Investigation of step-size adaptation methods for CMA-ES based on population midpoint fitness

Jarosław Arabas and Eryk Warchulski

Warsaw University of Technology, Institute of Computer Science, Poland
jarabas@elka.pw.edu.pl, ewarchul@mion.elka.pw.edu.pl

Abstract. *Abstract to be compacted: JA*
Covariance Matrix Adaptation Evolution Strategy (CMA-ES) is the basic algorithm of many top-performing methods that have been the winners of black box competitions, such as BBOB or CEC. The effectiveness of CMA-ES is occupied by a considerable computational effort that is needed to perform matrix operations, including the factorization of the inverse matrix that is needed to implement the cumulative step adaptation method (CSA) used to adapt the step-size multiplier σ.

Therefore, a line of research is aimed at simplification or substitution matrix operations. In effect, several alternative methods to adapt σ have been introduced in the earlier literature.

In this paper we introduce and investigate three different new methods to adapt σ. All investigated methods are based on the comparison of fitness values between the population midpoint and the offspring. All methods reveal geometric convergence for the quadratic fitness function.

In the first two methods, σ is increased when the success ratio exceeds 1/5, and is decreased when the success ratio falls below 1/5. The success ratio in each iteration is defined as a fraction of the number of points generated in that iteration whose fitness value exceeds the fitness of the weighted mean (for the first method) or of the simple arithmetic mean (for the second method) of points from the previous iteration.

In the third version, σ is increased when the fitness of the current iteration midpoint is located below a certain percentile of the fitness values in that population.

We performed tests to compare the quality of results obtained by CMA-ES coupled with several different rules to control σ. These rules included three investigated methods, the standard CSA rule, and the median success rule (MSR) which was introduced in the earlier literature. The tests were performed using the CEC'2017 benchmark set.

According to the results of tests, CSA allows for the best performance of CMA-ES. Among other step-size control techniques, the third introduced method based on the percentile analysis of the midpoint yielded consistently better results than all other methods.

Keywords: CMA-ES, 1/5 success rate rule, CSA

1 Convergence of CMA-ES with varions step-size adaptation

1.1 Compared methods

In this section we want to investigate rate of convergence and computing overhead of proposed methods. As mentioned in previous sections we consider 5 methods with different σ adaptations rule:

1. CMA-ES-CSA
2. CMA-ES-MSR
3. CMA-ES-EXPTH
4. CMA-ES-JA
5. CMA-ES-QUANT

Each method use default settings for basic parameters suggested by authors expect for population size λ. All methods generated $\lambda = 4N$ points in each generation. For CMA-ES-MSR to count K_{succ} of better points in current population than j-th best point of the previous population we set j as:

$$j = 0.3\lambda.$$

1.2 Fitness functions: quadratic, linear, gutter

To ilustrate and compare rate of convergence we use different fitness landscapes i.e. linear (1), sphere (2) and gutter function (3).

$$f_l(\mathbf{x}) = x_1 \tag{1}$$

$$f_s(\mathbf{x}) = \sum_{i=1}^{D} x_i^2 \tag{2}$$

$$f_g(\mathbf{x}) = x_1 + \sum_{i=2}^{D} x_i^2, \ \mathbf{x} \in \mathbb{R}^D \tag{3}$$

To assess the rate of convergence each function is treated in a minimization manner and we record fitness of the arthimetic mean and the best point in the population.
The curves depicted in the figures below for iteration t show the fitness of the best point and midpoint in the tth generation.
For each problem and algorithm, the mean point of population is initialized as:

$$\mathbf{m}_i = [100, \ldots, 100]^D.$$

1.3 Computing overhead: comparison of the number of fitness evaluations without and with adaptation of sigma

todo

1.4 Conclusions: convergence rate, computing overhead, benefits from using midpoint, search space dimension

todo

2 Benchmarking step-size adaptation methods on CEC'2013 and CEC'2017

2.1 Overview of CEC'2013 and CEC'2017

We tested performance of proposed methods using standardized set of single objective optimization problems i.e. CEC2017 [**?**] and CEC2013 [**?**]. Both of them are based on benchmark set created in 2005 [**?**].

Each problem in set is defined as follows

$$\min_{\mathbf{x}\in[-100,100]^D} f(\mathbf{x})$$

and should be considered as a black-box problem. One knows only dimensionality D, evaluation budget and value of global optimum. Note that number of benchmark functions and functions by itself in CEC2013 and CEC2017 vary.

CEC2013 has 28 functions which are dived into three groups: unimodal, multimodal and composition functions. Structure of CEC2017 is similar with the difference that functions are split into unimodal, multimodal, hybrid and composition functions. Hybrid and composition functions are composed of several multimodal functions which are defined in such a way that in various regions of the domain the dynamics of the objective function is dominated by different components of that composition. Thus, the optimization algorithm must be able to capture these regularities over the whole run.

Structures of benchmarks are presented in table below (1).

<table>
<tr><th>Function type</th><th>CEC2017</th><th>CEC2013</th></tr>
<tr><td>Unimodal</td><td>$f_1, \dots, f_3$</td><td>$f_1, \dots, f_5$</td></tr>
<tr><td>Multimodal</td><td>$f_4, \dots, f_{10}$</td><td rowspan="2">$f_6, \dots, f_{20}$</td></tr>
<tr><td>Hybrid</td><td>$f_{11}, \dots, f_{20}$</td></tr>
<tr><td>Composition</td><td>$f_{21}, \dots, f_{30}$</td><td>$f_{21}, \dots, f_{28}$</td></tr>
</table>

Table 1. Summary of CEC2017 and CEC2013.

We evaluated each algorithm with default settings for CEC benchmarks i.e. 51 independent repetitions for function, $10^4 \cdot D$ objective function evaluations and dimensionality $D = 10, 30, 50$.

2.2 Presentation of results using ECDF curves

Instead of using tabular form of results presentation suggested by authors of CEC competitions we decided to use ECDF (empirical cumulative distribution function) curve. ECDF curve were propesed by Hansen [?] as a convenient form of results aggregation. A single curve defines the average dynamics of the algorithm. More formally, let's suppose that one of the tested algorithms in k-th repetition of D-dimensional benchmark function $f \in \mathcal{F}$ recorded in generation t value $Q^f_{k,D,t}$ which is the difference between the best-so-far objective function value and the global optimum.
Consider next mapping from $Q^f_{k,D,t}$ to unit interval $[0,1]$:

$$q^f_{k,D,t} = \frac{\log\left(\frac{Q^f_{k,D,t}}{m^f_D}\right)}{\log\left(\frac{M^f_D}{m^f_D}\right)} \tag{4}$$

Where m^f_D and M^f_D are respectively minimal and maximal value achieved among all tested algorithms, repetitions and generations for given set of problems.

To obtain ECDF points for functions from $\mathcal{F}$ one has to aggregate above values as follows:

$$q^F_{D,t} = \frac{1}{K \cdot |F|} \sum_{f \in F} \sum_{k=1}^{K} q^f_{k,D,t} \tag{5}$$

where K is the number of repetitions.
The curve is constructed by plotting values of $q^F_{D,t}$ against the set of the fitness evaluation milestones based on fraction of given budget.

2.3 Results

We set the fractions as

$$\{0.01, 0.02, 0.03, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$$

3 Conclusions

1. which method is advisable **JA+EW**
2. how much computing we can spare by avoiding matrix operations **JA+EW**
3. future research directions **JA+EW**

bibliografia do uzupelnienia: EW