Population Size Adaptation for the CMA-ES Based on the Estimation Accuracy of the Natural Gradient

Conference Paper · July 2016

DOI: 10.1145/2908812.2908864

2 authors:

Kouhei Nishida
Shinshu University
7 PUBLICATIONS **17** CITATIONS
SEE PROFILE

Youhei Akimoto
University of Tsukuba
84 PUBLICATIONS **390** CITATIONS
SEE PROFILE

Population Size Adaptation for the CMA-ES Based on the Estimation Accuracy of the Natural Gradient

Kouhei Nishida
Graduate School of Science and Engineering,
Shinshu University
15tm232a@shinshu-u.ac.jp

Youhei Akimoto
Faculty of Engineering,
Shinshu University
y_akimoto@shinshu-u.ac.jp

ABSTRACT

We propose a novel strategy to adapt the population size, i.e. the number of candidate solutions per iteration, for the rank-μ update covariance matrix adaptation evolution strategy (CMA-ES). Our strategy is based on the interpretation of the rank-μ update CMA-ES as the stochastic natural gradient approach on the parameter space of the sampling distribution. We introduce a measurement of the accuracy of the current estimate of the natural gradient. We propose a novel strategy to adapt the population size according to the accuracy measure. The proposed strategy is evaluated on test functions including rugged functions and noisy functions where a larger population size is known to help to find a better solution. The experimental results show the advantage of the adaptation of the population size over a fixed population size. It is also compared with the state-of-the-art uncertainty handling strategy for the CMA-ES, namely UH-CMA-ES, on noisy test functions.

Keywords

Covariance Matrix Adaptation; Natural Gradient; Population Size Adaptation; Noisy Optimization; Ruggedness

1. INTRODUCTION

The Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [6,8,10] is nowadays recognized as a state-of-the-art stochastic search algorithm for optimization of a black-box function $f : \mathbb{R}^d \to \mathbb{R}$. It produces multiple candidate solutions from a multivariate normal distribution and the parameters of the probability distribution are adapted according to the candidate solutions and the ranking of their observed objective values. Due to the covariance matrix adaptation, the CMA-ES efficiently works on ill-conditioned and non-separable functions. Moreover, all the so-called strategy parameters such as the population size (i.e., the number of candidate solutions at each iteration) and the learning rates for the parameter update have their default values (e.g., [6]). Therefore, it is considered as a quasi parameter free method, which is important when optimizing a black-box function since the parameter tuning can be easily the bottleneck of the process of optimization.

When the CMA-ES is applied to an optimization of a rugged function or a noisy function, however, it is empirically known that a larger population size than the default value, namely $\lambda = 4 + \lfloor 3\ln(d) \rfloor$, helps to find a better solution [7]. A reasonable value for the population size varies widely depending on the ruggedness of the objective function and the noise strength. In the black-box scenario where the priori knowledge is limited, it is a prohibitively expensive task to find a reasonable population size in advance. Moreover, if the objective function is noisy and its noise-to-signal ratio increases as it gets close to an optimum, a fixed population size will not be sufficient.

A simple and successful strategy is to restart the CMA-ES with increasing population size [3–5]. It is effective on relatively well-structured multimodal functions. On the other hand, if the function is noisy and its noise-to-signal ratio is unbounded, a large but fixed population size is not sufficient. The state-of-the-art uncertainty handling for the CMA-ES adapts the number of resampling of the noisy function values for each candidate solution and estimates the expected objective value to make the ranking of the candidate solutions accurate [9]. This strategy makes the behavior of the CMA-ES on a noisy function similar to the behavior on the noiseless counterpart except that the number of function evaluations per candidate solution increases. If the noiseless counterpart is rugged, we also need to increase the population size.

To address the ruggedness of the objective function and its uncertainty at the same time, we propose a novel strategy to adapt the population size for the CMA-ES. Two advantages of adapting the population size online are expected. One is that it does not require tuning of the population size in advance. The other is that it may accelerate the search by reducing the population size after converging in a basin of attraction of a local minimum on a rugged function. What is observed in common when applying the CMA-ES to a rugged function and to a noisy function is that the parameter update has a relatively high variance and less tendency compared to when it is applied to unimodal functions. We measure the uncertainty of the parameter update in the CMA-ES and adapt the population size so that the uncertainty measure is kept constant.

The rest of the paper is organized as follows. We first formulate the optimization of a noisy function on a con-

GECCO '16, July 20 - 24, 2016, Denver, CO, USA

ISBN 978-1-4503-4206-3/16/07…$15.00

DOI: http://dx.doi.org/10.1145/2908812.2908864

tinuous domain and review the rank-μ update CMA-ES in Section 2. Since our population size adaptation is based on the interpretation of the rank-μ update CMA-ES as a natural gradient approach on the set of the parameters of the multivariate normal distribution, we derive the rank-μ update CMA-ES as the natural gradient approach. In Section 3 we introduce a novel measurement for the accuracy of the natural gradient estimate and propose an adaptation mechanism for the population size based on the introduced measurement. In Section 4, we conduct experiments to see how the proposed population size adaptation works on unimodal and multimodal functions with and without additive Gaussian noise. It is also compared with the state-of-the-art noise handling for the CMA-ES, namely UH-CMA-ES. We conclude the paper with summary and future work in Section 5.

2. CMA-ES

2.1 Noisy Objective Function

In this paper, we consider both noisy and noiseless unconstrained continuous optimization. In the noisy scenario, we observe the objective function value corrupted by an additive noise ε, namely

$$f(x, \varepsilon) = f(x) + \varepsilon \tag{1}$$

where $f : \mathbb{R}^d \to \mathbb{R}$ is the objective function to be minimized w.l.o.g., $x \in \mathbb{R}^d$ is a candidate solution, and ε is a zero-mean noise that is independently drawn from the identical distribution for each observation.

Our objective is to find the vector $x^\star$ that minimizes the expected objective value

$$J(x) = \int f(x, \varepsilon) p(\varepsilon) d\varepsilon \ , \tag{2}$$

where $p(\varepsilon)$ is the probability density function of ε. Since the noise has zero mean, the expected objective value $J(x)$ is equal to the noiseless function value $f(x)$.

