Doubly Trained Evolution Control for the Surrogate CMA-ES

Zbyněk Pitra[1,2,3(✉)], Lukáš Bajer[4,5], and Martin Holeňa[3]

[1] National Institute of Mental Health, Topolová 748, 250 67 Klecany, Czech Republic
z.pitra@gmail.com
[2] Faculty of Nuclear Sciences and Physical Engineering,
Czech Technical University in Prague, Břehová 7, 115 19 Prague 1, Czech Republic
[3] Institute of Computer Science, Academy of Sciences of the Czech Republic,
Pod Vodárenskou věží 2, 182 07 Prague 8, Czech Republic
{bajer,holena}@cs.cas.cz
[4] Faculty of Mathematics and Physics, Charles University in Prague,
Malostranské nám. 25, 118 00 Prague 1, Czech Republic
[5] Unicorn College, V Kapslovně 2767/2, 130 00 Prague 3, Czech Republic

Abstract. This paper presents a new variant of surrogate-model utilization in expensive continuous evolutionary black-box optimization. This algorithm is based on the surrogate version of the CMA-ES, the Surrogate Covariance Matrix Adaptation Evolution Strategy (S-CMA-ES). Similarly to the original S-CMA-ES, expensive function evaluations are saved through a surrogate model. However, the model is retrained after the points in which its prediction was most uncertain have been evaluated by the true fitness in each generation. We demonstrate that within small budget of evaluations, the new variant of S-CMA-ES improves the original algorithm and outperforms two state-of-the-art surrogate optimizers, except a few evaluations at the beginning of the optimization process.

Keywords: Black-box optimization · Surrogate model · Evolution control · Gaussian process

1 Introduction

In many research and engineering tasks, optimization of real-world black-box functions that are costly to evaluate is a challenging problem of great importance. A single evaluation of the expensive function may require a great amount of resources in terms of time and performed experiments, measurements or simulations. In order to decrease the number of evaluations of the costly black-box function and still produce reasonably good solutions, a surrogate model can be employed [15]. Such models are built using the previous evaluations of the black-box function, and then are used to predict the values of new points instead of the original function.

J. Handl et al. (Eds.): PPSN XIV 2016, LNCS 9921, pp. 59–68, 2016.
DOI: 10.1007/978-3-319-45823-6_6

Nowadays, the *Covariance Matrix Adaptation Evolution Strategy* (CMA-ES) [5] is one of the most robust algorithms on real-world problems and is considered to be the state-of-the-art of continuous black-box optimization. In recent years, several surrogate-model approaches have been developed to increase the performance of the CMA-ES.

The ^{s*}ACM-ES [12] employs ordinal regression models based on SVM to estimate the ordering of the fitness function values. Furthermore, the parameters of the ordinal model are themselves optimized utilizing the CMA-ES algorithm during the optimization of the black-box function. In order to avoid premature convergence to local optima, the strategies ^{s*}ACM-ES-k [11] and BIPOP-^{s*}ACM-ES-k [13] improving ^{s*}ACM-ES by increasing the population size in generations evaluated by the model have been developed.

Another surrogate-assisted approach using continuous regression models to estimate the function values, called the Surrogate CMA-ES (S-CMA-ES), has been proposed in [1]. This approach employs models capable to predict the whole distribution of values of the objective function; however, the S-CMA-ES does not make use of that capacity of the models, and exploits only the means of the distribution of its values. On the other hand, several authors [10,14] have demonstrated the effective utilization of various criteria using the variances of predictions (e.g. expected improvement, probability of improvement) in optimization.

Different usage of surrogate modelling presents *Sequential Model-based Algorithm Configuration* method (SMAC) [8]. It fits surrogate models of algorithm settings in a parameter space and utilizes those models to make decisions about which settings to investigate. To make SMAC more useful in continuous optimization, random forest were replaced by Gaussian processes as a surrogate model in SMAC-BBOB [7].

The main contribution of this paper is to introduce Doubly Trained S-CMA-ES, the extension of the S-CMA-ES, using not only the means of the distributions predicted by the surrogate model, but also variances of those distributions. We experimentally evaluate different settings of this approach on the BBOB/COCO testing set [3,4] and compare it with the original version of the S-CMA-ES, the surrogate-assisted ^{s*}ACM-ES-k, and the SMAC method.

