A Dynamic Partial Update for Covariance Matrix Adaptation

Hiroki Shimizu
The University of Tokyo
shimizu@tkl.iis.u-tokyo.ac.jp

Masashi Toyoda
Institute of Industrial Science, the University of Tokyo
toyoda@tkl.iis.u-tokyo.ac.jp

ABSTRACT

Tackling large-scale and ill-conditioned problems is demanding even for the covariance matrix adaptation evolution strategy (CMA-ES), which is a state-of-the-art algorithm for black-box optimization. The coordinate selection is a technique that mitigates the ill-conditionality of large-scale problems by updating parameters in partially selected coordinate spaces. This technique can be applied to various CMA-ES variants and improves their performance especially for ill-conditioned problems. However, it often fails to improve the performance of well-conditioned problems, because it is difficult to choose appropriate coordinate spaces according to the ill-conditionality of problems. We introduce a dynamic partial update method for coordinate selection to solve the above problem. We use the second-order partial derivatives of an objective function to estimate the condition number and select coordinates so that the condition number of each pair does not exceed the given allowable value. In this method, the number of clusters becomes to be small for well-conditioned problems and large for ill-conditioned cases. In particular, the selection does not execute if the condition number of the full space is less than the allowable value. We observe significant improvements in well-conditioned problems and comparable performances in ill-conditioned cases in numerical experiments.

CCS CONCEPTS

• **Computing methodologies** → **Artificial intelligence**; • **Theory of computation** → *Design and analysis of algorithms*;

KEYWORDS

Covariance Matrix Adaptation, Large scale optimization

ACM Reference Format:
Hiroki Shimizu and Masashi Toyoda. 2023. A Dynamic Partial Update for Covariance Matrix Adaptation. In *Genetic and Evolutionary Computation Conference Companion (GECCO '23 Companion), July 15–19, 2023, Lisbon, Portugal.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3583133.3596385

1 INTRODUCTION

Our world is full of optimization problems, *e.g.* searching for the shortest path [1], designing a high-speed train [9], designing a jet engine [12], and so on. However, most of these problems are too complex to formulate and solve analytically. Therefore, they have been treated as black-box optimization (BBO) problems. In the BBO scenario, it is impossible to know in advance all properties of objective functions such as the derivative, dependence among variables, ill-conditionality, and multimodality [5]. Hence, it is necessary to design algorithms that can adapt to the above properties.

The covariance matrix adaptation evolution strategy(CMA-ES) [4] is known as an algorithm that can efficiently search for solutions in black box optimization where the objective function is a continuous function. The CMA-ES generates individuals on the basis of a multivariate normal distribution, and searches for a solution through optimization of the variables of the multivariate normal distribution (a mean vector and a covariance matrix and a step-size). The off-diagonal components of the covariance matrix adapt to the dependence among variables, and the diagonal components adapt to the scale of each variable axis.

One of the issues of the CMA-ES is a computational cost for high-dimensional problems because it requires $O(d^3)$ time complexity and $O(d^2)$ space complexity for the input dimension d. To tackle this, various approaches, such as sep-CMA [10], VD-CMA [2], VkD-CMA [3], and LM-MA [8] have been proposed. In addition, it has been reported that when the objective function is high-dimensional and ill-conditioned, the adaptation of the covariance matrix to the ill-conditionality is inhibited and the function evaluation to reach a solution increases significantly [7]. The coordinate selection method[11] has been proposed as a solution to this problem. In this method, several coordinates are randomly selected for updating in each iteration, instead of updating all variables. Thereby, the number of conditions in the selected coordinate space is reduced from the original function. They achieved an improvement in performance for high-dimensional and ill-conditioned problems. However, they performed worse than the original algorithms on well-conditioned functions because selecting coordinates does not significantly reduce the number of conditions for the well-conditioned function, since the number of conditions can never be smaller than 1.

In this paper, we propose a dynamic partial update method for CMA variants to solve the problem of the coordinate selection. Our approach estimates the condition number by the finite difference method during the optimization process and selects coordinates with the constraint that the condition number of each cluster is less than an acceptable condition number (hyperparameter). Our approach prevents unnecessary coordinate selections in the optimization of well-condition functions, and furthermore, selects effective coordinates for objective functions whose condition number changes depending on the input values. Experimental results show that the proposed method improves on the coordinate selection method for well-conditioned functions, and achieves the same level of performance for ill-conditioned cases.

