Black-box optimization benchmarking of IPOP-saACM-ES on the BBOB-2012 noisy testbed
Ilya Loshchilov, Marc Schoenauer, Michèle Sebag

To cite this version:

HAL Id: hal-00690543
https://inria.hal.science/hal-00690543
Submitted on 23 Apr 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Black-box optimization benchmarking of IPOP-saACM-ES on the BBOB-2012 noisy testbed

Ilya Loshchilov
TAO, INRIA Saclay
U. Paris Sud, F-91405 Orsay

Marc Schoenauer
TAO, INRIA Saclay
U. Paris Sud, F-91405 Orsay

Michèle Sebag
CNRS, LRI UMR 8623
U. Paris Sud, F-91405 Orsay

ABSTRACT

In this paper, we study the performance of IPOP-*aACM-ES, recently proposed self-adaptive surrogate-assisted Covariance Matrix Adaptation Evolution Strategy. The algorithm was tested using restarts till a total number of function evaluations of $10^6 D$ was reached, where $D$ is the dimension of the function search space. The experiments show that the surrogate model control allows IPOP-*aACM-ES to be as robust as the original IPOP-aCMA-ES and outperforms the latter by a factor from 2 to 3 on 6 benchmark problems with moderate noise. On 15 out of 30 benchmark problems in dimension 20, IPOP-*aACM-ES exceeds the records observed during BBOB-2009 and BBOB-2010.

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

Algorithms

Keywords

Benchmarking, black-box optimization, evolution strategy, CMA-ES, self-adaptation, surrogate models, ranking support vector machine, surrogate-assisted optimization

1. INTRODUCTION

When dealing with expensive optimization objectives, the surrogate-assisted approaches proceed by learning a surrogate model of the objective, and using this surrogate to reduce the number of computations of the objective function in various ways.

Many surrogate modelling approaches have been used within Evolution Strategies (ESs) and Covariance Matrix Adaptation Evolution Strategy (CMA-ES): Radial Basis Functions network [8], Gaussian Processes [17], Artificial Neural Network [3], Support Vector Regression [12], Local-Weighted Regression [11, 1], Ranking Support Vector Machine (Ranking SVM) [16, 13, 9]. In most cases, the surrogate model is used as a filter (to select $\lambda_{pre}$ promising pre-children) and/or to estimate the fitness of some individuals in the current population. An example of surrogate-assisted CMA-ES with filtering strategy can be found in [13].

A well-known drawback of surrogate-assisted optimization is a strong dependence of the results on hyper-parameters used to build the surrogate model. Some optimal settings of hyper-parameters for a specific set of problems can be found by offline tuning, however for a new problem they are unknown in the black-box scenario. Moreover, the optimal hyper-parameters may dynamically change during the optimization of the function.

Motivated by this open issues, new self-adapted surrogate-assisted *aACM-ES algorithm have been proposed combining surrogate-assisted optimization of the expensive function and online optimization of the surrogate model hyper-parameters [14].

2. THE ALGORITHMS

2.1 The $(\mu/\mu_w, \lambda)$-CMA-ES

In each iteration $t$, $(\mu/\mu_w, \lambda)$-CMA-ES [6] samples $\lambda$ new solutions $x_i \in \mathbb{R}^D$, where $i = 1, \ldots, \lambda$, and selects the best $\mu$ among them. These $\mu$ points update the distribution of parameters of the algorithm to increase the probability of successful steps in iteration $t + 1$. The sampling is defined by a multi-variate normal distribution, $\mathcal{N}(m^t, \sigma^2 C^t)$, with current mean of distribution $m^t$, $D \times D$ covariance matrix $C^t$ and step-size $\sigma^t$.

The active version of the CMA-ES proposed in [7, 10] introduces a weighted negative update of the covariance matrix taking into account the information about $\lambda - \mu$ worst points as well as about $\mu$ best ones. The new version improves CMA-ES on 9 out of 12 tested unimodal functions by a factor up to 2, and the advantages are more pronounced in larger dimension. While the new update scheme does not guarantee the positive-definiteness of the covariance matrix, it can be numerically controlled [7]. Since in our study we do not observe any negative effects of this issue, we will use aCMA-ES, the active version of the CMA-ES, for comparison with the surrogate-assisted algorithms.

2.2 The *aACM-ES

The *aACM-ES [14] is the surrogate-assisted version of

Author's version.