The remainder of the paper is structured as follows. Section 2 describes the S-CMA-ES and its model-training method. Section 3 defines its proposed extension, Doubly Trained S-CMA-ES. Section 4 contains the experimental part. Section 5 summarizes the results and draws conclusions.

2 Surrogate CMA-ES and Generation Evolution Control

The S-CMA-ES, introduced in [1], is a surrogate-model-based modification of the CMA-ES. After the initialization step, the following steps shown in Algorithm 1 are proceeded by the S-CMA-ES until the target fitness value is found: First, the population of one generation is sampled using the CMA-ES. Then, the evolution control is employed to evaluate sampled points. Finally, the CMA-ES strategy parameters (σ, $\mathbf{m}$, $\mathbf{C}$, etc.) are calculated using the original CMA-ES algorithm.

Algorithm 1. S-CMA-ES

Input: λ (population-size), y_{target} (target value), f (original fitness function), r (maximal distance between training points and $\mathbf{m}$), n_{MIN}, n_{MAX} (minimal and maximal number of points for model training), n_{orig} (number of original-evaluated points), $\mathcal{C}$ (uncertainty criterion), g_m (number of model generations)

1: $\sigma, \mathbf{m}, \mathbf{C}, g \leftarrow$ CMA-ES initialize
2: $\mathcal{A} \leftarrow \emptyset$
3: **while** $\min_{k \in \{1,\ldots,\lambda\}} y_k > y_{\text{target}}$ **do**
4: $\quad \mathbf{x}_k \sim \mathcal{N}\left(\mathbf{m}, \sigma^2\mathbf{C}\right) \qquad k \in \{1, \ldots, \lambda\}$ $\qquad$ {*CMA-ES sampling*}
5: $\quad (\{y_k\}_{k=1}^{\lambda}, \mathcal{A}) \leftarrow \text{evolutionControl}(\lambda, f, \mathcal{A}, \{\mathbf{x}_k\}_{k=1}^{\lambda}, \sigma, \mathbf{m}, \mathbf{C}, \ldots)$
6: $\quad \sigma, \mathbf{m}, \mathbf{C}, g \leftarrow$ CMA-ES update
7: **end while**
8: $\mathbf{x}_{\text{res}} \leftarrow \mathbf{x}_k$ where y_k is minimal

Output: $\mathbf{x}_{\text{res}}$

Algorithm 2. Generation evolutionControl in S-CMA-ES

Input: λ, σ, $\mathbf{m}$, $\mathbf{C}$, f, $\mathcal{A}$, $\{\mathbf{x}_k\}_{k=1}^{\lambda}$ (CMA-ES sampled population), g (generation), g_m (number of model generations), r, n_{MIN}, n_{MAX}

1: **if** g is original-evaluated **then**
2: $\quad y_k \leftarrow f(\mathbf{x}_k) \qquad k = 1, \ldots, \lambda$ $\qquad$ {*fitness evaluation*}
3: $\quad \mathcal{A} = \mathcal{A} \cup \{(\mathbf{x}_k, y_k)\}_{k=1}^{\lambda}$
4: $\quad f_{\mathcal{M}} \leftarrow \text{trainModel}(\mathcal{A}, \sigma, \mathbf{m}, \mathbf{C}, r, n_{\text{MIN}}, n_{\text{MAX}})$
5: **else**
6: $\quad y_k \leftarrow f_{\mathcal{M}}(\mathbf{x}_k) \qquad k = 1, \ldots, \lambda$ $\qquad$ {*model evaluation*}
7: $\quad$ **if** g_m model generations passed **then** mark $(g + 1)$ as original-evaluated
8: **end if**

Output: $(y_k)_{k=1}^{\lambda}$, $\mathcal{A}$

The *generation-based evolution control* (following Jin's terminology [9]) is used in S-CMA-ES as the evolution control step (Step 5 in Algorithm 1). This step is presented in more detail in Algorithm 2. At first, the population of one generation sampled using CMA-ES is evaluated by the original fitness function. Then, a surrogate model is constructed using the original-evaluated data. However, if the model has not enough training points, the original fitness function is utilized to evaluate sampled points. In the few subsequent generations, the function values of the samples are computed using the surrogate model; they are, consequently, used to calculate new CMA-ES parameters.