Since the probability distribution of the noise is unkwown, we can not compute $J(x)$. Instead, we can resample the noisy function value $f(x) + \varepsilon$ several times for each candidate solution x, and approximate the expected function value $J(x)$ by averaging them, namely

$$\hat{J}(x) = f(x) + \frac{1}{N_{\mathrm{eval}}} \sum_{i=1}^{N_{\mathrm{eval}}} \varepsilon_i, \tag{3}$$

where N_{eval} is the number of resampling of the function value for each candidate solution , and ε_i for $i = 1, \ldots, N_{\mathrm{eval}}$ are i.i.d. copies of ε. In the noiseless scenario, N_{eval} is set to 1 and $\hat{J}(x) = J(x) = f(x) = f(x, \varepsilon)$. We refer to [11] for an overview of the optimization under uncertainty.

2.2 Rank-μ update CMA-ES

The Covariance Matrix Adaptation Evolution Strategy (CMA-ES) is a stochastic, ranking based search algorithm for black-box continuous optimization. In the CMA-ES the candidate solutions are generated independently from the multivariate normal distribution. Their noisy or noiseless function values are observed N_{eval} times for each candidate solution. They are sorted according to their averaged function values (3). Using the candidate solutions and their ranking information, the parameters of the multivariate normal distribution are updated.

The rank-μ update CMA-ES [8] is a component of the CMA-ES where the Gaussian distribution $\mathcal{N}(\boldsymbol{m}, \mathbf{C})$ is parameterized with the mean vector $\boldsymbol{m} \in \mathbb{R}^d$ and the covariance matrix $\mathbf{C}$ that is a symmetric and positive-definite matrix of dimension d. The rank-μ update CMA-ES is known to be an instantiation of the information-geometric optimization (IGO) [12] that is a unified framework for stochastic search algorithms on an arbitrary domain and is based on the natural gradient [2] on the manifold of a given family of probability distributions. Here we derive the rank-μ update CMA-ES from the IGO framework since our strategy is based on this interpretation of the rank-μ update CMA-ES.

IGO. The IGO transforms the minimization problem on the search space to the maximization problem on the set Θ of the parameters of the probability distributions. For the moment, we assume that we have a full access to the expected objective function $J(x)$. At each iteration t with the parameter $\theta^t \in \Theta$, it transforms J to a function that is invariant to the strictly increasing transformation $g \circ J$ of J, namely,

$$W^J_{\theta^t}(x) = w(P_{\theta^t}[x'; J(x') \leqslant J(x)]) \ , \tag{4}$$

where $w : [0, 1] \to \mathbb{R}$ is a monotonically nonincreasing function. It is easy to see that maximizing $W^J_{\theta^t}(x)$ implies minimizing $J(x)$. We denote the expected value of $W^J_{\theta^t}(x)$ given a distribution P_θ for x by

$$K(\theta) = \int W^J_{\theta^t}(x) q(x|\theta) dx \ . \tag{5}$$

Here $q(x|\theta)$ is the probability density function associated with P_θ on the Lebesgue measure dx.

The IGO takes the steepest ascent step on Θ to maximize (5) at each iteration. As the natural metric on the parameter space Θ of the probability distribution, the Fisher metric is employed. Then, the steepest ascent direction of $K(\theta)$ is known to be given by the so-called natural gradient, which is the product of the inverse Fisher information matrix $\mathcal{I}(\theta)^{-1}$ and the vector $\nabla_\theta K$ of the partial derivatives with respect to each component of θ, which we call as the vanilla gradient of K. The Fisher information matrix for a given parameter θ is

$$\mathcal{I}(\theta) = \int \nabla_\theta \ln q(x|\theta) \nabla_\theta \ln q(x|\theta)^{\mathrm{T}} q(x|\theta) dx \tag{6}$$

and the vanilla gradient of K is expressed as

$$\nabla_\theta K(\theta) = \int W^J_{\theta^t}(x) \nabla_\theta \ln q(x|\theta) q(x|\theta) dx \tag{7}$$

by allowing the exchange of the integration and the differentiation and using $\nabla_\theta q(x|\theta) = q(x|\theta) \nabla_\theta \ln q(x|\theta)$. Letting $\tilde{\nabla}_\theta = \mathcal{I}(\theta)^{-1} \nabla_\theta$, the update is written as $\theta^{t+1} = \theta^t + \eta \tilde{\nabla}_\theta K(\theta^t)$, where η is the step-size, aka the learning rate. In this paper, we allow η to be a diagonal matrix. That is, we may set different learning rates for different components of θ.

We have assumed the full access to J so far, however, we do not know J in practice and we can not analytically compute $W^J_\theta(x)$ and $\tilde{\nabla}_\theta K(\theta)$. Instead, we estimate these quantities by Monte-Carlo. At each iteration we generate λ^t candidate solutions x_i independently from P_{θ^t}. First,

$W^J_{\theta^t}(x)$ is estimated by using

$$P_{\theta^t}[x'; J(x') \leqslant J(x)] = \int \mathbb{I}\{x'; J(x') \leqslant J(x)\} q(x'|\theta^t) dx' \approx \frac{1}{\lambda^t} \sum_{k=1}^{\lambda^t} \mathbb{I}\{x_k; \hat{J}(x_k) \leqslant \hat{J}(x)\} \ . \quad (8)$$

With the estimated $\hat{W}^{\hat{J}}_{\theta^t}(x) = w(\frac{1}{\lambda^t}\sum_{k=1}^{\lambda^t} \mathbb{I}\{x_k; \hat{J}(x_k) \leqslant \hat{J}(x)\})$, the natural gradient is approximated by

$$\tilde{\nabla}_\theta K(\theta^t) \approx G^t := \frac{1}{\lambda^t} \sum_{j=1}^{\lambda^t} \hat{W}^{\hat{J}}_{\theta^t}(x_j) \tilde{\nabla}_\theta \ln q(x_j|\theta^t) \ . \quad (9)$$

The parameter update then follows

$$\theta^{t+1} = \theta^t + \eta G^t \ . \quad (10)$$

Note that the gradient is taken with respect to $\theta \in \Theta$ and the objective function J is not necessarily differentiable with respect to $x \in \mathbb{R}^d$. It only requires the noisy objective values for each candidate solution to estimate the natural gradient. Note also that the estimated weight $\hat{W}^{\hat{J}}_{\theta^t}(x_j)$ takes a value in $\{w(i/\lambda^t) : i = 1, \ldots, \lambda^t\}$. We will write $w^t_i = w(i/\lambda^t)/\lambda^t$ and assume $\sum_{i=1}^{\lambda^t} w^t_i = 1$ w.l.o.g.

