Modified Box Constraint Handling for the Covariance Matrix Adaptation Evolution Strategy

Naoki Sakamoto
Graduate School of Science and Technology
Shinshu University
4-17-1 Wakasato
Nagano 380-8553, Japan
16w2032k@shinshu-u.ac.jp

Youhei Akimoto
Faculty of Engineering
Shinshu University
4-17-1 Wakasato
Nagano 380-8553, Japan
y_akimoto@shinshu-u.ac.jp

ABSTRACT

We propose a modified box constraint handling technique for the covariance matrix adaptation evolution strategy (CMA-ES). The existing box constraint handling turns the box-constrained optimization problem into an unconstrained optimization by introducing an artificial fitness landscape, where a penalty function is added to the function values at the nearest feasible solutions. By adapting the penalty coefficients, that determine the sensitivity of constraints over the objective function value, it creates a reasonable virtual function landscape outside the feasible domain. In this paper, we address the issue of the original box constraint handling technique that it performs slow adaptation of the penalty coefficients when the objective function scales non-quadratically, in particular when the objective function scales exponentially. The optimization is then stagnated until reasonable penalty coefficients are achieved. It is due to a relatively long history of the dispersion measure of the objective function values and the adaptation of the penalty coefficients using the median of the history. In the proposed algorithm, we look at a recent subsequence of the history when the dispersion measures in the history differ significantly. The current dispersion of the objective values is then estimated using the median of the computed subsequence of the history. Experimental results reveal that the proposed algorithm can converges without stagnation on a function with exponential factor, where the original algorithm exhibits stagnation.

CCS CONCEPTS

•Mathematics of computing → Bio-inspired optimization; •Theory of computation → Bio-inspired optimization;

KEYWORDS

CMA-ES, box constraint handling, adaptive penalty

ACM Reference format:

Naoki Sakamoto and Youhei Akimoto. 2017. Modified Box Constraint Handling for the Covariance Matrix Adaptation Evolution Strategy. In *Proceedings of GECCO '17 Companion, Berlin, Germany, July 15-19, 2017,* 2 pages.
DOI: http://dx.doi.org/10.1145/3067695.3075986

GECCO '17 Companion, Berlin, Germany

DOI: http://dx.doi.org/10.1145/3067695.3075986

1 CMA-ES

The covariance matrix adaptation evolution strategy (CMA-ES) [1, 2] samples λ candidate solution $\boldsymbol{x}_i$, for $i = 1, \dots, \lambda$, from the multivariate normal distribution $\mathcal{N}(\boldsymbol{m}, \sigma^2\mathbf{C})$, where $\boldsymbol{m} \in \mathbb{R}^n$ is the mean vector, $\sigma > 0$ is the step-size, and $\mathbf{C} \in \mathbb{R}^{n\times n}$ is the covariance matrix. These distribution parameters are updated by using the candidate solutions and their ranking information. The CMA-ES repeats the following steps until a termination criterion is satisfied.

Step 1. Draw λ samples $\boldsymbol{x}_i$ independently from $\mathcal{N}(\boldsymbol{m}, \sigma^2\mathbf{C})$.

Step 2. Evaluate the candidate solutions $\boldsymbol{x}_i$ on the fitness function L, and sort them in the ascending order. The fitness function L is the objective function f if the problem is unconstrained.

Step 3. Update the distribution parameters $\boldsymbol{m}$, σ and $\mathbf{C}$ based on the candidate solutions and their ranking information.

2 BOX CONSTRAINT HANDLING

In the following, we consider the minimization of $f : \mathbb{R}^n \to \mathbb{R}$ under the box-constraint, $[\mathsf{LB}]_i \leqslant [\boldsymbol{x}]_i \leqslant [\mathsf{UB}]_i$ for $i = 1, \dots, n$. For simplicity of notation, we write the feasible domain as $\boldsymbol{x} \in [\mathsf{LB}, \mathsf{UB}]$. We do not require that the objective function is defined in the infeasible domain, i.e., $f(\boldsymbol{x})$ for $\boldsymbol{x} \notin [\mathsf{LB}, \mathsf{UB}]$ can be undefined.