The phase of training the surrogate model is shown in Algorithm 3. In order to increase the accuracy of surrogate-model predictions (e.g. Gaussian process predictions), the points that have the Mahalanobis distance from the current CMA-ES mean $\mathbf{m}$ less than or equal to a specific bound r are selected for training. If the size of the training set is sufficient, k-NN clustering chooses n_{MAX} training points which are transformed to the basis defined by eigenvectors of CMA-ES' covariance matrix $\mathbf{C}$ through multiplication by $((\sigma^2\mathbf{C})^{-1/2})^{\top})$. Finally, the surrogate model is build using these transformed points. Naturally, the points for prediction of the model are transformed in the same way to ensure prediction with respect to the same base vectors.

3 Doubly Trained Evolution Control for the S-CMA-ES

In this section, an alternative to S-CMA-ES, called the *Doubly Trained* S-CMA-ES (DTS-CMA-ES), will be described. It uses not only model-predicted values of sampled points, but also their variances. Therefore, models capable to provide both values for each point have to be employed, in particular Gaussian processes [17] or random forests [2].

The DTS-CMA-ES differs from the S-CMA-ES through using *doubly trained evolution control* instead of the generation evolution control in Step 5 of Algorithm 1. The doubly trained evolution control is described in Algorithm 4 as follows: First, the values $\hat{y}$ and variances s^2 of CMA-ES sampled points are predicted by the surrogate model which is previously trained using the points evaluated by the original fitness function from previous generations. Second, the points are sorted according to the values of some uncertainty criterion $\mathcal{C}$ based on predicted $\hat{y}$ and s^2. Third, the n_{orig} most uncertain points are evaluated by the original fitness function. Next, the model is retrained using the points (chosen similarly to S-CMA-ES) evaluated by the original fitness function including the n_{orig} points from the previous step. Eventually, denoting λ as the population size, the $\lambda - n_{\text{orig}}$ points function values are predicted by the retrained model, and returned to the original S-CMA-ES to compute new parameters. Note that training the new model in step 1 differs from using the model from the previous generation since it uses updated CMA-ES state variables σ, $\mathbf{m}$ and $\mathbf{C}$.

3.1 Uncertainty Criteria

The following criteria $\mathcal{C}$, which determine the points for evaluation by the original fitness function, can be used in the DTS-CMA-ES (Algorithm 4).

Algorithm 3. S-CMA-ES trainModel

Input: σ, $\mathbf{m}$, $\mathbf{C}$, $\mathcal{A}$, r (maximal distance between training points and $\mathbf{m}$), n_{MIN}, n_{MAX} (min. and max. number of points for training)

1: $(\mathbf{X}_{\text{tr}}, \mathbf{y}_{\text{tr}}) \leftarrow \{(\mathbf{x}, y) \in \mathcal{A} \mid (\mathbf{m}-\mathbf{x})^\top (\sigma^2 \mathbf{C})^{-1/2} (\mathbf{m}-\mathbf{x}) \leq r\}$
2: **if** $|\mathbf{X}_{\text{tr}}| \geq n_{\text{MIN}}$ **then**
3: $\quad (\mathbf{X}_{\text{tr}}, \mathbf{y}_{\text{tr}}) \leftarrow$ choose n_{MAX} points by k-NN if $|\mathbf{X}_{\text{tr}}| > n_{\text{MAX}}$
4: $\quad \mathbf{X}_{\text{tr}} \leftarrow \{((\sigma^2 \mathbf{C})^{-1/2})^\top \mathbf{x}_{\text{tr}} | \mathbf{x}_{\text{tr}} \in \mathbf{X}_{\text{tr}}\}$
5: $\quad f_{\mathcal{M}} \leftarrow$ buildModel$(\mathbf{X}_{\text{tr}}, \mathbf{y}_{\text{tr}})$
6: **else**
7: $\quad f_{\mathcal{M}} \leftarrow \emptyset$
8: **end if**

Output: $f_{\mathcal{M}}$

Algorithm 4. Doubly Trained evolutionControl in DTS-CMA-ES

Input: λ, σ, $\mathbf{m}$, $\mathbf{C}$, f, $\mathcal{A}$, $\{\mathbf{x}_k\}_{k=1}^{\lambda}$ (CMA-ES sampled population), $\mathcal{C}$ (uncertainty criterion), n_{orig} (number of original-evaluated points), r, n_{MIN}, n_{MAX}

