Tuning Parameters across Mixed Dimensional Instances: A Performance Scalability Study of Sep-G-CMA-ES

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

Marco A. Montes de Oca
IRIDIA, CoDE, Université Libre de Bruxelles, Brussels, Belgium
mmontes@ulb.ac.be

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

ABSTRACT

Sep-G-CMA-ES is a variant of G-CMA-ES with lower time complexity. In this paper, we evaluate the impact that various ways of tuning have on the performance of Sep-G-CMA-ES on scalable continuous benchmark functions. We have extracted seven parameters from Sep-G-CMA-ES and tuned them across training functions with different features using an automatic algorithm configuration tool called Iterated F-Race. The best performance of Sep-G-CMA-ES was obtained when it was tuned using functions of different dimensionality (a strategy that we call *mixed dimensional*). Our comparative study on scalable benchmark functions also shows that the default Sep-G-CMA-ES outperforms G-CMA-ES. Moreover, the tuned version of Sep-G-CMA-ES significantly improves over both G-CMA-ES and default Sep-G-CMA-ES.

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Learning—*Parameter learning*

General Terms

Algorithms

Keywords

Sep-G-CMA-ES, Large scale continuous optimization, Parameter tuning, Mixed dimensions

1. INTRODUCTION

The interest for solving large scale continuous optimization problems using modern continuous optimization algorithms is increasing. CMA-ES [6] is a state-of-the-art evolutionary algorithm for continuous optimization and G-CMA-ES [1], as an improved variant of CMA-ES, has shown an impressive performance in the real-parameter optimization special session of the 2005 IEEE Congress on Evolutionary Computation (CEC 2005) [3]. Due to the high computational cost of covariance matrix adaption, G-CMA-ES cannot be directly used to tackle high-dimensional problems. In fact, in a special issue of the Soft Computing journal [11] (throughout the rest of the paper, we will refer to this special issue as SOCO), G-CMA-ES was investigated for scalable functions with up to 1000 dimensions. However, due to its time complexity, G-CMA-ES was not applied to the 1000 dimensional benchmark functions.

Recently, Sep-G-CMA-ES, which is a modification of G-CMA-ES with lower time complexity, was proposed [14]. Sep-G-CMA-ES was benchmarked on BOBB-2009 functions [4] but only on functions of up to 20 dimensions [13]. In this paper, we test the performance of Sep-G-CMA-ES on large scale optimization problems. We extracted and tuned seven parameters of Sep-G-CMA-ES using an automatic algorithm configuration procedure, called Iterated F-Race [2], on training functions with different features. In the first stage of the experiment, the best performance of the tuned version of Sep-G-CMA-ES was obtained using mixed dimensional[1] instances, when we consider tuning Sep-G-CMA-ES on 5, 10, 20, 40 and mixed dimensional training instances. The second stage of the experiment on SOCO functions shows that default Sep-G-CMA-ES outperforms G-CMA-ES. Moreover, the tuned version of Sep-G-CMA-ES significantly improves over both G-CMA-ES and default Sep-G-CMA-ES.

The contributions of this paper are two. First, tuning across mixed dimensional instances is shown to be an effective parameter tuning strategy that improves the performance of Sep-G-CMA-ES on scalable continuous optimization functions. Second, we fill the gap left in SOCO by providing results of the tuned version of Sep-G-CMA-ES on the 19 SOCO functions.

2. THE SEP-G-CMA-ES ALGORITHM

Hansen et al. [5, 8, 14] have presented several ways to reduce the time complexity of CMA-ES. Sep-G-CMA-ES [14] is one of these proposals. In contrast to G-CMA-ES, the covariance matrix of Sep-G-CMA-ES is diagonal, thus the degrees of freedom in the covariance matrix are reduced from $\frac{n(n+1)}{2}$ to n, where n is the dimensionality of the search space. With such modification, the learning rate for covariance matrix can be increased. Its increased factor of covariance matrix learning rate is $\frac{n+1.5}{3}$ [13]. Sep-G-CMA-ES is not rotational invariant. Details of Sep-G-CMA-ES

[1]The mixed dimensions adopted in training instances are a random combination of 5, 10, 20 and 40 dimensions