GECCO ’12, July 7–11, 2012, Philadelphia, USA.
the \((\mu/\mu_w, \lambda)\)-CMA-ES, where the surrogate model is used periodically instead of the expensive function for direct
optimization. The use of Ranking SVM allows to preserve the property of CMA-ES of invariance with respect to rank-
preserving transformation of the fitness function. The property of invariance with respect to the orthogonal transforma-
tion of the search space is preserved thanks to the definition of the kernel function by the covariance matrix, adapted
during the search.

In \(\ast\)ACM-ES we perform the following surrogate-assisted optimization loop: we optimize the surrogate model \(f\) for \(\hat{n}\) generations by the CMA-ES, then we continue and optimize the expensive function \(f(x)\) for one generation. To adjust the number of generations \(\hat{n}\) for the next time, the model error can be computed as a fraction of incorrectly predicted comparison relations that we observe, when we compare the ranking of the last \(\lambda\) evaluated points according to \(f(x)\) and \(\hat{f}\). The \(\ast\)ACM-ES uses the generation of the CMA-ES as a black-box procedure, and it has been shown in [14], that the improvement of the CMA-ES from passive to active ver-
sion (aCMA-ES) leads to a comparable improvement of its surrogate-assisted versions (\(\ast\)ACM-ES and \(\ast\)aCMA-ES).

The main novelty of the \(\ast\)ACM-ES is the online optimization of the surrogate model hyper-parameters during the op-
timization of the fitness function. The algorithm performs the search in the space of model hyper-parameters, generating \(\lambda_{hyp}\) surrogate models in each iteration. The fitness of
the model can be measured as a prediction error of the ranking on the last \(\lambda\) evaluated points. This allows the user
to define only the range of hyper-parameters and let algorithm to find the most suitable values for the current iteration \(\hat{f}\).

The detailed description of \(\ast\)ACM-ES is given in [14].

2.3 The Benchmarked Algorithms

For benchmarking we consider \(\ast\)ACM-ES in IPOP restart scenario (IPOP-\(\ast\)-aCMA-ES) using default parameters and
termination criteria as given in [7] and [14]. The only one parameter of the surrogate part of IPOP-\(\ast\)-aCMA-ES differ-
ent from the default one is the index of generation \(g_{\text{start}}\) when we start to use the surrogate model. For noisy case
we set \(g_{\text{start}} = 5(i_{\text{restart}} + 1)\) instead of default \(g_{\text{start}} = 10\). We found that sometimes it makes sense to postpone the surrogate-assisted search if several restarts \(i_{\text{restart}}\) were performed.

3. RESULTS

Results from experiments according to [4] on the bench-
mk functions given in [2, 5] are presented in Figures 3 and 4, and Figure 2. The expected running time (ERT),
used in the figures and table, depends on a given target function value, \(f_t = f_{\text{opt}} + \Delta f\), and is computed over all
relevant trials (on the first 15 instances) as the number of function evaluations executed during each trial while the
best function value did not reach \(f_t\), summed over all trials and divided by the number of trials that actually reached \(f_t\)
[4, 15]. Statistical significance is tested with the rank-
sum test for a given target \(\Delta f_t\) using, for each trial, either
the number of needed function evaluations to reach \(\Delta f_t\)
(inverted and multiplied by \(-1\)), or, if the target was not
reached, the best \(\Delta f\)-value achieved, measured only up to
the smallest number of overall function evaluations for any
unsuccessful trial under consideration if available. Tables 1 and
2 give the Expected Running Time (ERT) for targets

\[10^1, -1, -3, -5, -7\] divided by the best ERT obtained during
BBOB-2009 (given in the ERT\(_{\text{best}}\) row), respectively in 5-D
and 20-D. Bold entries correspond to the best (or 3-best if
there are more than 3 algorithms) values. The median num-
ber of conducted function evaluations is additionally given
in \(\text{italics}\), if \(\text{ERT}(10^{-7}) = \infty\). \#succ is the number of trials
that reached the final target \(f_{\text{opt}} + 10^{-8}\). Entries with the \(\downarrow\)
symbol are statistically significantly better (according to the
rank-sum test) compared to the best algorithm in BBOB-
2009, with \(p = 0.05\) or \(p = 10^{-k}\) where \(k > 1\) is the number
following the \(\downarrow\) symbol, with Bonferroni correction of 30.