$f_{\mathcal{M}} \leftarrow \text{trainModel}(\mathcal{A})$
$(\hat{y}_k, s_k^2) \leftarrow f_{\mathcal{M}}(\mathbf{x}_k)$ $\quad k \in \mathcal{I} := \{1, \ldots, \lambda\}$ $\quad$ {*model evaluation*}
$c_k \leftarrow \mathcal{C}(\hat{y}_k, s_k^2)$ $\quad k \in \mathcal{I}$ $\quad$ {*criterion evaluation*}
$\{c_{k_i}\}_{i=1}^{\lambda} \leftarrow \text{sort}\{c_k\}_{k=1}^{\lambda}$
$\mathcal{I}_{\text{orig}} = \{k_i \in \mathcal{I} \,|\, \{c_{k_i}\}_{i=1}^{n_{\text{orig}}}\}$
$y_k \leftarrow f(\mathbf{x}_k)$ $\quad k \in \mathcal{I}_{\text{orig}}$ $\quad$ {*fitness evaluation*}
$\mathcal{A} = \mathcal{A} \cup \{(\mathbf{x}_k, y_k)\}_{k \in \mathcal{I}_{\text{orig}}}$
$f_{\mathcal{M}} \leftarrow \text{trainModel}(\mathcal{A}, \sigma, \mathbf{m}, \mathbf{C}, r, n_{\text{MIN}}, n_{\text{MAX}})$
$y_k \leftarrow f_{\mathcal{M}}(\mathbf{x}_k)$ $\quad k \in \mathcal{I} \setminus \mathcal{I}_{\text{orig}}$ $\quad$ {*model evaluation*}

Output: $(y_k)_{k \in \mathcal{I}}$, $\mathcal{A}$

Variance. The variance s^2 of model-predicted function values $\hat{y}$:

$$\mathcal{C}_{s^2} = s^2. \tag{1}$$

The larger the variance, the higher the uncertainty of the predicted fitness.

Lower Confidence Bound (LCB). The lower confidence bound has been proposed in [14]:

$$\mathcal{C}_{\text{LCB}} = \hat{y} - 2s^2. \tag{2}$$

The points with lower values of the LCB criterion are considered more interesting for evaluation by the original fitness function than the points with higher values.

Probability of Improvement (PoI). The probability of improvement with respect to a given target $T \le y_{\min}$ can be expressed as follows:

$$\mathcal{C}_{\text{PoI}} = P(f(\mathbf{x}) \le T | y_1, \ldots, y_n) = \phi\left(\frac{T - \hat{y}}{s}\right), \tag{3}$$

where ϕ denotes the distribution function of $\mathcal{N}(0,1)$ and $y_{\min}$ is the minimum value found so far.

Expected Improvement (EI). The expected improvement is described by [10]:

$$\mathcal{C}_{\text{EI}} = E((y_{\min} - f(\mathbf{x}))I(f(\mathbf{x}) < y_{\min}) | y_1, \ldots, y_n), \tag{4}$$

where

$$I(f(\mathbf{x}) < y_{\min}) = \begin{cases} 1 & f(\mathbf{x}) < y_{\min} \\ 0 & f(\mathbf{x}) \ge y_{\min}. \end{cases} \tag{5}$$

Similarly, $\mathcal{C}_{\text{EI}}$ can be expressed as [10]:

$$\mathcal{C}_{\text{EI}} = (y_{\min} - \hat{y})\,\phi\left(\frac{y_{\min} - \hat{y}}{s}\right) + s\varphi\left(\frac{y_{\min} - \hat{y}}{s}\right), \tag{6}$$

where φ denotes the density of $\mathcal{N}(0,1)$.

4 Experimental Evaluation

We compared the performance of the DTS-CMA-ES to the original CMA-ES [5], two surrogate-model-based CMA-ES algorithms, the S-CMA-ES [1] and the BIPOP-^{s*}ACM-ES-k [13], and the SMAC algorithm [8] on the set of all 24 noiseless functions from the COCO/BBOB framework [3,4].

4.1 Experimental Setup

The considered algorithms were compared in dimensions $D = 2, 3, 5, 10$, and 20 using the standard BBOB settings, i.e. on 15 different function instances. The BBOB stopping criteria were reaching the distance from the function optimum $\Delta f_T = 10^{-8}$ and expending maximal number of evaluations per dimension (FE/D) which we have set to 100 due to our interest in expensive optimization where very few evaluations are available [6]. The parameters of the compared algorithms are summarized in the following paragraphs.