IGO with Gaussian. In our case, the probability distributions are the multivariate normal distributions $\mathcal{N}(\boldsymbol{m}, \mathbf{C})$. The parameter vector is $\theta = (\boldsymbol{m}, \mathrm{vech}(\mathbf{C}))$, where $\mathrm{vech}(\mathbf{C})$ is the rearrangement of the lower left triangular part of the symmetric matrix $\mathbf{C}$ such that the $i-j+1+\sum_{k=1}^{j-1}(d-k+1)$th element of $\mathrm{vech}(\mathbf{C})$ is the $(i,\ j)$th element of $\mathbf{C}$. We denote its inverse operation by vech^{-1}. Then, we can analytically compute the natural gradient $\tilde{\nabla}_\theta \ln q(x|\theta)$ of the log-likelihood (log of the probability density function). Decomposing the natural gradient $\tilde{\nabla}_\theta \ln q(x|\theta)$ into two parts $(\tilde{\nabla}_{\boldsymbol{m}} \ln q(x|\theta),\ \tilde{\nabla}_{\mathbf{C}} \ln q(x|\theta))$, we can write

$$\tilde{\nabla}_{\boldsymbol{m}} \ln q(x|\theta) = x - \boldsymbol{m},$$
$$\mathrm{vech}^{-1}\left(\tilde{\nabla}_{\mathbf{C}} \ln q(x|\theta)\right) = (x-\boldsymbol{m})(x-\boldsymbol{m})^{\mathrm{T}} - \mathbf{C} \ .$$

Introducing different learning rates, $\eta_{\boldsymbol{m}}$ and $\eta_{\mathbf{C}}$, for $\boldsymbol{m}$ update and $\mathbf{C}$ update, we finally obtain the parameter update rule for the rank-μ update CMA-ES, $\boldsymbol{m}^{t+1} = \boldsymbol{m}^t + \eta_{\boldsymbol{m}} G^t_{\boldsymbol{m}}$ and $\mathbf{C}^{t+1} = \mathbf{C}^t + \eta_{\mathbf{C}} G^t_{\mathbf{C}}$, where

$$G^t_{\boldsymbol{m}} = \frac{1}{\lambda^t}\sum_{j=1}^{\lambda^t} \hat{W}^{\hat{J}}_{\theta^t}(x_j)(x_j - \boldsymbol{m}^t),$$
$$G^t_{\mathbf{C}} = \frac{1}{\lambda^t}\sum_{j=1}^{\lambda^t} \hat{W}^{\hat{J}}_{\theta^t}(x_j)\left((x_j - \boldsymbol{m}^t)(x_j - \boldsymbol{m}^t)^{\mathrm{T}} - \mathbf{C}^t\right) \ .$$

3. POPULATION SIZE ADAPTATION

3.1 Motivation

There are two factors influencing the precision of the estimation of the natural gradient when the rank-μ update CMA-ES is applied to noisy optimization problems. One is the mis-ranking of candidate solutions due to the noise. In noisy optimization we rank the candidate solutions in (8) according to the expected fitness value $\hat{J}(x)$ averaged over a finite number N_{eval} of resampling of the fitness value for each candidate solution instead of the precise expected fitness value $J(x)$. When the noise-to-signal ratio, i.e., noise strength compared to the function value range of the current distribution, gets greater, the probability of the wrong estimation of the dominance relation of two candidate solutions will be closer to 0.5. Then, the ranking of the candidate solutions estimated in (8) will be different from the ranking of the precise expected function value. It results in low precision of the estimation of the natural gradient in (9).

Increasing the number N_{eval} of resampling of the function value for each candidate solution in (2) will reduce the probability of mis-ranking for fixed or bounded noise-to-signal ratio scenarios. For example, if $f(x) = \sum_{i=1}^d x_i^2$ and the noise is multiplicative such as $f(x, \varepsilon) = (1+\varepsilon)f(x)$ with $\varepsilon \sim \mathcal{N}(0, \sigma^2)$, this works well as long as N_{eval} is sufficiently large depending on the noise strength σ. However, if the noise is additive as in (1), then since the noise-to-signal ratio will become arbitrarily large as the sampling distribution converges towards the optimum, a fixed N_{eval} does not work. In [9], a strategy to adapt N_{eval} is proposed and applied to the CMA-ES, and the resulting algorithm is called UH-CMA-ES. By increasing N_{eval}, the behavior of the CMA-ES on a noisy function becomes similar to the behavior on a corresponding noiseless function except that N_{eval} times more function evaluations are required.

The other factor is in the Monte-Carlo estimate of the natural gradient using a finite number λ of samples. Since the natural gradient is estimated with finite samples, it is stochastic and the precision of the estimate in (9) depends on λ. It is essentially independent of whether the objective function is noisy or not. The ruggedness of the objective function can also be the cause of low accuracy of the natural gradient estimate. We observe that the parameter update in the CMA-ES has a relatively high variance and less tendency, implying a low accuracy of the natural gradient estimate, when applying the CMA-ES to a rugged and/or noisy function.

We empirically know that a larger population helps to find a better solution both on rugged functions and noisy functions. However, similar to what is discussed above for a fixed N_{eval}, we need to adapt λ to optimize the noisy function with increasing noise-to-signal ratio. If we adapt λ so that the precision of the natural gradient estimate is kept sufficiently high, it will be helpful both for noisy functions and rugged functions, especially on the latter of which the adaptation of λ is expected to accelerate the search and relax the tuning cost compared to finding a reasonable λ and fixing it during the optimization.

3.2 Uncertainty Measure

When the objective function is noiseless, it is known from [12] that the natural gradient estimate (9) is consistent, i.e., the natural gradient estimated in (9) converges to $\tilde{\nabla}_\theta K(\theta^t)$ with probability one as $\lambda \to \infty$. When the objective function is noisy, since the mis-ranking probability will not decrease by increasing λ, the estimated natural gradient does not converge to the same limit. However, under some regularity condition it converges towards a different limit, that is,

$$\mathbb{E}_x[\mathbb{E}_{\bar{\varepsilon}}[w(\mathbb{E}_y[c_{\bar{\varepsilon}}(J(x) - J(y) + \bar{\varepsilon})])]\tilde{\nabla}_\theta \ln q(x|\theta^t)] \ , \quad (11)$$

where $\bar{\varepsilon} = N_{\mathrm{eval}}^{-1}\sum_{i=1}^{N_{\mathrm{eval}}} \varepsilon_i$ is the averaged noise; $c_{\bar{\varepsilon}}(t) = \Pr[\bar{\varepsilon} \leqslant t]$ is the cumulative density function of the averaged noise; and the distribution of x and y are P_{θ^t}. Therefore, the behavior of the CMA-ES becomes more and more deterministic as we increase the population size λ, though it will be different from the behavior on the noiseless counterpart. It

indicates that if we quantitatively detect the randomness of the natural gradient estimate, it will be a reasonable quantity which the adaptation of λ is based on.