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

Table 1: **The parameters and the factors to be tuned**

Parameters	Formulas	Factor	Range	Default
Pop size (λ)	$4 + \lfloor a \ln(n) \rfloor$	a	[1,10]	3
Parent size (μ)	$\lfloor \lambda / b \rfloor$	b	[1,5]	2
Init step size ($\sigma^{(0)}$)	$c(B - A)$	c	(0,1)	0.5
IPop factor(d)	d	d	[1,4]	2
stopTolFun	10^e	e	[-20,-6]	-15
stopTolFunHist	10^f	f	[-20,-6]	-20
stopTolX	10^g	g	[-20,-6]	-15

can be found in [14]. Instead of a partly time and space linear Sep-G-CMA-ES in [13], a full time and space linear Sep-G-CMA-ES is adopted in our present study. In our default setting of Sep-G-CMA-ES, the initial population size is $\lambda = 4 + \lfloor 3 \ln(n) \rfloor$. The number of selected search points in the population is $\mu = \lfloor 0.5\lambda \rfloor$. The initial step-size is $\sigma^{(0)} = 0.5(B - A)$, where $[A, B]^n$ is the initial search interval. The increasing factor of population multiplication of each restart is 2. Restarts occur if the stopping criterion is met. Please consult [1] for the definition of the three stopping criteria $stopTolFunHist$, $stopTolFun$ and $stopTolX$. We defined seven factors to be tuned. Each of these seven factors is used in a formula to actually compute the value of a parameter used by CMA-ES. The parameters, factors and function formulas are presented in Table 1.

3. EXPERIMENTAL STUDY

We used the 19 SOCO benchmark functions suite (functions labeled as $f_{soco}*$) over 50, 100, 200, 500, 1000 dimensions for our experiments. The detailed description of SOCO functions is available in [7]. We used the same termination conditions defined for SOCO. We report error values defined as $f(\boldsymbol{x}) - f(\boldsymbol{x}^*)$, where $\boldsymbol{x}$ is a candidate solution and $\boldsymbol{x}^*$ is the optimal solution. Error values lower than 10^{-14} are clamped to 10^{-14}. Our analysis is based on average errors.

3.1 Parameter tuning

Iterated F-race [2, 10] is used to automatically tune parameters in our study. In the automatic parameter tuning process, SOCO functions with different features were randomly sampled as training instances [9, 12]. The maximum experimental budget is set to 5000 runs of Sep-G-CMA-ES. The number of function evaluations of each run is equal to $5000 \times n$, where n is the problem's dimension. We used the default setting for Iterated F-race [2]. In the first stage of the experiment, all the parameter configurations tuned on 5, 10, 20, 40 and mixed dimensional training instances are used to compare their impact on the performance of Sep-G-CMA-ES on large scale functions. In the second stage of the experiment, the parameters tuned on mixed dimensional training instances are used. The tuned parameters are presented in Table 2.

3.2 Experimental results and comparison

The average errors obtained on each of the 19 SOCO benchmark functions of each dimensionality was used to evaluate the five different set of training instances. The statistical analysis using one sided Wilcoxon matched-pairs signed-rank test shows that mixed dimensional instance tuning strategy significantly outperforms tuning on fixed dimensional problems in the 13 of the total 20 cases at 0.05 α-level and in 17 of total 20 cases at 0.1 α-level. In the second stage of the experiment, our comparative study also shows that tuning across mixed dimensional training instances is an effective way to do parameter configuration for large scale optimization problems.

Table 2: The results of the factors tuned on different dimensional and mixed dimensional training instances

Extracted factor	Tune ($D = 5$)	Tune ($D = 10$)	Tune ($D = 20$)	Tune ($D = 40$)	Tune ($D = Mixed$)
a	6.525	9.254	9.065	9.188	9.426
b	2.405	3.548	3.132	3.182	1.995
c	0.7693	0.2803	0.937	0.8273	0.4216
d	1.579	3.282	3.503	2.721	3.415
e	-13.92	-14.64	-17.03	-15.59	-16.84
f	-13.13	-19.54	-17.75	-16.75	-15.10
g	-14.08	-18.91	-18.5	-18.72	-16.21

Figure 1 and 2 show correlation plots that illustrate the relative performance for all pairs of G-CMA-ES, default Sep-G-CMA-ES and the tuned version of Sep-G-CMA-ES on dimensions 50, 100, 200, 500 and 1000, respectively. As seen from their statistical comparison, the results indicate that default Sep-G-CMA-ES outperforms G-CMA-ES. Moreover, the tuned version of Sep-G-CMA-ES significantly improves over both G-CMA-ES and default Sep-G-CMA-ES.

