Testing the Impact of Parameter Tuning on a Variant of IPOP-CMA-ES with a Bounded Maximum Population Size on the Noiseless BBOB Testbed

Tianjun Liao
IRIDIA, CoDE, Université Libre de Bruxelles (ULB), Brussels, Belgium
tliao@ulb.ac.be

Thomas Stützle
IRIDIA, CoDE, Université Libre de Bruxelles (ULB), Brussels, Belgium
stuetzle@ulb.ac.be

ABSTRACT
In this paper, we experimentally explore the influence tuned parameter settings have on an IPOP-CMA-ES variant that uses a maximum bound on the population size. We followed our earlier work, where we exposed seven parameters that control parameters of IPOP-CMA-ES, and tune them by applying irace, an automatic algorithm configuration tool. A comparison of the tuned to the default settings on the BBOB benchmark shows that for difficult problems such as multi-modal functions with weak global structure, the tuned parameter settings can result in significant improvements over the default settings.

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

1. INTRODUCTION
Assigning appropriate values for the parameters of optimization algorithms is an important task [4]. Over the recent years, evidence has been given that the performance of many algorithms can be improved by using automatic algorithm configuration and tuning tools [1–3, 5, 10, 11, 17]. Many successful studies involve configuring discrete optimization algorithms [2, 5, 10], but also the tuning of continuous optimization algorithms has received some attention [3, 11–13, 16, 19, 20]. Even if automatic algorithm configuration tools sometimes do not result in very strong performance improvements, they have the advantage of reducing typically the time necessary for tuning parameter values and they help in reducing the bias in algorithm comparisons.

In this paper, we tune the parameters of a IPOP-CMA-ES variant for which we have presented computational results in another to BBOB 2013 [14]. That variant, to which we refer as IP-10DDr, uses a bound on the maximum population size to be used in IPOP-CMA-ES and, once this bound is reached, it restarts the scheme for the variation of the population size at the initial default value used by IPOP-CMA-ES. Here, we experimentally explore the influence of the tuned parameter settings in the performance of IP-10DDr. We found that for the difficult problems such as multi-modal functions with weak global structure, the tuned parameter settings can result in significant improvements over the default ones.

2. EXPERIMENTAL PROCEDURE
For tuning IP-10DDr, we considered seven parameters related to the above mentioned default settings, following previous work we have presented in [13]. The seven default settings of IP-10DDr are as follows. The initial population size is $\lambda = 4 + \lfloor 3 \ln(D) \rfloor$, where $D$ is the number of dimensions of the function to be optimized. The number of selected search points in the parent population is $\mu = \lfloor 0.5 \lambda \rfloor$. The initial step-size is $\sigma^{(0)} = 0.5(B - A)$, where $[A, B]^D$ is the initial search interval. At each restart, the population size is multiplied by a factor of two. Restarts occur if the stopping criterion is met. The three parameters are $stopTol\text{Hist}(=10^{-20})$, $stopTol\text{Fun}(=10^{-12})$ and $stop\text{TolX}(=10^{-13})$; they refer to the range of the improvement of the best objective function values in the last $10 + \lfloor 30D/\lambda \rfloor$ generations, all function values of the recent generation, and the standard deviation of the normal distribution in all coordinates, respectively.

The tuned parameters are given in Table 1. The first four parameters are actually used in a formula to compute some internal parameters of IP-10DDr and the remaining three are used to define the termination of CMA-ES. The first three columns of Table 1 give the parameters we use, the formula where they are used, the range that we considered for tuning, and the bias in algorithm comparisons.

As tuner we use irace [15], a publicly available implementation of the automatic configuration method Iterated F-Race [2]. The budget of each run of irace is set to 5 000 runs of IP-10DDr. We consider a separation between tun-
ing and test sets. The training instances are the same as those used in [16], which are a subset of the functions in the SOCOb benchmark [9], with dimension $D \in [5, 40]$. The performance measure used for tuning is the error of the objective function value obtained by the tuned algorithm after 100 $\times$ $D$ function evaluations. It should be highlighted that we focus here on the first 100D function evaluations as the same results are re-used in the expensive optimization scenario at BBOB 2013 and we wanted to avoid a re-tuning. (In fact, the exploration of the impact different tuning setups have on the performance of the tuned IP-10DDr we leave for future work.) From this tuning we obtain a configuration of IP-10DDr we call henceforth texp. We also use the parameter settings we obtained in another paper [16], where IPOP-CMA-ES was tuned to improve its anytime behavior; the algorithm using these parameter settings is labeled as tany (Table 1). The default parameter settings are labeled def in this paper.

3. RESULTS

Results from experiments according to [7] on the benchmark functions given in [6,8] are presented in Figures 1, 2 and 3 and in Tables 2 and 3. The expected running time (ERT), in the figures and table, depends on a given target function value, $f_t = f_{opt} + \Delta f$, and is computed over all relevant trials 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 $f_t$ [7,18]. Statistical significance is tested with the rank-sum test for a given target $\Delta f_t$ ($10^{-8}$ as in Figure 1) 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.