We have employed the original CMA-ES in its IPOP-CMA-ES version (Matlab code v. 3.61) with the following parameters: number of restarts = 4, IncPopSize = 2, $\sigma_{start} = \frac{8}{3}$, $\lambda = 4 + \lfloor 3 \log D \rfloor$. The remaining parameters were left default.

Loshchilov's ^{s*}ACM-ES-k was used in its bi-population version published in [13]. The BIPOP-^{s*}ACM-ES-k results have been downloaded from the BBOB results data archive[1] in its GECCO 2013 settings.

Gaussian processes (GP) have been employed in the S-CMA-ES as surrogate models for $g_m = 5$ model-evaluated generations. The distance r (see Algorithm 1) has been set to 10. The covariance function $K^{\nu=5/2}_{\text{Matérn}}$ with starting values $(\sigma_n^2, l, \sigma_f^2) = \log(0.01, 2, 0.5)$ were used for the GP model (see [1] for the details).

As opposed to [16], all the function values were normalized to zero mean and unit variance before training surrogate models in order to increase numerical accuracy. The CMA-ES parameter values have been set the same as in the original CMA-ES. All other settings were left default [1].

GP have been also employed in SMAC-BBOB [7], the continuous optimization version of the SMAC. The SMAC results were downloaded from the BBOB results data archive[2].

The DTS-CMA-ES was tested with multiple settings of parameters. First, all the uncertainty criteria from Sect. 3.1 (s^2, LCB, EI, PoI) were compared using $\lambda = 4 + \lfloor 3 \log D \rfloor$ and $n_{\text{orig}} = \lceil 0.1\lambda \rceil$ (see Algorithm 4) to find the most suitable one. For the remaining investigations, two different population sizes $\lambda_{\text{1pop}} = 4 + \lfloor 3 \log D \rfloor$ and $\lambda_{\text{2pop}} = 8 + \lfloor 6 \log D \rfloor$ and four n_{orig} values $\lceil 0.05\lambda \rceil$, $\lceil 0.1\lambda \rceil$, $\lceil 0.2\lambda \rceil$, $\lceil 0.4\lambda \rceil$ were used for comparison. The CMA-ES parameters, the distance r, and the GP model have been taken over from the S-CMA-ES.

---

[1] http://coco.gforge.inria.fr/data-archive/2013/BIPOP-saACM-k_loshchilov_noiseless.tgz.

[2] http://coco.gforge.inria.fr/data-archive/2013/SMAC-BBOB_hutter_noiseless.tgz.

4.2 Results

We have compared the performances of DTS-CMA-ES for four different uncertainty criteria described in Sect. 3.1. The results aggregated through the full set of benchmark functions show that the different criteria exhibit very similar convergence rate. However, the usage of $\mathcal{C}_{s^2}$ leads to a slightly better performances on most of BBOB functions, especially in 20D, and $\mathcal{C}_{\mathrm{EI}}$ performs the best on f16, f22, and f23 (if aggregated through dimensions).

Figure 1 presents comparison of DTS-CMA-ES employing criteria $\mathcal{C}_{s^2}$, $\mathcal{C}_{\mathrm{LCB}}$, $\mathcal{C}_{\mathrm{EI}}$, $\mathcal{C}_{\mathrm{PoI}}$ with $n_{\mathrm{orig}} = \lceil 0.1\lambda \rceil$ in 5D and 20D. Let Δ_f be the minimal distance found from the function optimum for the considered number of fitness function evaluations. The graphs depict a scaled logarithm of Δ_f depending on FE/D. Since all the algorithms ran for each function and dimension on 15 independent instances, only the empirical medians Δ_f^{med} over those 15 runs of Δ_f were taken for further processing. The scaled logarithms of Δ_f^{med} are calculated as

$$\Delta_f^{\log} = \frac{\log \Delta_f^{\mathrm{med}} - \Delta_f^{\mathrm{MIN}}}{\Delta_f^{\mathrm{MAX}} - \Delta_f^{\mathrm{MIN}}} \log_{10}\left(1/10^{-8}\right) + \log_{10} 10^{-8}$$

where Δ_f^{MIN} (Δ_f^{MAX}) is the minimum (maximum) $\log \Delta_f^{\mathrm{med}}$ found among all the compared algorithms for the particular function f and dimension D between 0 and 100 FE/D. Afterwards, graphs of $\Delta_f^{\log}$ can be aggregated across arbitrary number of functions and dimensions. Values in presented graphs are averages of $\Delta_f^{\log}$ across all 24 functions. More detailed results can be found on authors' webpage[3].