The box constraint handling proposed in [3][1] creates an artificial fitness landscape, L, over the infeasible domain. To create an artificial fitness in the infeasible domain, the penalty, f_p, is added to the objective function value at the closest feasible point, $\boldsymbol{x}^{\text{feas}}$.

Artificial Fitness with Adaptive Penalty. The fitness of a solution is defined as follows. If the solution $\boldsymbol{x}$ is feasible, the fitness is equal to the objective function value, $L(\boldsymbol{x}) = f(\boldsymbol{x})$. If the solution is infeasible, the closest (in the Euclidean sense) feasible solution is first computed as follows

$$[\boldsymbol{x}^{\text{feas}}]_i = [\mathsf{LB}]_i \ \ (\text{if } [\boldsymbol{x}]_i < [\mathsf{LB}]_i), [\mathsf{UB}]_i \ \ (\text{if } [\mathsf{UB}]_i < [\boldsymbol{x}]_i), [\boldsymbol{x}]_i \ \ (\text{otherwise}). \quad (1)$$

Then, the penalty is computed as follows

$$f_p(\boldsymbol{x}) = \tfrac{1}{n} \textstyle\sum_{i=1}^n \gamma_i([\boldsymbol{x}]_i - [\boldsymbol{x}^{\text{feas}}]_i)^2 \ , \quad (2)$$

where γ_i ($i = 1, \dots, n$) are the penalty coefficients adapted by the box constraint handling algorithm. Then, the fitness of $\boldsymbol{x}$ is defined as the sum of the objective function value and the penalty, namely

$$L(\boldsymbol{x}) = f(\boldsymbol{x}^{\text{feas}}) + f_p(\boldsymbol{x}) \ . \quad (3)$$

Adaptation of the penalty coefficients. The penalty coefficients, $\boldsymbol{\gamma} = (\gamma_1, \dots, \gamma_n) \in \mathbb{R}^n$, are initialized as $\gamma_i = 0$, for $i = 1, \dots, n$. The following adaptation steps are performed every iteration after the sampling step in CMA-ES.

[1]In this paper, we apply the correction and minor modifications of the algorithm, described in the author version. See https://www.lri.fr/~hansen/TEC2009online.pdf

Step 1. Update IQR history Compute the normalized interquartile range (IQR) of the objective function values as

$$\delta f := \mathrm{IQR}(f(\boldsymbol{x}_1^{\mathrm{feas}}), \ldots, f(\boldsymbol{x}_\lambda^{\mathrm{feas}}))/(\sigma^2 \mathrm{Tr}(C)/n) \tag{4}$$

and keep it for $20 + \lfloor 3n/\lambda \rfloor$ iterations.

Step 2. Set penalty coefficients If $\boldsymbol{m}$ is infeasible and either the penalty coefficients were not set yet or $t = 2$, set

$$\gamma_i = 2 \cdot \delta_{\mathrm{fit}}, \quad \forall i = 1, \ldots, n \ , \tag{5}$$

where $\delta_{\mathrm{fit}} \in \mathbb{R}$ is the median of the history of δf.

Step 3. Update penalty coefficients For each i ($i = 1, \ldots, n$), update penalty coefficients according to the following.

(a) Increase If the i-th coordinate of the mean vector is infeasible, compute the distance to the closest boundary normalized by the standard deviation of i-th coordinate, namely,

$$\delta m_i := |[\boldsymbol{m}]_i - [\boldsymbol{m}^{\mathrm{feas}}]_i|/(\sigma\sqrt{C_{ii}}) \ , \tag{6}$$

where $\boldsymbol{m}^{\mathrm{feas}}$ is the feasible point closest to $\boldsymbol{m}$ that is computed by (1), and $C_{ii} \in \mathbb{R}$ is the i-th diagonal element of the covariance matrix. Then, increase each penalty coefficient according to

$$\gamma_i \leftarrow \gamma_i \cdot \exp((d_\gamma/2)\tanh(\max(0, \delta m_i - \delta_{\mathrm{th}})/3)) \ , \tag{7}$$

where we set $d_\gamma = \min(1, \mu_{\mathrm{w}}/10n)$ and $\delta_{\mathrm{th}} = 3 \cdot \max(1, \sqrt{n}/\mu_{\mathrm{w}})$.