GECCO '23 Companion, July 15–19, 2023, Lisbon, Portugal

ACM ISBN 979-8-4007-0120-7/23/07…$15.00
https://doi.org/10.1145/3583133.3596385

2 RELATED WORK

2.1 The CMA-ES

The CMA-ES generates solution candidates with a multivariate normal distribution $\mathcal{N}(\boldsymbol{m}, \sigma^2 \boldsymbol{C})$ and searches for an optimal value by updating a mean vector $\boldsymbol{m}^{(t)}$, a covariance matrix $\boldsymbol{C}^{(t)}$ and a step-size σ. We describe a specific algorithm for minimizing an objective function $f(\boldsymbol{x}) \in \mathbb{R}$, $\boldsymbol{x} \in \mathbb{R}^d$ below. First, we initialize all variables. Determine the mean vector $\boldsymbol{m}^{(0)} \in \mathbb{R}^d$, the covariance matrix $\boldsymbol{C}^{(0)} \in \mathbb{R}^{d \times d}$, and the step-size $\sigma^{(0)} \in \mathbb{R}$ according to the search region, respectively. We also denote the respective evolution paths $\boldsymbol{p}_{\boldsymbol{c}}^{(0)} \in \mathbb{R}^d$ and $\boldsymbol{p}_{\boldsymbol{\sigma}}^{(0)} \in \mathbb{R}^d$ of the variance-covariance matrix and the step-size as $\mathbf{0}$. Then, the following steps are repeated until the predetermined termination conditions are met.

[Step 1.] Sample λ individuals $\boldsymbol{x}^{(t)} \in \mathbb{R}^{\lambda \times d}$ from a multivariate normal distribution $\mathcal{N}(\boldsymbol{m},\ \boldsymbol{C})$, $\boldsymbol{m} \in \mathbb{R}^d$, $\boldsymbol{C} \in \mathbb{R}^{d \times d}$ as

$$\boldsymbol{z}^{(t)} \sim \mathcal{N}(\mathbf{0},\ \boldsymbol{I}), \tag{1}$$

$$\boldsymbol{y}^{(t)} = \boldsymbol{z}^{(t)} \sqrt{\boldsymbol{C}^{(t)}}, \tag{2}$$

$$\boldsymbol{x}^{(t)} = \boldsymbol{m}^{(t)} + \sigma^{(t)} \boldsymbol{y}^{(t)}. \tag{3}$$

[Step 2.] For each individual of $\boldsymbol{x}^{(t)}$, compute objective values from $f(\boldsymbol{x_i})$, $(i = 1, ..., \lambda)$ and arrange $\boldsymbol{x}^{(t)}$, $\boldsymbol{y}^{(t)}$ and $\boldsymbol{z}^{(t)}$ in ascending order on the first axis.

[Step 3.] Compute $\boldsymbol{dy}^{(t)}$ and $\boldsymbol{dz}^{(t)} \in \mathbb{R}^d$ from the inner product of weight $\boldsymbol{w} \in \mathbb{R}^\lambda$ and, $\boldsymbol{y}^{(t)}$ and $\boldsymbol{z}^{(t)}$ as

$$\boldsymbol{dy}^{(t)} = \sum_i^\lambda w_i \boldsymbol{y}_{\boldsymbol{i}}^{(t)}, \tag{4}$$

$$\boldsymbol{dz}^{(t)} = \sum_i^\lambda w_i \boldsymbol{z}_{\boldsymbol{i}}^{(t)}. \tag{5}$$

Here $\boldsymbol{w}$ satisfies

$$\begin{aligned} 1 < \mu < \lambda, \qquad & w_1 \geq \cdots \geq w_\mu > 0, \\ w_{\mu+1}, \cdots, w_\lambda = 0, \qquad & \|\boldsymbol{w}\|_1 = 0. \end{aligned} \tag{6}$$

[Step 4.] Update the evolution paths as

$$h_\sigma^{(t+1)} = \begin{cases} 1 & \|\boldsymbol{p}_{\boldsymbol{\sigma}}^{(t+1)}\| < (1.4 + \frac{2}{n+1})\chi_d \\ 0 & else \end{cases}, \tag{7}$$

$$\boldsymbol{p}_{\boldsymbol{\sigma}}^{(t+1)} = (1 - c_\sigma)\boldsymbol{p}_{\boldsymbol{\sigma}}^{(t)} + \sqrt{c_\sigma(2 - c_\sigma)\mu_w}\boldsymbol{dz}, \tag{8}$$

$$\begin{aligned} \boldsymbol{p}_{\boldsymbol{c}}^{(t+1)} &= (1 - c_c)\boldsymbol{p}_{\boldsymbol{c}}^{(t)} \\ &+ h_\sigma^{(t+1)} \sqrt{c_c(2 - c_c)\mu_w}\boldsymbol{dy}. \end{aligned} \tag{9}$$