The graphs in Fig. 2 summarize the performance of four different n_{orig} values and two population sizes $\lambda_{1\mathrm{pop}}$ and $\lambda_{2\mathrm{pop}}$. This and all the following experiments use the criterion $\mathcal{C}_{s^2}$ which performed best in the first set of experiments. We found that the lower the n_{orig}, the better the performance is observed. Moreover, testing showed overall best performance of $n_{\mathrm{orig}} = \lceil 0.05\lambda_{2\mathrm{pop}} \rceil$, which is in 2D, 3D, 5D equal to 1, and in 10D and 20D is equal to 2.

Table 1 illustrates the counts of the 1st ranks of the compared algorithms according to the lowest achieved Δ_f^{med} for 20, 40, and 80 FE/D respectively. These counts are for different dimensions summed across all 24 functions.

As can be seen in Fig. 3, DTS-CMA-ES provides the best average results among the tested algorithms during the middle part of the optimization process, i.e. between 30 and 80 FE/D. The SMAC excels at the very beginning of optimization progress (up to ca. 15 FE/D), and starting from ca. 80–130 FE/D (depending on dimension), the fastest converging algorithm is the ^{s*}ACM-ES-k.

The new algorithm demonstrates speed-up compared to the S-CMA-ES with the exception of f1; however, it still has problem with few multimodal functions (f17–f20). It can be interpreted as premature convergence in local optima.

---

[3] http://bajeluk.matfyz.cz/scmaes/ppsn2016/.