To quantify the randomness of the estimated natural gradient, we introduce the *evolution path* in the space of the distribution parameter $\theta = (\boldsymbol{m}, \mathbf{C})$. The evolution paths are used in the CMA-ES in two ways. One is used to accelerate the adaptation of the shape of the covariance matrix, where the evolution path is the accumulation of the successive steps in the search space. The other is used to adapt the overall step-size of the search distribution and the step-size adaptation using the evolution path is called the cumulative step-size adaptation [10, 13]. In both usage the evolution paths accumulate the successive steps in the search space $\mathbb{X} \subseteq \mathbb{R}^d$ or in its linear transformation $T(\mathbb{X}) \subseteq \mathbb{R}^d$. In our case, we accumulate the successive steps in the space Θ of the distribution parameters. The step in Θ at each iteration is the estimated natural gradient (9) multiplied by the learning rate. It is updated at each iteration as follows

$$p^{t+1} = (1-\beta)p^t + \sqrt{\beta(2-\beta)}\eta G^t, \tag{12}$$

where β is the cumulation factor for the evolution path.

We measure the length of the evolution path using the Fisher information matrix $\mathcal{I}(\theta)$ of the sampling distribution as

$$\|p\|_\theta^2 := p^T \mathcal{I}(\theta) p \ . \tag{13}$$

It is related to the Kullback-Leibler (KL) divergence between the distributions parameterized by θ and $\theta + p$ as

$$KL(\theta \parallel \theta + p) \approx \frac{1}{2}\|p\|_\theta^2. \tag{14}$$

It means that the length of the evolution path is measured by the magnitude of the influence of the evolution path to the current distribution in terms of the KL-divergence. One reason of our choice is that the KL-divergence is invariant to the choice of the parameterization of the distribution. If we measure the length of the evolution path by its Euclidean norm for example, then it depends on the parameterization of the distribution.

The randomness of the estimated natural gradient is now measured by the ratio between $\|p^{t+1}\|_{\theta^t}^2$ and its expected value $\mathbb{E}[\|p^{t+1}\|_{\theta^t}^2]$ under a random function $f(x) = \varepsilon$, where ε is independently drawn from the identical distribution for each f-call. Note $\mathbb{E}[\|p\|_\theta^2] = \mathbb{E}[p]^{\mathrm{T}} \mathcal{I}(\theta) \mathbb{E}[p] + \mathrm{Tr}(\mathcal{I}(\theta)\,\mathrm{Cov}[p])$. Under the random function, we find that $\mathbb{E}[G^t] = 0$ and $\mathrm{Cov}[G^t] = \mathcal{I}(\theta^t)^{-1}/\mu_{\mathrm{eff}}^t$, where

$$\mu_{\mathrm{eff}}^t = \frac{(\lambda^t)^2}{\sum_{j=1}^{\lambda^t} (\hat{W}_\theta^j(x_j))^2} \ . \tag{15}$$

Therefore we find $\mathbb{E}[\|G^t\|_{\theta^t}^2] = d(d+3)/(2\mu_{\mathrm{eff}}^t)$. Moreover, if η is a block diagonal matrix $\eta = \mathrm{diag}(\eta_1 \mathbf{I}_1, \ldots, \eta_k \mathbf{I}_k)$ with the same block diagonal structure as the Fisher information matrix $\mathcal{I}(\theta) = \mathrm{diag}(\mathcal{I}_1(\theta), \ldots, \mathcal{I}_k(\theta))$, we find that $\mathbb{E}[\|\eta G^t\|_{\theta^t}^2] = \mathrm{Tr}(\eta^2)/\mu_{\mathrm{eff}}^t$. Here, $\mathbf{I}_i$ is the identity matrix with the same dimension as $\mathcal{I}_i(\theta)$, and $\eta_i > 0$. See Appendix A for the derivation.

For the further deviation of $\mathbb{E}[\|p^{t+1}\|_{\theta^t}^2]$ we assume that $(\mathcal{I}(\theta^i))_{i=0,\ldots,t}$ are almost constant under the random function. It reflects the situation where the learning rate is small enough or the population size is large enough for $(\theta^i)_{i=0,\ldots,t}$ to stay close due to the small learning rate or the short natural gradient estimate with zero expectation. Under this assumption, since $(G^i)_{i=0,\ldots,t}$ are uncorrelated to each other, we obtain

$$\begin{aligned}
\mathbb{E}[\|p^{t+1}\|_{\theta^t}^2] &= \beta(2-\beta)\textstyle\sum_{i=0}^t (1-\beta)^{2(t-i)} \mathbb{E}[\|G^i\|_{\theta^t}^2] \\
&\approx \beta(2-\beta)\textstyle\sum_{i=0}^t (1-\beta)^{2(t-i)} \mathbb{E}[\|G^i\|_{\theta^i}^2] \\
&= \mathrm{Tr}(\eta^2)\beta(2-\beta)\textstyle\sum_{i=0}^t \frac{(1-\beta)^{2(t-i)}}{\mu_{\mathrm{eff}}^i} \ .
\end{aligned}$$

We denote the right-most side of the above equality by γ^{t+1} and use it as an approximation of $\mathbb{E}[\|p^{t+1}\|_{\theta^t}^2]$ under a random function. It can be updated iteratively as

$$\gamma^{t+1} = (1-\beta)^2\gamma^t + \beta(2-\beta)\frac{\mathrm{Tr}(\eta^2)}{\mu_{\mathrm{eff}}^t} \ . \tag{16}$$

We measure the uncertainty of the natural gradient estimate by the ratio $\|p^{t+1}\|_{\theta^t}^2/\gamma^{t+1}$.

3.3 Population Size Adaptation

We adapt the population size using the uncertainty measure introduced in the last section. If the ratio $\|p^{t+1}\|_{\theta^t}^2/\gamma^{t+1}$ is lower than a given constant $\alpha > 1$, we regard the natural gradient estimate as inaccurate and increase the population size for the next iteration as

$$\lambda^{t+1} = \left\lfloor \lambda^t \exp\left(\beta\left(\alpha - \frac{\|p^{t+1}\|_{\theta^t}^2}{\gamma^{t+1}}\right)\right)\right\rfloor \vee \lambda^t + 1 \ . \tag{17}$$

If $\|p^{t+1}\|_{\theta^t}^2/\gamma^{t+1} > \alpha$, we regard the natural gradient estimate as sufficiently accurate and decrease the population size according to

$$\lambda^{t+1} = \left\lfloor \lambda^t \exp\left(\beta\left(\alpha - \frac{\|p^{t+1}\|_{\theta^t}^2}{\gamma^{t+1}}\right)\right)\right\rfloor \vee \lambda_{\min} \ , \tag{18}$$

where $\lambda_{\min}$ is the minimum population size and is set to 4 in the experiments, $\vee$ is a binary operator that takes the maximum of two values, $\lfloor \cdot \rfloor$ is the floor function.