Here $c_\sigma \in \mathbb{R}$ and $c_c \in \mathbb{R}$ are the learning rate of evolution paths, and $\mu_w = \frac{1}{\|\boldsymbol{w}\|}$, $\chi_d = \mathbb{E}[\|\mathcal{N}(\mathbf{0}, \boldsymbol{I})\|] \simeq \sqrt{d}(1 - \frac{1}{4d} + \frac{1}{21d^2})$.

[Step 5.] Update the parameters as

$$\boldsymbol{m}^{(t+1)} = \boldsymbol{m}^{(t)} + \eta_m \sigma^{(t)} \boldsymbol{dy}^{(t)}, \tag{10}$$

$$\sigma^{(t+1)} = \sigma^{(t)} \exp\left(\frac{c_\sigma}{d_\sigma}\left(\frac{\|\boldsymbol{p}_{\boldsymbol{\sigma}}^{(t+1)}\|}{\chi_d} - 1\right)\right), \tag{11}$$

$$\begin{aligned} \boldsymbol{C}^{(t+1)} &= \boldsymbol{C}^{(t)} + \eta_{c_1}\left(OP(\boldsymbol{p}_{\boldsymbol{c}}^{(t+1)}) - \boldsymbol{C}^{(t)}\right) \\ &+ \eta_{c_\mu} \sum_i^\lambda w_i \left(OP(\boldsymbol{y}_{\boldsymbol{i}}^{(\boldsymbol{t})}) - \boldsymbol{C}^{(t)}\right), \end{aligned} \tag{12}$$

where $OP(\cdot) \in \mathbb{R}^{n \times n}$ denotes outer product of vectors. Here $\eta_m \in \mathbb{R}$, $\eta_{c_1} \in \mathbb{R}$ and $\eta_{c_\mu} \in \mathbb{R}$ are the learning rates of the mean $\boldsymbol{m}$, rank-one update and rank-μ update respectively, and $d_\sigma \in \mathbb{R}$ is the decay rate of the step-size.

By repeating [Step 1.] above to [Step 5.], the mean vector $\boldsymbol{m}$ converges to the solution, the step-size σ converges to 0, and the covariance matrix $\boldsymbol{C}$ converges to $\mathbf{0}$, then the multivariate normal distribution converges to the delta function and the optimal solution is obtained.

2.2 The Coordinate Selection

The coordinate selection is a technic to mitigate the condition number, which indicates the ill-conditionality of an objective function and is determined by a ratio of max. and min. eigenvalues of the Hessian of the objective function. In this method, they update only the parameters in a randomly selected coordinate space at each iteration and regard the variables in the coordinates that are not selected as numerical constants. Therefore, the condition number at the generation is recalculated from the Hessian of the selected coordinate space. In the Ellipsoid function case, the expected value of the condition number in a randomly selected coordinate space with 100 axes is approximately 7.7×10^5 which is about 25% less than the original value 10^6.

Despite the success of the ill-conditioned function, the coordinate selection degrades the performance of well-conditioned functions like the Sphere function and the Chain-Rosenbrock function. Selecting coordinates does not reduce the number of conditions for the well-conditioned function, since the number of conditions can never be smaller than 1 (e.g. the original condition number of the Sphere function is 1).

3 PROPOSAL

In this section, we propose the dynamic partial update for CMA variants by selecting coordinates on the basis of the approximated second derivative of the objective function. Coordinate selection can improve performance on ill-conditioned problems. However, for well-conditioned problems, coordinate selection reduces performance because it limits the number of variables to be updated during a single iteration. Therefore, we replace the strategy of randomly selecting coordinates at each iteration with a strategy of selecting such that the condition number in the selected coordinate space is less than an acceptable number.