Next, we compared both the default and the tuned version of Sep-G-CMA-ES with the 16 algorithms featured in SOCO.[2] The box-plots of Figure 3 for 500 and 1000 dimensional functions show that the performance of Sep-G-CMA-ES is not very good. The box-plots of the comparison on other smaller dimensions are available in `http://iridia.ulb.ac.be/supp/IridiaSupp2011-012/`.

For further proving the effectiveness of the mixed dimensional tuning strategy and clarifying the issue of the inferior performance of Sep-G-CMA-ES, we tuned the parameters of Sep-G-CMA-ES on 200 dimensional SOCO functions several times and tested this "over-tuned" Sep-G-CMA-ESs on the same functions. We found the "over-tuned" performance on 200 dimensions were similar to the performance of the mixed small dimensional tuning strategy. On the one hand, the similar performance indicates that mixed small dimensional tuning strategy can effectively be used to tune parameters for large scale optimization problems with a smaller computational budget than directly tuning on high dimensionality. On the other hand, we can conclude that Sep-G-CMA-ES is not competitive for large scale benchmark functions is due to the optimization mechanism itself, rather than due to parameter configuration problem, since overtuning even can not make its performance competitive.

4. CONCLUSIONS AND FUTURE WORK

In this paper, we evaluate the impact that five different tuning strategies have on the performance of Sep-G-CMA-ES on large scale continuous benchmark problems. Considering different compositions of the training sets for tuning, the best performance of the tuned version of Sep-G-CMA-ES was obtained using mixed dimensional problems. Based on tuning across mixed dimensional instances, we benchmarked the tuned version of Sep-G-CMA-ES on SOCO

[2]For information about these 16 algorithms please go to `http://sci2s.ugr.es/eamhco/CFP.php`