(b) Decrease If $\gamma_i > 5\delta_{\mathrm{fit}}$, γ_i is decreased according to

$$\gamma_i \leftarrow \gamma_i \cdot \exp(-d_\gamma/3) \quad . \tag{8}$$

3 MODIFIED BOX CONSTRAINT HANDLING

Trimmed Median. The search efficiency sometimes deteriorates remarkably in the original box constraint handling (Orig-BCH). The cause of the deterioration is that the history of the normalized inter-quartile range (δf) is too long and the median δ_{fit} of the history not a reasonable estimate of the current δf. To estimate δ_{fit} more properly, we compute the **trim**med **med**ian (trimmed) which is the median of the last K iterations.

Let $\delta f_{[:]}$ be the history of δf in (4) of length $20+\lfloor 3n/\lambda \rfloor$. Let $\delta f_{[i]}$ denote the δf at the i-th to last iteration, i.e., $\delta f_{[1]}$ is the current δf (at iteration t), $\delta f_{[i]}$ is δf at iteration $t+1-i$. Let $\delta f_{[j:k]}$ denote the subsequence of the history from index j to index k. Let $\mathtt{med}_3$ be the median δf in the last 3 iterations, namely, $\mathtt{median}(\delta f_{[1:3]})$.

Let T be the length of the δf history, i.e., $T = \min(t, 20 + \lfloor 3n/\lambda \rfloor)$. If $T \leqslant 3$, the trimmed returns the median of the history, $\mathtt{trimmed}(\delta f_{[:]}) = \mathtt{median}(\delta f_{[1:T]})$. If $T > 3$, the trimmed is computed as follows. (1) Compute $\mathtt{med}_3$. (2) Find the maximum $K \leqslant T$ such that $|\ln(\delta f_{[k]}) - \ln(\mathtt{med}_3)| < \ln(5)$ for all $k \leq K$. (3) Then, $\mathtt{trimmed}(\delta f_{[:]}) = \mathtt{median}(\delta f_{[1:K]})$.

Modified Penalty Coefficients Reduction. The other reason of the deterioration in Orig-BCH is due to slow reduction of the penalty coefficients in (8). Note that once (8) is performed, it is kept performed until $\gamma_i < 5\delta_{\mathrm{th}}$. To speed up the reduction, we replace **Step 3(b)** of Orig-BCH with the following

$$\gamma_i \leftarrow \gamma_i \cdot \min(3 \cdot \mathtt{trimmed}(\delta f_{[:]})/(\tfrac{1}{n}\textstyle\sum_{j=1}^n \gamma_j), 1) \quad \text{for all } i. \tag{9}$$

If the mean of the penalty coefficients is equal to or greater than three times the $\mathtt{trimmed}(\delta f_{[:]})$, all the penalty coefficients are reduced by the same factor so that the mean of the penalty coefficients is three times the $\mathtt{trimmed}(\delta f_{[:]})$. **Step 3(b)** of Orig-BCH makes the scale between penalty coefficients close to 1, while (9) reduces all the penalty coefficients evenly so that it is possible to keep the scale between already learned penalty coefficients.

Table 1: Benchmark function suite.