3.1 Approximation of the condition number

We approximate the condition number of the objective function by the second-order partial derivative. While the condition number is originally obtained by the eigenvalues of the Hessian of the objective function, we approximate it by the absolute value of the second-order partial derivative obtained by the finite difference method as follows,

$$\frac{\partial^2 f}{\partial m_i^2} \approx \frac{f(m_1,...,m_i+h,...,m_d) - 2f(\boldsymbol{m}) + f(m_1,...,m_i-h,...,m_d)}{h^2}, \tag{13}$$

where m is the mean vector of a CMA and h is a small step-size, which is a different term from a variable of a CMA. The computational cost to obtain a value is a function evaluation of $3d$ for the input dimension d. This cost is not small enough to ignore if we

Table 1: Benchmark functions. R is an orthogonal matrix generated randomly.

Name	Definition
Sphere	$f_{sph}(\boldsymbol{x}) = \sum_{i=1}^{d} x_i^2$
Ellipsoid	$f_{ell}(\boldsymbol{x}) = \sum_{i=1}^{d} (1000^{\frac{i-1}{d-1}} x_i)^2$
Rot Ellipsoid	$f_{ellrot}(\boldsymbol{x}) = f_{ell}(\boldsymbol{Rx})$
Chain-Rosenbrock	$f_{cros}(\boldsymbol{x}) = \sum_{i=1}^{d-1} [10^2(x_i^2 - x_{i+1})^2 + (x_i - 1)^2]$
Star-Rosenbrock	$f_{sros}(\boldsymbol{x}) = \sum_{i=2}^{d} \left(100(x_1 - x_i^2)^2 + (1 - x_i)^2\right)$

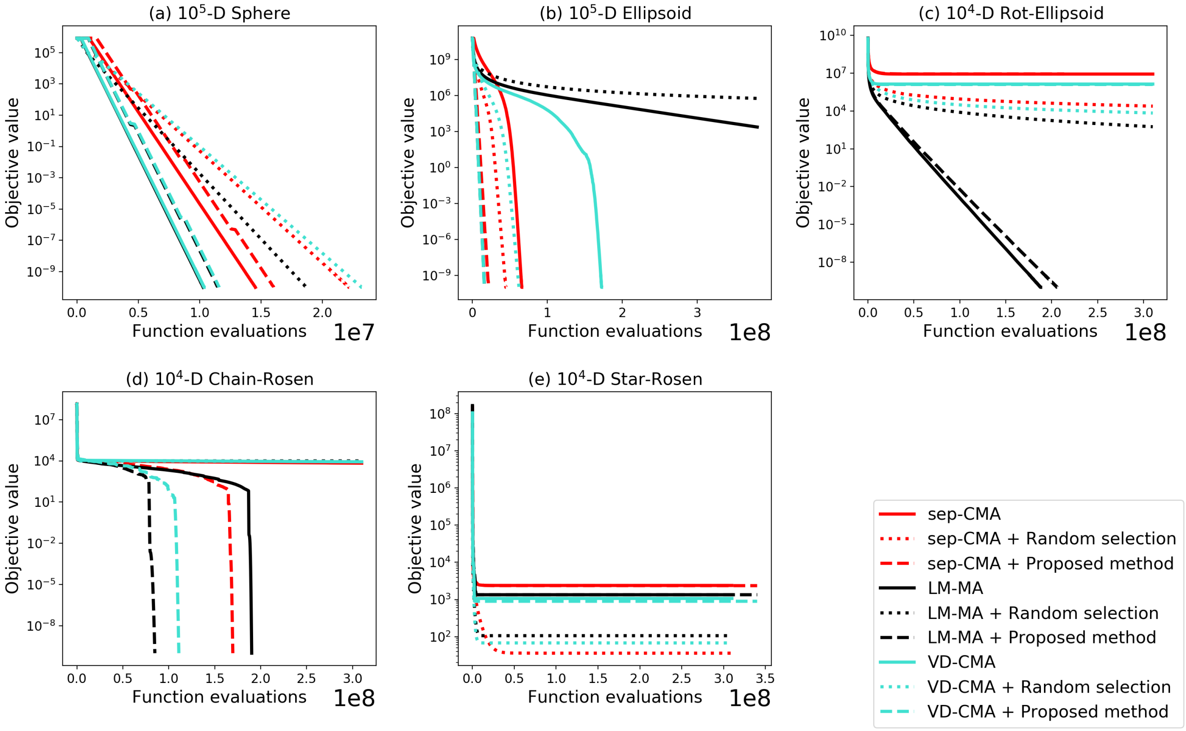

Figure 1: Trajectories of original sep-CMA, LM-MA, and VD-CMA (solid lines), the ones with the random coordinate selection (dotted lines), and the ones with the dynamic partial update (dashed lines). The solid black and cyan lines overlap each other in (a). The dashed black and cyan lines overlap each other in (a). The dashed black and cyan lines overlap each other in (b).