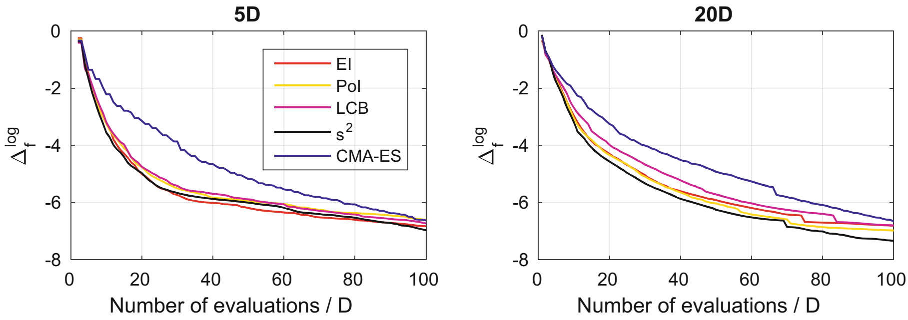

Fig. 1. Criterion comparison in 5D and 20D

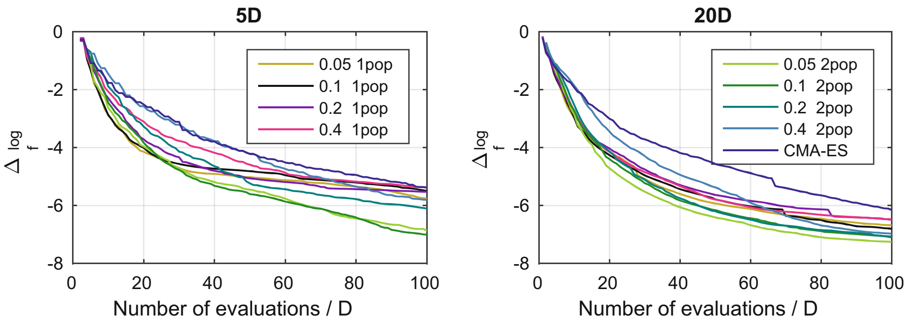

Fig. 2. Population-size and n_{orig} comparison in 5D and 20D

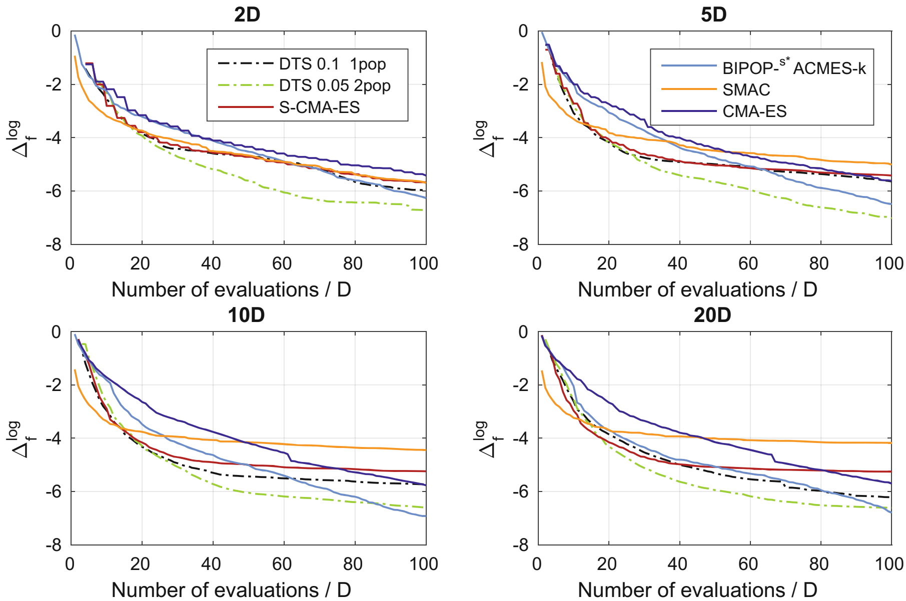

Fig. 3. Algorithm comparison in 2, 5, 10 and 20D

Table 1. Counts of the 1st ranks from 24 benchmark functions according to the lowest achieved Δ_f^{med} for different FE/D = {20, 40, 80} and dimensions D = {2, 3, 5, 10, 20}. Ties of the 1st ranks are counted for all respective algorithms. The ties often occure when $\Delta f_T = 10^{-8}$ is reached (mostly on f1 and f5).

FE/D	2D			3D			5D			10D			20D			$\sum$		
	20	40	80	20	40	80	20	40	80	20	40	80	20	40	80	20	40	80
DTS 0.1 1pop	6	3	2	**13**	6	3	**10**	4	3	**10**	7	3	2	4	5	**41**	24	16
DTS 0.05 2pop	**8**	**17**	**13**	7	**11**	**11**	9	**14**	**13**	6	**13**	**11**	**11**	**10**	8	**41**	**65**	**56**
S-CMA-ES	5	4	3	1	4	5	5	4	2	7	3	2	9	6	3	27	21	15
BIPOP-^{s*}ACM-ES-k	2	1	7	3	3	6	1	2	4	1	2	8	2	4	**9**	9	12	34
SMAC	5	4	4	4	4	5	3	4	5	4	2	2	3	4	3	19	18	19
CMA-ES	1	2	3	1	3	2	0	3	5	0	1	6	0	0	4	2	9	20

5 Conclusion and Future Work

This article presents a new version of the surrogate-based optimization algorithm S-CMA-ES. It further investigates the possibility to use surrogate models based on Gaussian processes in connection with the state-of-the-art black-box optimization algorithm CMA-ES. This improved algorithm introduces an additional model training within one generation, which shows a faster convergence to the global optima on many benchmark functions, independently of dimensions.

The choice of uncertainty criteria was not found as crucial in the speed of DTS-CMA-ES convergence. Furthermore, the comparison shows that the lower numbers of reevaluated points in each generation can lead to higher performance of the algorithm. We found that new approach usually reduces the number of necessary evaluations in expensive optimization more than other compared surrogate-model-based versions of the CMA-ES, namely BIPOP-^{s*}ACM-ES-k and S-CMA-ES, and except very early stages of the exploitation even more than SMAC-BBOB algorithm.

The main perspective of improving DTS-CMA-ES is to make the number of reevaluated points online adjustable, which should lead to more precise control of exploitation and facilitate escaping from the local optima. Another perspective is to additionally investigate different properties of surrogate models for better utilization of uncertainty criteria.