The IPOP-\(\ast\)-aCMA-ES outperforms IPOP-\(\ast\)-aCMA-ES usu-
ally by a factor from 2 to 3 on functions with moderate
noisy. This is the case for Sphere \((f_{101}, f_{102}, f_{103})\) and Rosen-
brock \((f_{104}, f_{105}, f_{106})\) functions with Gaussian, Uniform
and Cauchy noise models. It seems that the moderate noise only
slightly affects the quality of the surrogate model and allows
the IPOP-aCMA-ES usu-
ally to have a speedup comparable to one of the noise-less case.

On most functions with severe noise the surrogate model
usually is not used \((\hat{n}\) oscillates around zero), because it
gives a very imprecise prediction of the expensive function.
On 20-dimensional \(f_{24}\) Schaffer function with Cauchy noise
IPOP-\(\ast\)-aCMA-ES requires about 7 times more functions
evaluations to reach the optimum with \(\Delta f_{\text{opt}} = 10^{-7}\) than
IPOP-aCMA-ES (see Fig. 2). However, according to Ta-
ble 2, the performance for \(\Delta f_{\text{opt}} = 10^{-3}\) is exactly
the same for both algorithms, therefore, we suppose that the loss of performance for \(\Delta f_{\text{opt}} = 10^{-7}\) can be explained by
some influence of surrogate-assisted search on restart condi-
tions. We also found that if use default coefficients \(c_1\) and \(c_2\)
for the covariance matrix update instead of noisy settings of
\(c_1/5\) and \(c_2/5\), then IPOP-\(\ast\)-aCMA-ES performs as well as
IPOP-aCMA-ES. However, the use of default coefficients
worsen the results of IPOP-aCMA-ES and IPOP-\(\ast\)-aCMA-
ES on other problems.

We also observe some loss in performance of \(f_{125}\) Griewank-
Rosenbrock and \(f_{248}\) Gallagher functions for \(d \leq 5\). For \(f_{125}\)
we observe the speedup for \(d = 5\) and loss for \(d = 20\), prob-
ably because of the same reasons as for \(f_{124}\).

The IPOP-\(\ast\)-aCMA-ES improves (sometimes insignificantly)
the records in dimension 20 on \(f_{101}, f_{102}, f_{103}, f_{104}, f_{105}, f_{106}, f_{107}, f_{109}, f_{112}, f_{114}, f_{117}, f_{118}, f_{120}, f_{122}, f_{123}, f_{127}, f_{129}\).

4. CPU TIMING EXPERIMENT

For the timing experiment the IPOP-\(\ast\)-aCMA-ES was run on
noiseless \(f_1, f_8, f_{10}\) and \(f_{15}\) functions without self-adaptation
of surrogate model hyper-parameters. The crucial hyper-
parameter for CPU time measurements, the number of train-
ing points was set \(N_{\text{training}} = [40 + 4D^{0.7}]\) as a function of

dimension \(D\) [14].

These experiments have been conducted on a single core
with 2.4 GHz under Windows XP using Matlab R2006a.

On uni-modal functions the time complexity of surrogate
model learning increases cubically in the search space di-

mension (see Fig. 1) and quadratically in the number of
training points. For small dimensions \((D < 10)\) the overall
time complexity increases super-linearly in the dimension.
The time complexity per function evaluation depends on
the population size, because one model is used to estimate
the ranking of all points of the population. This leads to a
smaller computational complexity on multi-modal functions,
e.g. \(f_{15}\) Rastrigin function, where the population becomes
much larger after several restarts. The time complexity on noisy functions is more similar to the one of Rastrigin function, because in both cases the large populations are used. 

The results presented here does not take into account the model hyper-parameters optimization, where $\lambda_{hyp}$ surrogate models should be build at each iteration, which leads to an increase of CPU time per function evaluation by a factor of $\lambda_{hyp}$. For IPOP-aACM-ES $\lambda_{hyp}$ was set to 20.

5. CONCLUSION

In this paper, we have compared the recently proposed self-adaptive surrogate-assisted IPOP-aACM-ES with the IPOP-aCMA-ES on noisy benchmark problems. The surrogate-assisted IPOP-aACM-ES algorithm outperforms the original IPOP-aCMA-ES by a factor from 2 to 3 on the functions with moderate noise, and usually performs not worse on other functions. The IPOP-aACM-ES algorithm improves the records on 15 out of 30 functions in dimension 20.

6. ACKNOWLEDGMENTS

The authors would like to acknowledge Anne Auger, Zyed Bouzarkouna, Nikolaus Hansen and Thomas P. Runarsson for their valuable discussions. This work was partially funded by FUI of System@tic Paris-Region ICT cluster through contract DGT 117 407 Complex Systems Design Lab (CSDL).