To understand the algorithm mathematically, we assume that $(G^i)_{i=0,\ldots,t}$ independently follow the same distribution that has the mean vector $\mathbb{E}[G]$ and the covariance matrix $\mathrm{Cov}[G]$, and initialize $p^0 = 0$. Then, it follows that the expected value $\mathbb{E}[p^{t+1}]$ and the covariance matrix $\mathrm{Cov}[p^{t+1}]$ of p^{t+1} updated by (12) are

$$\begin{aligned}
\mathbb{E}[p^{t+1}] &= [1-(1-\beta)^{t+1}]\sqrt{(2-\beta)/\beta}\,\mathbb{E}[\eta G] \\
\mathrm{Cov}[p^{t+1}] &= [1-(1-\beta)^{2(t+1)}]\,\mathrm{Cov}[\eta G] \ ,
\end{aligned}$$

respectively. Then, we have

$$\mathbb{E}[\|p^{t+1}\|_\theta^2] \approx \frac{2-\beta}{\beta}\|\mathbb{E}[\eta G]\|_{\theta^t}^2 + \mathrm{Tr}(\mathcal{I}(\theta^t)\,\mathrm{Cov}[\eta G]) \ . \tag{19}$$

If the noise-to-signal ratio is high and the population size is not sufficiently large, the ranking according to the averaged function value (8) is nearly random and the situation will be close to the random function. Then, $\|p^{t+1}\|_\theta^2/\gamma^{t+1}$ will be close to one. On the other hand, since $\gamma^{t+1} \to 0$ and $\mathrm{Cov}[G] \to 0$ as $\lambda \to \infty$ and $\mathbb{E}[G] \not\to 0$ in general, the first term on the RHS of (19) will be more emphasized and the ratio $\|p^{t+1}\|_\theta^2/\gamma^{t+1}$ can be greater than α if λ is sufficiently large. Therefore, by (17) and (18), we expect that λ is adapted so that $\|p^{t+1}\|_\theta^2/\gamma^{t+1}$ is stabilized around α.

The required λ to reach α will be smaller if β is smaller due to the factor $\frac{2-\beta}{\beta}$ in (19). If the learning rate η for the parameter update is small, the effect of the noise on

Table 1: Test function definitions.

Definition
$f_{\text{Sphere}}(x) = \sum_{i=1}^{d} x_i^2$
$f_{\text{Ellipsoid}}(x) = \sum_{i=1}^{d} 10^{\frac{6(i-1)}{d-1}} x_i^2$
$f_{\text{Rastrigin}}(x) = \sum_{i=1}^{d} \left(x_i^2 + 10(1 - \cos 2\pi x_i)\right)$
$f_{\text{Schaffer}}(x) = \sum_{i=1}^{d-1} [x_i^2 + x_{i+1}^2]^{1/4} [\sin^2(50(x_i^2 + x_{i+1}^2)^{0.1}) + 1]$

the natural gradient estimates is averaged over iterations. Therefore, it is natural to set β smaller as η is set smaller.

3.4 Explicit Formula for Gaussian

The proposed strategy can be applied to any instance of the IGO since we assume up to here nothing on the sampling distribution and its parameterization, except that the learning rate η has the same block diagonal structure as the Fisher information matrix, which is used to derive γ.

In the rank-μ update CMA-ES, the sampling distribution is the Gaussian distribution $\mathcal{N}(\boldsymbol{m}, \mathbf{C})$ with the mean vector $\boldsymbol{m}$ and the covariance matrix $\mathbf{C}$. The Fisher information matrix of $\theta = (\boldsymbol{m}, \mathbf{C})$ is known to be a block diagonal matrix $\text{diag}(\mathcal{I}_{\boldsymbol{m}}, \mathcal{I}_{\mathbf{C}})$, where $\mathcal{I}_{\boldsymbol{m}}$ is of $d \times d$, $\mathcal{I}_{\mathbf{C}}$ is of $d(d+1)/2 \times d(d+1)/2$. Therefore, having different learning rates $\eta_{\boldsymbol{m}}$ and $\eta_{\mathbf{C}}$ for the mean vector update and the covariance matrix update, respectively, does not violate the assumption to derive γ. Moreover, we have an explicit formula for $\|p\|_\theta^2$.

First, we can update the evolution path for the mean vector and covariance matrix separately as

$$\begin{aligned} p_{\boldsymbol{m}}^{t+1} &= (1-\beta) p_{\boldsymbol{m}}^{t+1} + \sqrt{\beta(2-\beta)}\eta_{\boldsymbol{m}} G_{\boldsymbol{m}}^t \\ p_{\mathbf{C}}^{t+1} &= (1-\beta) p_{\mathbf{C}}^{t+1} + \sqrt{\beta(2-\beta)}\eta_{\mathbf{C}} G_{\mathbf{C}}^t \quad . \end{aligned} \tag{20}$$

Note that $p_{\mathbf{C}}$ is a symmetric matrix, not a vector. Then, $p^{t+1} = (p_{\boldsymbol{m}}^{t+1}, \text{vech}(p_{\mathbf{C}}^{t+1}))$ and

$$\|p^{t+1}\|_{\theta^t}^2 = (p_{\boldsymbol{m}}^{t+1})^{\mathrm{T}} (\mathbf{C}^t)^{-1} p_{\boldsymbol{m}}^{t+1} + \frac{\text{Tr}((p_{\mathbf{C}}^{t+1}(\mathbf{C}^t)^{-1})^2)}{2} \quad . \tag{21}$$

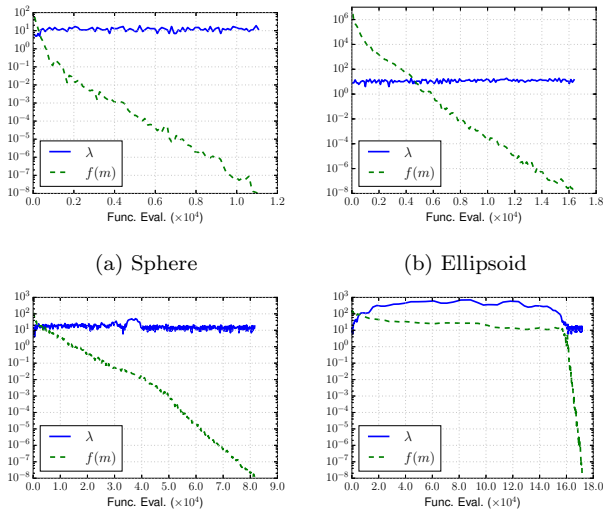
Note that the second term on the right-hand side of the above equality is the sum of the square of each element of $p_{\mathbf{C}}^{t+1}(\mathbf{C}^t)^{-1}$. In practice, we typically compute the eigendecomposition of the covariance matrix to sample candidate solutions, which is sufficient to compute the inverse of $\mathbf{C}$, so we do not need to perform an additional matrix inversion. For the derivation of (21), see Appendix B.