Acknowledgements. This work was supported by the Grant Agency of the Czech Technical University in Prague with its grant No. SGS14/205/OHK4/3T/14 by the Czech Health Research Council project NV15-33250A, by the project "National Institute of Mental Health (NIMH-CZ)", grant number ED2.1.00/03.0078 and the European Regional Development Fund, and by the project Nr.LO1611 with a financial support from the MEYS under the NPU I program. Further, access to computing and storage facilities owned by parties and projects contributing to the National Grid Infrastructure MetaCentrum, provided under the programme "Projects of Large Infrastructure for Research, Development, and Innovations" (LM2010005), is greatly appreciated.

References

1. Bajer, L., Pitra, Z., Holeňa, M.: Benchmarking Gaussian processes and random forests surrogate models on the BBOB noiseless testbed. In: Proceedings of the 17th GECCO Conference Companion. ACM, Madrid, July 2015
2. Breiman, L.: Classification and Regression Trees. Chapman & Hall/CRC, Boca Raton (1984)
3. Hansen, N., Auger, A., Finck, S., Ros, R.: Real-parameter black-boxoptimization benchmarking 2012: experimental setup. Technical report, INRIA (2012)
4. Hansen, N., Finck, S., Ros, R., Auger, A.: Real-parameter black-box optimization benchmarking 2009: noiseless functions definitions. Technical report RR-6829, INRIA (2009). Updated February 2010
5. Hansen, N.: The CMA evolution strategy a comparing review. In: Lozano, J.A., Larrañaga, P., Inza, I., Bengoetxea, E. (eds.) Towards a New Evolutionary Computation. Studies in Fuzziness and Soft Computing, vol. 192, pp. 75–102. Springer, Heidelberg (2006)
6. Holeňa, M., Linke, D., Bajer, L.: Surrogate modeling in the evolutionary optimization of catalytic materials. In: Soule, T. (ed.) Proceedings of the 14th GECCO, pp. 1095–1102. ACM, New York, Philadelphia (2012)
7. Hutter, F., Hoos, H., Leyton-Brown, K.: An evaluation of sequential model-based optimization for expensive blackbox functions. In: Proceedings of the 15th Annual Conference Companion on Genetic and Evolutionary Computation, GECCO 2013 Companion, pp. 1209–1216. ACM, New York (2013)
8. Hutter, F., Hoos, H.H., Leyton-Brown, K.: Sequential model-based optimization for general algorithm configuration. In: Coello, C.A.C. (ed.) LION 2011. LNCS, vol. 6683, pp. 507–523. Springer, Heidelberg (2011)
9. Jin, Y.: A comprehensive survey of fitness approximation in evolutionary computation. Soft Comput. **9**(1), 3–12 (2005)
10. Jones, D.R.: A taxonomy of global optimization methods based on response surfaces. J. Glob. Optim. **21**(4), 345–383 (2001)
11. Loshchilov, I., Schoenauer, M., Sebag, M.: Intensive surrogate model exploitation in self-adaptive surrogate-assisted CMA-ES (saACM-ES). In: Genetic and Evolutionary Computation Conference (GECCO), pp. 439–446. ACM Press, July 2013
12. Loshchilov, I., Schoenauer, M., Sebag, M.: Self-adaptive surrogate-assisted covariance matrix adaptation evolution strategy. In: Proceedings of the 14th GECCO, GECCO 2012, pp. 321–328. ACM, New York (2012)
13. Loshchilov, I., Schoenauer, M., Sebag, M.: BI-population CMA-ES algorithms with surrogate models and line searches. In: Genetic and Evolutionary Computation Conference (GECCO Companion), pp. 1177–1184. ACM Press, July 2013
14. Lu, J., Li, B., Jin, Y.: An evolution strategy assisted by an ensemble of local Gaussian process models. In: Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation, GECCO 2013, pp. 447–454. ACM, New York (2013)
15. Ong, Y.S., Nair, P.B., Keane, A.J.: Evolutionary optimization of computationally expensive problems via surrogate modeling. AIAA J. **41**(4), 687–696 (2003)
16. Pitra, Z., Bajer, L., Holeňa, M.: Comparing SVM, Gaussian process and random forest surrogate models for the CMA-ES. In: ITAT 2015: Information Technologies - Applications and Theory, pp. 186–193. CreateSpace Independent Publishing Platform, North Charleston (2015)
17. Rasmussen, C.E., Williams, C.K.I.: Gaussian Processes for Machine Learning. Adaptative Computation and Machine Learning Series. MIT Press, Cambridge (2006)