7. REFERENCES


Figure 2: Expected running time (ERT) divided by dimension for target function value $10^{-8}$ as $\log_{10}$ values versus dimension. Different symbols correspond to different algorithms given in legend of $f_{101}$ and $f_{130}$. Light symbols give the maximum number of function evaluations from all trials divided by the dimension. Horizontal lines give linear scaling, the slanted dotted lines give quadratic scaling. Legend: $\circ$: IPOP-aCMA, $\triangledown$: IPOP-saACM
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1e-1</td>
<td>1e-0</td>
<td>1e-1</td>
<td>1e-0</td>
<td>1e-1</td>
<td>1e-0</td>
<td>1e-1</td>
<td>1e-0</td>
<td>1e-1</td>
<td>1e-0</td>
</tr>
<tr>
<td>Δf</td>
<td>f101</td>
<td>1.1</td>
<td>3.0</td>
<td>3.0</td>
<td>3.0</td>
<td>3.0</td>
<td>3.0</td>
<td>3.0</td>
<td>3.0</td>
<td>3.0</td>
</tr>
<tr>
<td>IPOP-aC</td>
<td>3.0(1)</td>
<td>4.2(1)</td>
<td>4.2(1)</td>
<td>4.2(1)</td>
<td>4.2(1)</td>
<td>4.2(1)</td>
<td>4.2(1)</td>
<td>4.2(1)</td>
<td>4.2(1)</td>
<td>4.2(1)</td>
</tr>
<tr>
<td>IPOP-sa</td>
<td>3.0(1)</td>
<td>4.2(1)</td>
<td>4.2(1)</td>
<td>4.2(1)</td>
<td>4.2(1)</td>
<td>4.2(1)</td>
<td>4.2(1)</td>
<td>4.2(1)</td>
<td>4.2(1)</td>
<td>4.2(1)</td>
</tr>
<tr>
<td>Δf</td>
<td>f102</td>
<td>1.1</td>
<td>3.0</td>
<td>3.0</td>
<td>3.0</td>
<td>3.0</td>
<td>3.0</td>
<td>3.0</td>
<td>3.0</td>
<td>3.0</td>
</tr>
<tr>
<td>IPOP-aC</td>
<td>3.0(1)</td>
<td>4.2(1)</td>
<td>4.2(1)</td>
<td>4.2(1)</td>
<td>4.2(1)</td>
<td>4.2(1)</td>
<td>4.2(1)</td>
<td>4.2(1)</td>
<td>4.2(1)</td>
<td>4.2(1)</td>
</tr>
<tr>
<td>IPOP-sa</td>
<td>3.0(1)</td>
<td>4.2(1)</td>
<td>4.2(1)</td>
<td>4.2(1)</td>
<td>4.2(1)</td>
<td>4.2(1)</td>
<td>4.2(1)</td>
<td>4.2(1)</td>
<td>4.2(1)</td>
<td>4.2(1)</td>
</tr>
</tbody>
</table>

Table 1: Expected running time (ERT in number of function evaluations) divided by the respective best ERT.

- **#succ** indicates the number of trials that reached the final target $f_{opt} + 10^{-8}$.
- Best results are printed in bold.

- The central 80% range divided by two is given in brackets.
- The median number of conducted function evaluations is additionally given in italics, if ERT(10⁻⁷) = ∞.
- IPOP-aC has optimal performance.
| Table 2: Expected running time (ERT in number of function evaluations) divided by the respective best ERT measured during BBOB-2009 (given in the respective first row) for different $\Delta f$ values in dimension 20. The central 80% range divided by two is given in braces. The median number of conducted function evaluations is additionally given in italics, if $\text{ERT}(10^{-9}) = \infty$. #succ is the number of trials that reached the final target $f_{\text{opt}} + 10^{-8}$. Best results are printed in bold. |
Figure 3: Bootstrapped empirical cumulative distribution of the number of objective function evaluations divided by dimension (FEvals/D) for 50 targets in $10^{-8..2}$ for all functions and subgroups in 5-D. The “best 2009” line corresponds to the best ERT observed during BBOB 2009 for each single target.
Figure 4: Bootstrapped empirical cumulative distribution of the number of objective function evaluations divided by dimension (FEvals/D) for 50 targets in $10^{1-8.2}$ for all functions and subgroups in 20-D. The “best 2009” line corresponds to the best ERT observed during BBOB 2009 for each single target.