In the experiments, we clearly observe that performance may vary with the parameter settings on at least some of the functions. On some functions, texp is slower in reaching optimal threshold than either tany or def, the most noteworthy example being $f_2$ ($D = 2, 3, 40$). However, texp performs better than tany or def on several multi-modal functions with weak global structure. For example, it reaches the optimal threshold in $f_{23}$ ($D = 20$) and $f_{24}$ ($D = 5, 20$) where tany and def cannot do so. tany reaches optimal threshold in $f_{23}$ ($D = 40$) and $f_{24}$ ($D = 40$) where texp and def cannot do so. We also clearly observe that texp uses fewer function evaluations to reach optimal threshold in $f_{21}$ ($D = 10, 20, 40$), $f_{22}$ ($D = 2, 3, 5, 10$), $f_{24}$ ($D = 2, 3$) than tany and def. Only on $f_{23}$ ($D = 2, 3, 5$), def obviously uses fewer function evaluations to reach optimal threshold than tany and texp. The overview of the overall results in Figure 2 confirms the observation above: texp performs better than tany or def especially on the weakly structured multi-modal functions.

4. CPU TIMING EXPERIMENT

The texp and tany were run on $f_8$ until at least 30 seconds have passed. These experiment were conducted with Intel Xeon E5410 (2.33 GHz) on Linux (kernel 2.6.9 - 78.0.22). The results of texp were 1.6E$-05$, 2.4E$-05$, 6.2E$-06$, 9.7E$-06$, 1.5E$-05$ and 5.8E$-05$ seconds per function evaluation in dimensions 2, 3, 5, 10, 20, and 40, respectively. The results of tany were 1.8E$-05$, 2.4E$-05$, 1.7E$-05$, 7.5E$-06$, 1.3E$-05$ and 4.6E$-05$ seconds per function evaluation in dimensions 2, 3, 5, 10, 20, and 40, respectively.

5. CONCLUSIONS

In this paper we have examined the influence of tuning on the performance of IP-10DDr. When compared to default settings, further performance improvements could be observed on few difficult functions. These results suggest that it would be interesting to further explore the impact different tuning setups have. In fact, here we used a setting where we considered only very short runs for tuning and also a tuning target, where we try to obtain for a fixed number of function evaluations as good as possible solutions. Changing the tuning setup to minimize the number of function evaluations to target, the point of view actually followed in the BBOB benchmark analysis, we would expect further improved performance.

6. ACKNOWLEDGMENTS

The authors would like to thank the great and hard work of the BBOB team. This work was supported by the Meta-X project funded by the Scientific Research Directorate of the French Community of Belgium. Thomas Stützle acknowledges support from the Belgian F.R.S.-FNRS, of which he is a Research Associate. Tianjun Liao acknowledges a fellowship from the China Scholarship Council.

7. REFERENCES

Figure 1: Expected running time (ERT in number of $f$-evaluations) divided by dimension for target function value $10^{-8}$ as $\log_{10}$ values versus dimension. Different symbols correspond to different algorithms given in the legend of $f_1$ and $f_{24}$. Light symbols give the maximum number of function evaluations from the longest trial divided by dimension. Horizontal lines give linear scaling, slanted dotted lines give quadratic scaling. Black stars indicate statistically better result compared to all other algorithms with $p < 0.01$ and Bonferroni correction number of dimensions (six). Legend: ○:def, ▼:texp, ★:tany
Figure 2: 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 3: 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.
### 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 $\Delta$ values in dimension 5.

The central 80% range divided by three is given in braces. The median number of conducted function evaluations is additionally given in italics, if ERT($10^{-7}$) = $\infty$. #succ is the number of trials that reached the final target $f_{\text{opt}} + 10^{-8}$. Best results are printed in bold.

<table>
<thead>
<tr>
<th>$\Delta$</th>
<th>ERT</th>
<th>$\frac{t_{\text{exp}}}{t_{\text{any}}}$</th>
<th>#succ</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>1.0</td>
<td>1.00(1) 1.00(1) 1.00(1)</td>
<td>$\infty$</td>
</tr>
<tr>
<td>0.2</td>
<td>1.0</td>
<td>1.00(1) 1.00(1) 1.00(1)</td>
<td>$\infty$</td>
</tr>
<tr>
<td>0.3</td>
<td>1.0</td>
<td>1.00(1) 1.00(1) 1.00(1)</td>
<td>$\infty$</td>
</tr>
<tr>
<td>0.4</td>
<td>1.0</td>
<td>1.00(1) 1.00(1) 1.00(1)</td>
<td>$\infty$</td>
</tr>
<tr>
<td>0.5</td>
<td>1.0</td>
<td>1.00(1) 1.00(1) 1.00(1)</td>
<td>$\infty$</td>
</tr>
</tbody>
</table>


Table 3: 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 \( ERT(10^{-7}) = \infty \). #succ is the number of trials that reached the final target \( f_{opt} + 10^{-8} \). Best results are printed in bold.