4. EXPERIMENT

We conduct experiments to see how the population size is adapted in the proposed algorithm on noiseless and noisy functions and to compare the proposed algorithm with the rank-μ update CMA-ES with a fixed population size and the UH-CMA-ES.

The test function definitions are summarized in Table 1. The initial mean vector is $\boldsymbol{m}^0 = (3, \ldots, 3)$, the initial covariance matrix is $\mathbf{C}^0 = 2^2 \cdot \mathbf{I}$ for all but the Schaffer function, where $\boldsymbol{m}^0 = (55, \ldots, 55)$ and $\mathbf{C}^0 = 45^2 \cdot \mathbf{I}$. For all the functions, the global optimal solution is located at $x^\star = \mathbf{0}$ and the function value is zero. In noisy scenario, we consider the additive Gaussian noise $\varepsilon \sim \mathcal{N}(0, \sigma_{\text{noise}}^2)$.

We use the following values for the strategy parameters. The minimal population size $\lambda_{\text{min}} = 4$. The weight value $w_i^t = [\ln((\lambda^t+1)/2) - \ln(i)] / \sum_{i=1}^{\lambda^t} [\ln((\lambda^t+1)/2) - \ln(i)]$ for $i \in [\![1, \lfloor \lambda^t/2 \rfloor]\!]$ and $w_i^t = 0$ for $i \geqslant \lfloor \lambda^t/2 \rfloor$. The learning rate for the mean update $\eta_{\boldsymbol{m}} = 0.1$ ($\eta_{\boldsymbol{m}} = 1/d$ would be the default value though) and the learning rate for the covariance matrix $\eta_{\mathbf{C}} = \sqrt{2/(d+1)}\eta_{\boldsymbol{m}}$, the latter of which is found from our preliminary study where $(\eta_{\boldsymbol{m}}/\eta_{\mathbf{C}})^2 = \frac{d(d+1)/2}{d}$ (ratio of the numbers of the parameters) tends to perform better for the rank-μ update CMA-ES than the default parameters used in e.g., [7]. The cumulation factor for the evolution path $\beta = \eta_{\boldsymbol{m}}$ and the threshold for the population size adaptation $\alpha = \sqrt{2}$. The result of the preliminary parameter survey is presented in Appendix C.



(a) Sphere (b) Ellipsoid

(c) Schaffer (d) Rastrigin

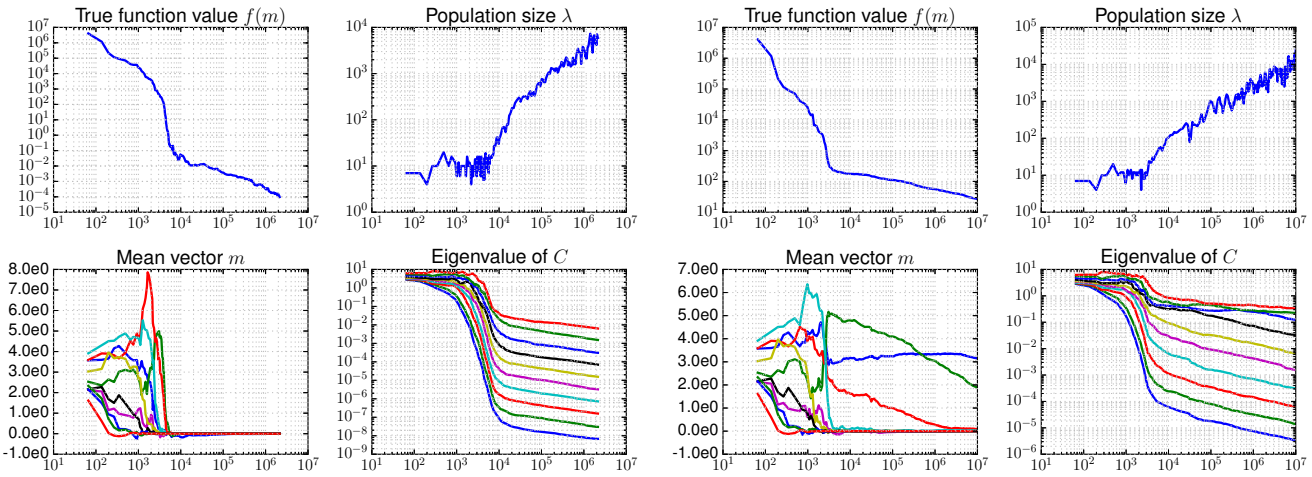
Figure 1: A typical behavior of the proposed algorithm. The noiseless function value $f(\boldsymbol{m}^t)$ and λ^t are displayed.

To measure and compare the performance of different algorithms, we employ the empirical cumulative density function used in the COmparing Continuous Optimizers (COCO) framework[1]. We define N_{target} target values. We record the number of function evaluations spent until the noiseless function value $f(\boldsymbol{m})$ hits a smaller value than each target value for the first time. The data is collected by running N_{trial} independent trials. In total, we have $N_{\text{target}} \cdot N_{\text{trial}}$ targets for each setting. Figures 3 and 4 show the proportion of the target values reached within each number of function evaluations. The target values are set to $10^{1-5(i-1)/(N_{\text{target}}-1)}$ for $i = 1, \ldots, N_{\text{target}}$, and the number of trials is $N_{\text{trial}} = 10$. In total, we have 500 targets for each setting. Note that the figures are essentially related with the convergence graph in log-scale in the range of $[10^{-4}, 10]$ if they are flipped vertically, and they agree if $N_{\text{trial}} = 1$.