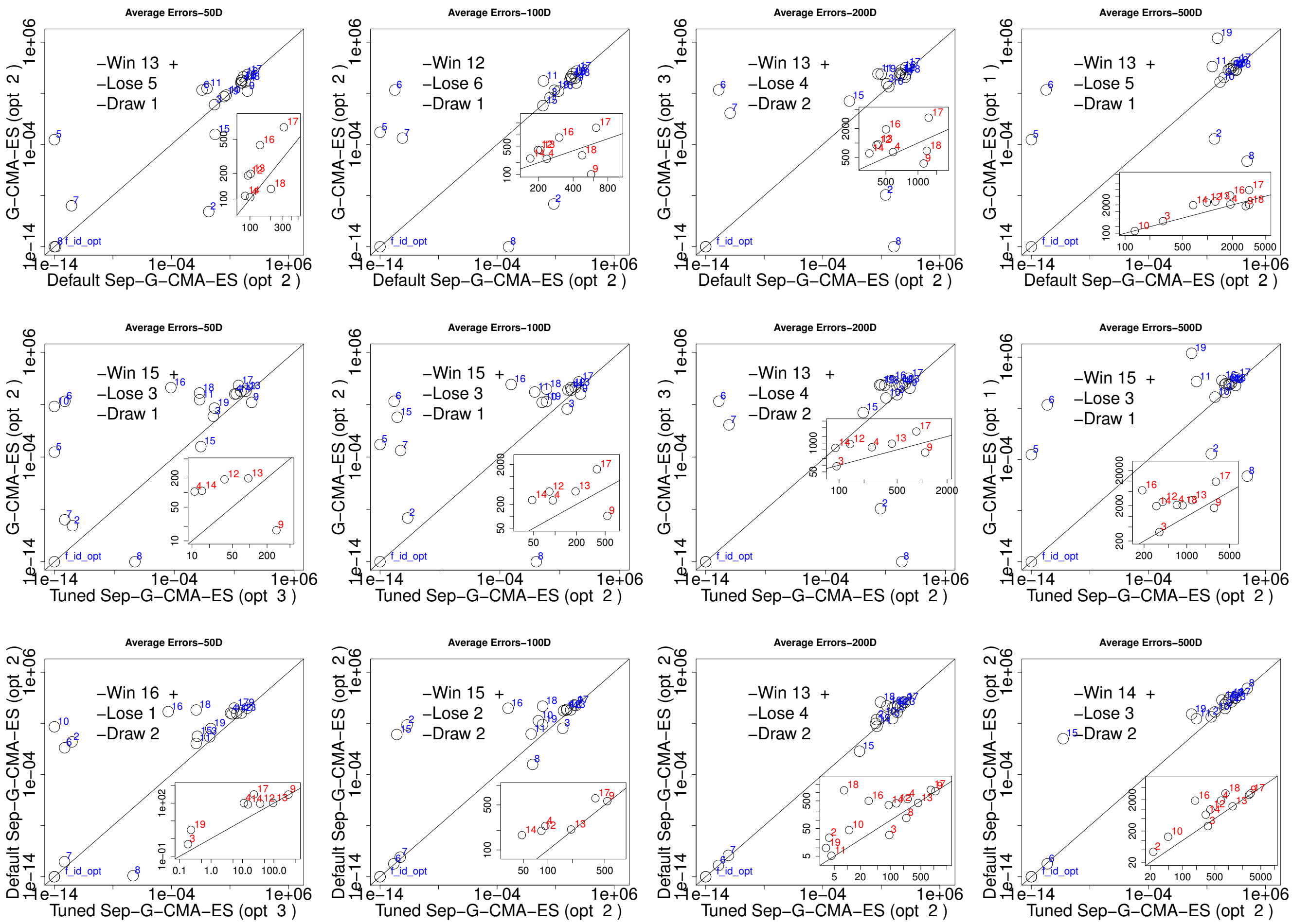


Figure 1: Correlation plots for all pairs of G-CMA-ES, default Sep-G-CMA-ES and the tuned version of Sep-G-CMA-ES on dimensions 50, 100, 200 and 500 respectively. Each point represents the average error value obtained by either of the two algorithms compared. A point on the upper triangle delimited by the diagonal indicates better performance for the algorithm on the x-axis; a point on the lower right triangle indicates better performance for the algorithm on the y-axis. The number labeled beside some outstanding point represent an index of corresponding function. The comparison is conducted based on average errors value and the comparison results of the algorithm on the x-axis are presented in form of -win, -draw, -lose, respectively. We marked with a + symbol those cases in which there is a statistically significant difference at the 0.05 α-level between the algorithms. The number of opt on the axises shows the number of optima obtained by the corresponding algorithm.

large scale functions and compensated for the unavailable results of G-CMA-ES. A comparative study indicates that default Sep-G-CMA-ES outperforms G-CMA-ES. Moreover, the tuned version of Sep-G-CMA-ES significantly improves over both G-CMA-ES and default Sep-G-CMA-ES. Therefore, we conclude that tuning on mixed dimensional training instances with different search landscapes is an effective way to handle parameters configuration for large scale continuous optimization problems. In our future work, the fixed vs. mixed dimensional tuning strategy for large scale optimization problems will be studied in more detail.

5. ACKNOWLEDGMENTS

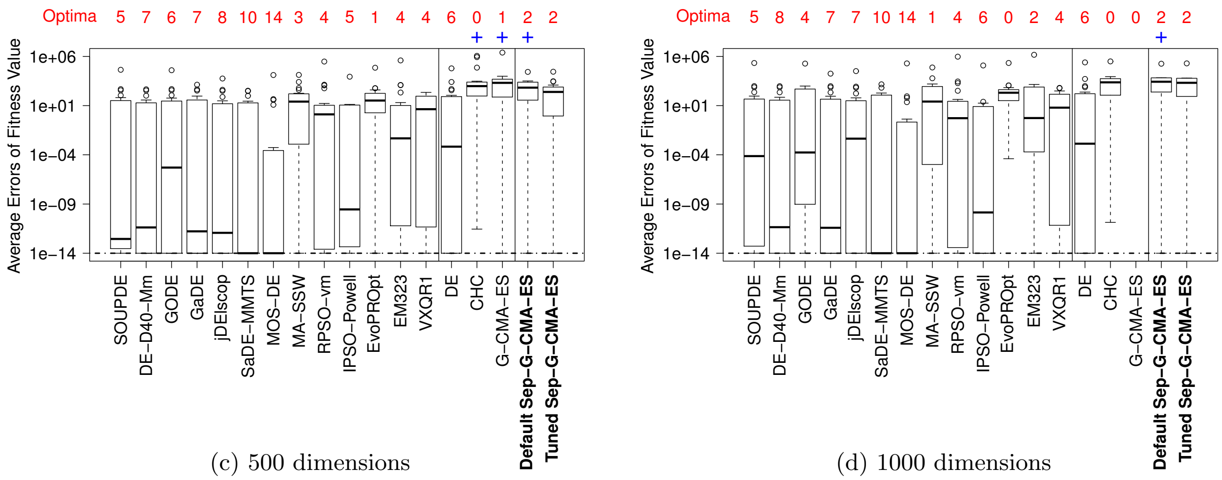
This work was supported by the Meta-X project funded by the Scientific Research Directorate of the French Community of Belgium. Thomas Stützle acknowledges support from the Belgian F.R.S.-FNRS, of which he is a Research Associate.