acquire the value each time the mean vector is updated. We consider the rate of change in the partial derivatives of each coordinate to mitigate the cost. If the derivative contains only a constant or low-depend variable, the rate is small, and therefore it is not necessarily computed every generation. In the implementation, for coordinates of which the rate of change is greater than 50%, we compute the derivatives every generation; otherwise, compute when the generation is a multiple of the number of dimensions.

3.2 Dynamic partial update

We divide the set of coordinates into some clusters with constraints that the condition number is less than an acceptable condition number $1 + \alpha$. The condition number of each cluster has approximated the ratio of maximum and minimum second-order partial derivatives.

[Step 1.] Estimate the second-order partial derivatives of each axis of the objective function by the finite difference method and obtain a vector $\boldsymbol{b} = (b_1, ..., b_d)$ whose elements are the absolute values of the second-order partial derivatives in dimension d.

[Step 2.] Find the smallest element $b_{min} = min(\boldsymbol{b})$, and extract the element satisfying $b_i \leq b_{min}(1 + \alpha)$ from the second-order partial differential vector $\boldsymbol{b}$ obtained in Step 1. Replace the vector $\boldsymbol{b}$ by removing the above-extracted elements.

[Step 3.] If the number of elements in $\boldsymbol{b}$ is 0, terminate the clustering; otherwise, repeat Step 2.

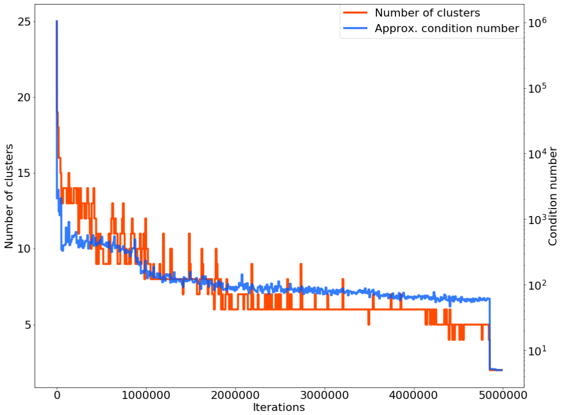

Figure 2: Result of sep-CMA with the dynamic partial update on Chain-Rosenbrock function. Each trajectory shows the number of clusters (red line) and the approximated condition number (blue line).

4 EXPERIMENTS

In this section, we confirm the performance of the dynamic partial update method. We apply our method to sep-CMA, LM-MA, and VD-CMA and compared it with the original algorithms and with the ones with the random coordinate selection. Table 1 shows the definition of the benchmark functions for the experiments. The initial values for all functions are $m^{(0)} = U(-5,5), C^{(0)} = I, \sigma^{(0)} = 1.0$, $\lambda = 4 + 3\lfloor \ln(d) \rfloor$, the target value of the objective function is 10^{-10}, the maximum number of evaluation is $\lambda \times 10^7$ as in study [6]. The number of selecting coordinates for the random coordinate selection is set as 100. The acceptable condition number α for the proposed method is set as $\alpha = 0.5$.

Figure 1 shows the evolutionary trajectories of each algorithm. The red, black, and cyan lines show sep-CMA, LM-MA, and VD-CMA respectively. The solid, dotted, and dashed lines show the original algorithms, the ones with the random coordinate selection, and the ones with the dynamic partial update (proposed method) respectively.

Figure 1 (a) shows our method reduces unnecessary function evaluations on the well-conditioned function compared to the random coordinate selection. In the sphere function case, the second-order partial derivative is 2 for all coordinates. Therefore, our method does not divide the set of coordinates and runs the same flow of original algorithms with the additional cost of occasionally computing the derivative.

Figure 1 (b) and (d) show our approach adapts to the ill-conditionality of the functions and improves the performance greatly. The Chain-Rosenbrock function is often recognized as the well-conditioned function while the Ellipsoid function is famous for its ill-conditionality. However, Figure 2 shows the condition number of the Chain-Rosenbrock function changes drastically according to the optimization steps (the blue line). The actual second-order partial derivative is obtained by the definition as follows,

$$\frac{\partial^2 f}{\partial x_i^2} = \begin{cases} 202 + 400(3x_i^2 - x_{i+1}) & , 1 \leq i \leq d-1 \\ 200 & , d = n \end{cases} . \tag{14}$$

This formula also shows that the value varies greatly depending on the value of the input. The red line in Figure 2 shows the number of clusters and the value changes depending on the condition number of the objective function.