4.1 Noiseless Functions

Figure 1 shows a typical behavior of the proposed algorithm on each noiseless function. On the unimodal functions (Sphere and Ellipsoid), the population size is kept around 10. It is a desired behavior since we have found from our preliminary experiments that the optimal fixed population size is around 10 for these functions. In contrast, on the

[1] http://coco.gforge.inria.fr/



(a) $\sigma_n = 10^0$ (b) $\sigma_n = 10^3$

Figure 2: A typical behavior on the noisy Ellipsoid function ($d = 10$). The noiseless function value $f(\boldsymbol{m}^t)$, the population size λ^t, the mean vector $\boldsymbol{m}^t$ and the eigenvalues of $\mathbf{C}^t$ versus the number of function evaluations are displayed.

multimodal functions (Schaffer and Rastrigin) the population size is increased at the beginning of the search and is decreased to around 10 when the function value starts converging towards zero. It is because that the function can be seen as a unimodal function after the sampling distribution is sufficiently concentrated at a basin of attraction of a minimal point. In the experiments, 20 and 16 runs among 20 runs have succeeded to find the target value of 10^{-8} on the Schaffer and Rastrigin functions, respectively.

4.2 On Noisy Functions

Figure 2 shows a typical behavior of the proposed algorithm on the 10 dimensional noisy Ellipsoid function with $\sigma_{\text{noise}} = 1$ and $\sigma_{\text{noise}} = 10^3$. When $\sigma_{\text{noise}} = 1$, the noise-to-signal ratio is very low at the beginning of the run and the population size stays around 10. After $\mathbf{C}$ becomes almost proportional to the inverse Hessian, the noise effect comes into play and the population size starts increasing. When $\sigma_{\text{noise}} = 10^3$, since the noise-to-signal ratio becomes relatively high after the condition number of the product of the inverse Hessian and $\mathbf{C}$ becomes less than roughly 10^3, the population size starts increasing earlier.

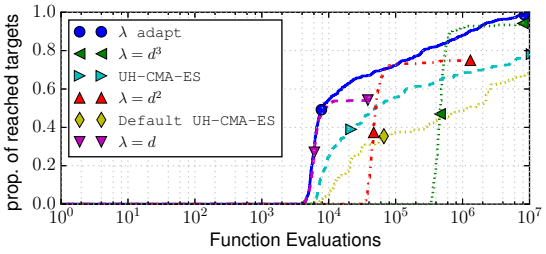
In Figures 3, the proposed algorithm is compared with the rank-μ update CMA-ES with fixed $\lambda = d, d^2, d^3$, the UH-CMA-ES with the default setting, the rank-μ update UH-CMA-ES with the same learning rate η as in the proposed algorithm. The objective function is the noisy Ellipsoid function with $\sigma_{\text{noise}} = 1$. If the population size is fixed, a larger population size helps to find a better solution. However, after reached a noise-to-signal ratio depending on the population size, the quality of the solution stops improving. Moreover, a large and fixed population size wastes the function evaluations when the noise-to-signal ratio is relatively small. On the contrary, the proposed approach is as fast as the one with a small population size to reach relatively weak target values, and continues to improve the quality of the solution by increasing the population size. The proposed approach is advantageous in terms of the quality of the solution and the efficiency of the search. Compared to the UH-CMA-ES, it is faster to reach relatively weak target values, but we do not see a significant difference in the convergence rate from this figure.

Figure 4 shows the empirical cumulative density function for the proposed algorithm and the rank-μ update UH-CMA-ES with $\lambda = d, d^{1.5}, d^2, d^3$. Since the Rastrigin function is well-structured but highly multimodal, the CMA-ES fails to find the global optimum with its default population size. Since the UH-CMA-ES makes its behavior closer to the behavior on the noiseless counterpart by increasing N_{eval}, it fails as well and a larger population size is needed to tackle the ruggedness. On the other hand, the proposed algorithm can tackle the noise and the ruggedness at the same time by increasing the population size.

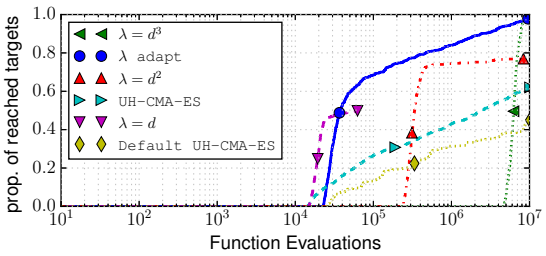
5. CONCLUSION

In this paper we have proposed a novel strategy to adapt the population size in the rank-μ update CMA-ES. We employ the evolution path on the manifold of the parameters of the sampling distribution to measure the uncertainty of the parameter update. We control the population size so that we keep the uncertainty measure of the parameter update to be constant. The experiments has been conducted and have shown that it keeps the population size constant on unimodal functions, increases the population size at the beginning on multimodal functions and decreases it once the distribution is concentrated at the basin of the attraction of the optimum. On noisy functions, we have observed that the population size is adapted (increased) according to the noise-to-signal ratio. Since the proposed algorithm can tackle the ruggedness and the uncertainty at the same time, it is advantageous on a rugged and noisy function where the population size needs to be tuned for the UH-CMA-ES.

In this paper we have shown an elementary results to understand and see how the proposed algorithm adapts the population size on noisy and noiseless functions. In the future work, we will conduct a thorough experiments using a standard benchmarking such as BBOB. We also incorporate

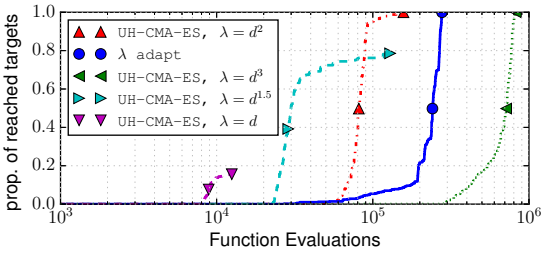


(a) $d = 10$

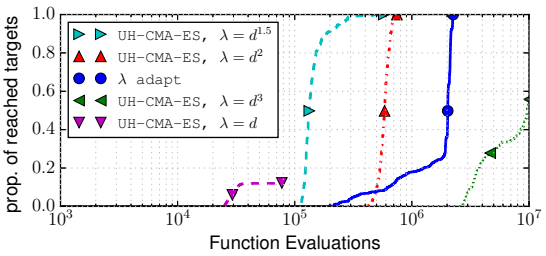


(b) $d = 20$

Figure 3: Performance measure on the noisy Ellipsoid function with the noise strength $\sigma_{\text{noise}} = 1$.