6. REFERENCES

[1] A. Auger and N. Hansen. A restart CMA evolution strategy with increasing population size. In *CEC 2005*, pages 1769–1776, Piscataway, NJ, 2005. IEEE Press.

[2] M. Birattari, Z. Yuan, P. Balaprakash, and T. Stützle. F-Race and iterated F-Race: An overview. *Experimental Methods for the Analysis of Optimization Algorithms*, pages 311–336, 2010. Springer, Berlin, Germany.

[3] N. Hansen. Compilation of results on the 2005 CEC benchmark function set. 2006. http://www.lri.fr/ hansen/cec2005compareresults.pdf.



(c) 500 dimensions

(d) 1000 dimensions

Figure 3: Box-plots show the distribution of the average errors obtained on the 19 SOCO benchmark functions on 500 and 1000 dimensions. On the left part are shown the results of 13 algorithms published in SOCO, in the middle part those of the three reference algorithms, and on the right part those of the default Sep-G-CMA-ES and the tuned version of Sep-G-CMA-ES. We marked with a + symbol those cases in which there is a statistically significant difference at the 0.05 α-level with respect to the tuned version of Sep-G-CMA-ES (in favor of the tuned version of Sep-G-CMA-ES). The line at the bottom of each plot represents the 0-threshold (10^{-14}). The numbers on top of a box-plot denotes the number of solution below the 0-threshold found by the corresponding algorithm.

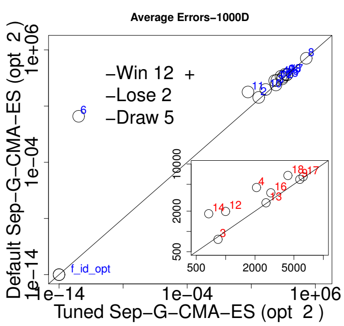


Figure 2: The correlation plot of 1000 dimensions between Tuned-Sep-G-CMA-ES and Default-Sep-G-CMA-ES

[4] N. Hansen, S. Finck, R. Ros, and A. Auger. Real-parameter black-box optimization benchmarking 2009: Noiseless functions definitions. 2009. Technical Report RR-6829, INRIA.

[5] N. Hansen, S. Muller, and P. Koumoutsakos. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation. *Evolutionary Computation*, 11(1):1–18, 2003.

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

[7] F. Herrera, M. Lozano, and D. Molina. Test suite for the special issue of soft computing on scalability of evolutionary algorithms and other metaheuristics for large scale continuous optimization problems, 2010. URL: `http://sci2s.ugr.es/eamhco/updated-functions1-19.pdf`.

[8] C. Igel, T. Suttorp, and N. Hansen. A computational efficient covariance matrix update and a (1+ 1)-CMA for evolution strategies. In *GECCO 2006*, pages 453–460, New York, 2006. ACM Press.

[9] T. Liao, M. A. Montes de Oca., D. Aydın, T. Stützle, and M. Dorigo. An Incremental Ant Colony Algorithm with Local Search for Continuous Optimization. In *GECCO 2011*, New York, 2011. ACM Press. accepted.

[10] M. López-Ibáñez, J. Dubois-Lacoste, T. Stützle, and M. Birattari. The `irace` package, iterated race for automatic algorithm configuration. Technical Report TR/IRIDIA/2011-004, IRIDIA, Université Libre de Bruxelles, 2011.

[11] M. Lozano, D. Molina, and F. Herrera. Editorial: Scalability of evolutionary algorithms and other metaheuristics for large-scale continuous optimization problems. *Soft Computing*, 2011. In Press. DOI: 10.1007/s00500-010-0639-2

[12] M. A. Montes de Oca, D. Aydın, and T. Stützle. An incremental particle swarm for large-scale optimization problems: An example of tuning-in-the-loop (re)design of optimization algorithms. *Soft Computing*, 2011. In press. DOI: 10.1007/s00500-010-0649-0

[13] R. Ros. Benchmarking sep-CMA-ES on the BBOB-2009 noisy testbed. In *GECCO 2009*, pages 2441–2446, New York, 2009. ACM Press.

[14] R. Ros and N. Hansen. A Simple Modification in CMA-ES Achieving Linear Time and Space Complexity. In *PPSN 2008*, pages 296–305. Springer-Verlag, 2008.