Sphere	$f_{\mathrm{sph}}(\boldsymbol{x})$	$\sum_{i=1}^n [\boldsymbol{x}]_i^2$
Ellipsoid	$f_{\mathrm{ell}}(\boldsymbol{x})$	$\sum_{i=1}^n 10^{6\frac{i-1}{n-1}} [\boldsymbol{x}]_i^2$
TwoAxes	$f_{\mathrm{twoax}}(\boldsymbol{x})$	$\sum_{i=1}^n \begin{cases} 10^6 [\boldsymbol{x}]_i^2 & \text{if } i \text{ is even} \\ [\boldsymbol{x}]_i^2 & \text{otherwise} \end{cases}$
Exponential	$f_{\exp}(\boldsymbol{x})$	$\sum_{i=1}^n \begin{cases} \exp(20([\boldsymbol{x}_i] - 1)) & \text{if } [\boldsymbol{x}]_i > 1 \\ [\boldsymbol{x}]_i^2 & \text{otherwise} \end{cases}$

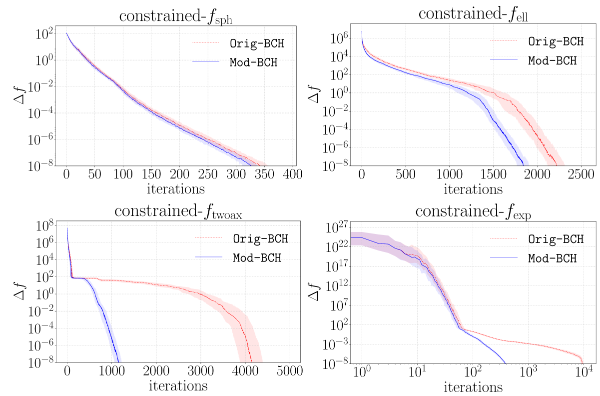

Figure 1: The median and 25%tile and 75%tiles of Δf over 100 trials on 20 dimensional f_{sph}, f_{ell}, f_{twoax} and $f_{\exp}$.

4 EXPERIMENTS

Table 1 summarizes the definitions of the test functions. The following box-constraint is considered $\mathtt{LB} = [-0.1, 0.1, \ldots, -0.1, 0.1]$, $\mathtt{UB} = \mathtt{LB} + [5, \ldots, 5]$. The global optimal solution is located at $\boldsymbol{x}^* = [0, 0.1, \ldots, 0, 0.1]$, that is, the optimum exists on the boundary of the feasible domain for the even-numbered coordinates, and it exists in the feasible domain for the odd-numbered coordinates. The global minimum is denoted by $f^* = f(\boldsymbol{x}^*)$. The initial step-size $\sigma^{(0)} = 1.25 = \frac{\mathtt{UB}-\mathtt{LB}}{4}$, mean vector $\boldsymbol{m}^{(0)} = [2.4, 2.6, \ldots, 2.4, 2.6] = \frac{\mathtt{UB}+\mathtt{LB}}{2}$, covariance matrix $\mathrm{C}^{(0)} = \boldsymbol{I}$ for all cases. Note that $f_{\exp}$ has the same function landscape as f_{sph} around the optimal solution. The function looks different only at the beginning of the search, though we will observe different behavior of Orig-BCH on these problems.

Figure 1 shows the median, the lower-quatile and the upper quartile over 100 trials of Orig-BCH and Modified Box Constraint Handling (Mod-BCH) on constrained f_{sph}, f_{ell}, f_{twoax} and $f_{\exp}$.

From the result of $f_{\exp}$, we can observe that the search efficiency deteriorates remarkably after the middle of the search in Orig-BCH. The reason for this is that the penalty coefficients decrease slowly. On the other hand, Mod-BCH converges at high speed. Furthermore, it exceeds the Orig-BCH in all functions.

REFERENCES

[1] Nikolaus Hansen and Anne Auger. 2014. Principled Design of Continuous Stochastic Search: From Theory to Practice. In *Theory and Principled Methods for the Design of Metaheuristics*, Y Borenstein and A Moraglio (Eds.). Springer.

[2] Nikolaus Hansen, Sibylle D. Muller, and Petros Koumoutsakos. 2003. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). *Evolutionary Computation* 11, 1 (2003), 1–18.

[3] Nikolaus Hansen, André S. P. Niederberger, Lino Guzzella, and Petros Koumoutsakos. 2009. A Method for Handling Uncertainty in Evolutionary Optimization With an Application to Feedback Control of Combustion. *IEEE Transactions on Evolutionary Computation* 13, 1 (2009), 180–197.