Figure 1 (e) shows that none of the approaches reached the target value on the Star-Rosenbrock function. Our approach performed as well as the original CMA variants, although less than the ones with the random coordinate selection. This function has a dependency of the first variable on all other variables. Therefore, the random coordinate selection, in which the first variable is assigned to clusters on the other coordinates with equal frequency, would have been most appropriate.

5 CONCLUSION

In this paper, we proposed a dynamic partial update method for CMA variants in which the number of coordinates to be selected adapts to the ill-conditionality of the objective function. This method has solved the problem of poor performance of conventional coordinate selection for well-conditional functions. We confirmed that the proposed method improves the performance of objective functions such as the Chain-Rosenbrock function, whose condition number varies depending on the search points.

ACKNOWLEDGEMENT

This work was supported by JST CREST JPMJCR19A4 and JSPS KAKENHI JP21H03445.

REFERENCES

[1] Chang Wook Ahn and R. S. Ramakrishna. 2002. A genetic algorithm for shortest path routing problem and the sizing of populations. *IEEE Transactions on Evolutionary Computation* 6, 6, 566–579.

[2] Youhei Akimoto, Anne Auger, and Nikolaus Hansen. 2014. Comparison-Based Natural Gradient Optimization in High Dimension. In *Proceedings of the 2014 Conference on Genetic and Evolutionary Computation - GECCO '14*. ACM Press, 373–380.

[3] Youhei Akimoto and Nikolaus Hansen. 2016. Projection-Based Restricted Covariance Matrix Adaptation for High Dimension. In *Proceedings of the Genetic and Evolutionary Computation Conference 2016* (Denver, Colorado, USA) *(GECCO '16)*. Association for Computing Machinery, New York, NY, USA, 197–204.

[4] Nikolaus Hansen. 2016. The CMA Evolution Strategy: A Tutorial. *arXiv: 1604.00772*.

[5] Nikolaus Hansen, Steffen Finck, Raymond Ros, and Anne Auger. 2009. Real-Parameter Black-Box Optimization Benchmarking 2009: Noiseless Functions Definitions. *Technical Report RR-6829, INRIA* (2009).

[6] Ilya Loshchilov. 2014. A Computationally Efficient Limited Memory CMA-ES for Large Scale Optimization. In *Proceedings of the 2014 Conference on Genetic and Evolutionary Computation - GECCO '14*. ACM Press, 397–404.

[7] Ilya Loshchilov. 2017. LM-CMA: An Alternative to L-BFGS for Large-Scale Black Box Optimization. *Evolutionary Computation* 25, 1 (2017), 143–171.

[8] Ilya Loshchilov, Tobias Glasmachers, and Hans-Georg Beyer. 2019. Large Scale Black-Box Optimization by Limited-Memory Matrix Adaptation. *IEEE Trans. Evol. Comput.* 23, 2 (April 2019), 353–358.

[9] J. Muñoz-Paniagua, J. García, and A. Crespo. 2014. Genetically aerodynamic optimization of the nose shape of a high-speed train entering a tunnel. *Journal of Wind Engineering and Industrial Aerodynamics* 130, 48 – 61. https://doi.org/10.1016/j.jweia.2014.03.005

[10] Raymond Ros and Nikolaus Hansen. 2008. A Simple Modification in CMA-ES Achieving Linear Time and Space Complexity. In *Parallel Problem Solving from Nature – PPSN X*. Vol. 5199. Springer Berlin Heidelberg, 296–305.

[11] Hiroki Shimizu and Masashi Toyoda. 2021. CMA-ES with Coordinate Selection for High-Dimensional and Ill-Conditioned Functions. *Proceedings of the Genetic and Evolutionary Computation Conference Companion* (2021), 209–210. https://doi.org/10.1145/3449726.3459575

[12] Siu Tong and David Powell. 2003. Genetic Algorithms: A Fundamental Component of an Optimization Toolkit for Improved Engineering Designs, Vol. 2724. 2347–2359. https://doi.org/10.1007/3-540-45110-2_127