 101 restarts, distributed over 22 configurations on f_{22} in $5D$ ($R^2 = 0.19$). The run reached f_t in 1.8×10^6 evaluations.

Figure 4: Kriging model of the CMA-ES parameters on some selected problem instances.

configuration, as BIPOP already does. However, preliminary experiments with an aggregated objective failed, which is why we suspect that a multi-objective approach would be necessary.

5. REFERENCES

[1] A. Auger and N. Hansen. A restart CMA evolution strategy with increasing population size. In *IEEE Congress on Evolutionary Computation (CEC'05)*, volume 2, pages 1769–1776. IEEE Press, 2005.

[2] T. Bartz-Beielstein, C. Lasarczyk, and M. Preuss. Sequential parameter optimization. In *IEEE Congress on Evolutionary Computation (CEC'05)*, volume 1, pages 773–780. IEEE Press, 2005.

[3] A. Eiben, R. Hinterding, and Z. Michalewicz. Parameter control in evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 3(2):124 –141, July 1999.

[4] S. Finck, N. Hansen, R. Ros, and A. Auger. Real-parameter black-box optimization benchmarking 2009: Presentation of the noiseless functions. Technical Report 2009/20, Research Center PPE, 2009. Updated February 2010.

[5] N. Hansen. The CMA Evolution Strategy: A Tutorial, April 26 2008. accessed 21-01-09, `http://www.bionik.tu-berlin.de/user/niko/cmatutorial.pdf`.

[6] N. Hansen. Benchmarking a BI-population CMA-ES on the BBOB-2009 function testbed. In *Workshop Proceedings of the GECCO Genetic and Evolutionary Computation Conference*, pages 2389–2395. ACM, July 2009.

[7] N. Hansen. CMA-ES implementation in Python. online, November 2010. `http://www.lri.fr/~hansen/cma.py`.

[8] N. Hansen, A. Auger, S. Finck, and R. Ros. Real-parameter black-box optimization benchmarking 2010: Experimental setup. Technical Report RR-7215, INRIA, 2010.

[9] N. Hansen and A. Ostermeier. Completely Derandomized Self-Adaptation in Evolution Strategies. *Evolutionary Computation*, 9(2):159–195, 2001.

[10] H. Hoos and T. Stützle. *Stochastic Local Search – Foundations and Applications*. Morgan Kaufmann, San Francisco, 2004.

[11] S. N. Lophaven, H. B. Nielsen, and J. Søndergaard. DACE – A Matlab Kriging Toolbox. Technical Report IMM-REP-2002-12, Informatics and Mathematical Modelling, Technical University of Denmark, Copenhagen, Denmark, August 2002.

[12] M. Lunacek, D. Whitley, and A. Sutton. The impact of global structure on search. In *Parallel Problem Solving from Nature – PPSN X*, volume 5199 of *Lecture Notes in Computer Science*, pages 498–507. Springer Berlin / Heidelberg, 2008.

[13] D. C. Montgomery. *Design and Analysis of Experiments*. Wiley, New York, 4th edition, 1997.

[14] M. Preuss, G. Rudolph, and S. Wessing. Tuning optimization algorithms for real-world problems by means of surrogate modeling. In *Genetic and Evolutionary Computation Conference (GECCO)*, pages 401–408, 2010.

[15] K. Price. Differential evolution vs. the functions of the second ICEO. In *Proceedings of the IEEE International Congress on Evolutionary Computation*, pages 153–157, 1997.

[16] R. Ros. Black-box optimization benchmarking the IPOP-CMA-ES on the noiseless testbed: comparison to the BIPOP-CMA-ES. In *Proceedings of the 12th annual conference companion on Genetic and evolutionary computation*, GECCO '10, pages 1503–1510. ACM, 2010.

[17] J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn. Design and analysis of computer experiments. *Statistical Science*, 4(4):409–423, November 1989.

[18] T. J. Santner, B. J. Williams, and W. I. Notz. *The Design and Analysis of Computer Experiments*. Springer, 2003.

[19] K. Sastry. Single and multiobjective genetic algorithm toolbox for matlab in c++. Technical Report 2007017, Illinois Genetic Algorithms Laboratory, University of Illinois at Urbana-Champaign, 2007.

[20] H.-P. Schwefel. *Evolution and Optimum Seeking*. Wiley, New York, 1995.

[21] R. G. Steel and J. H. Torrie. *Principles and Procedures of Statistics*. McGraw-Hill, 1960.

[22] S. Wessing and T. Wagner. A rank transformation can improve sequential parameter optimization. Algorithm Engineering Report TR10-2-007, Technische Universität Dortmund, 2010.

[23] K. Zielinski and R. Laur. Stopping Criteria for a Constrained Single-Objective Particle Swarm Optimization Algorithm. *Informatica*, 31(1):51–59, 2007.