(a) $d = 10$



(b) $d = 20$

Figure 4: Performance measure on the noisy Rastrigin function with the noise strength $\sigma_{\text{noise}} = 1$.

the rank-one update covariance matrix adaptation and the step-size adaptation. A theoretical and experimental investigation of the influence of the parameters α and β needs to be conducted.

Acknowledgements. This work is partially supported by JSPS KAKENHI Grant Number 15K16063 and the Kayamori Foundation of Informational Science Advancement.

6. REFERENCES

[1] Y. Akimoto, Y. Nagata, I. Ono, and S. Kobayashi. Theoretical Foundation for CMA-ES from Information Geometry Perspective. *Algorithmica*, 64:698–716, 2012.

[2] S. Amari. Natural Gradient Works Efficiently in Learning. *Neural Computation*, 10(2):251–276, 1998.

[3] A. Auger and N. Hansen. A Restart CMA Evolution Strategy With Increasing Population Size. In *2005 IEEE Congress on Evolutionary Computation*, pages 1769–1776. Ieee, 2005.

[4] N. Hansen. Benchmarking a BI-population CMA-ES on the BBOB-2009 function testbed. In *Workshop Proceedings of the GECCO Genetic and Evolutionary Computation Conference*, pages 2389–2395, New York, New York, USA, 2009. ACM Press.

[5] N. Hansen. Benchmarking a BI-population CMA-ES on the BBOB-2009 noisy testbed. In *GECCO '09: Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers*. ACM Request Permissions, July 2009.

[6] N. Hansen and A. Auger. Principled Design of Continuous Stochastic Search: From Theory to Practice. In Y. Borenstein and A. Moraglio, editors, *Theory and Principled Methods for the Design of Metaheuristics*. Springer, 2014.

[7] N. Hansen and S. Kern. Evaluating the CMA Evolution Strategy on Multimodal Test Functions. In *Parallel Problem Solving from Nature - PPSN VIII*, pages 282–291. Springer, 2004.

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

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

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

[11] Y. Jin and J. Branke. Evolutionary optimization in uncertain environments-a survey. *Evolutionary Computation, IEEE Transactions on*, 9(3):303–317, 2005.

[12] Y. Ollivier, L. Arnold, A. Auger, and N. Hansen. Information-geometric optimization algorithms: A unifying picture via invariance principles. *arXiv:1106.3708*, 2011.

[13] A. Ostermeier, A. Gawelczyk, and N. Hansen. Step-size adaptation based on non-local use of selection information. In *Parallel Problem Solving from Nature - PPSN III*, pages 189–198, 1994.

APPENDIX

A. DERIVATION FOR SECTION 3.2

Under a random function, since $f(x)$ is at random, so is the ranking of each candidate solution computed in (8). Then, $(\hat{W}_\theta^{\hat{j}}(x_j))_{j=1,\dots,\lambda}$ is uncorrelated on $(x_j)_{j=1,\dots,\lambda}$ and is a random permutation of $(w(j/\lambda))_{j=1,\dots,\lambda}$. Then, using $\mathbb{E}[\nabla \ln q(x_i|\theta^t)] = 0$, we have

$$\begin{aligned}\mathbb{E}[G^t] &= \tfrac{1}{\lambda}\textstyle\sum_{i=1}^{\lambda} \mathbb{E}[\hat{W}_\theta^{\hat{j}}(x_i)\tilde{\nabla} \ln q(x_i|\theta^t)] \\ &= \tfrac{1}{\lambda}\textstyle\sum_{i=1}^{\lambda} \mathbb{E}[\hat{W}_\theta^{\hat{j}}(x_i)]\mathbb{E}[\tilde{\nabla} \ln q(x_i|\theta^t)] = 0 \ .\end{aligned}$$

Moreover, using $\mathbb{E}[\nabla \ln q(x_i|\theta^t)\nabla \ln q(x_i|\theta^t)^{\mathrm{T}}] = \mathcal{I}(\theta^t)$ and $\mathbb{E}[\nabla \ln q(x_i|\theta^t)\nabla \ln q(x_j|\theta^t)^{\mathrm{T}}] = 0$ for $i \neq j$, we have

$$\begin{aligned}&\mathbb{E}[G^t(G^t)^{\mathrm{T}}] \\ &= \tfrac{1}{\lambda^2}\textstyle\sum_{i\neq j} \mathbb{E}[\hat{W}_\theta^{\hat{j}}(x_i)\hat{W}_\theta^{\hat{j}}(x_j)\tilde{\nabla} \ln q(x_i|\theta^t)\tilde{\nabla} \ln q(x_j|\theta^t)^{\mathrm{T}}] \\ &= \tfrac{1}{\lambda^2}\textstyle\sum_{i\neq j} \mathbb{E}[\hat{W}_\theta^{\hat{j}}(x_i)\hat{W}_\theta^{\hat{j}}(x_j)]\mathbb{E}[\tilde{\nabla} \ln q(x_i|\theta^t)\tilde{\nabla} \ln q(x_j|\theta^t)^{\mathrm{T}}] \\ &= \tfrac{1}{\lambda^2}\textstyle\sum_{i=1}^{\lambda} \mathbb{E}[\hat{W}_\theta^{\hat{j}}(x_i)^2]\mathbb{E}[\tilde{\nabla} \ln q(x_i|\theta^t)\tilde{\nabla} \ln q(x_i|\theta^t)^{\mathrm{T}}] \\ &= \tfrac{1}{\lambda^2}\textstyle\sum_{i} \mathbb{E}[\hat{W}_\theta^{\hat{j}}(x_i)^2]\mathcal{I}(\theta^t)^{-1} \\ &= \tfrac{1}{\lambda^2}\textstyle\sum_{i} w(i/\lambda)^2\mathcal{I}(\theta^t)^{-1} = \tfrac{1}{\mu_{\mathrm{eff}}^t}\mathcal{I}(\theta^t)^{-1} \ .\end{aligned}$$

Let $\mathcal{I}(\theta^t) = \mathrm{diag}(\mathcal{I}_1, \dots, \mathcal{I}_k)$ and $\eta = \mathrm{diag}(\eta_1\mathbf{I}, \dots, \eta_k\mathbf{I})$, where each identity matrix $\mathbf{I}_i$ has the same dimension as $\mathcal{I}_i$ for $i = 1, \dots, k$. In the same way, we decompose G^t as $(G_1^t, \dots, G_k^t)$. Since $\eta\mathcal{I}(\theta)\eta = \mathrm{diag}(\eta_1^2\mathcal{I}_1, \dots, \eta_k^2\mathcal{I}_k)$, we have

$$\begin{aligned}\mathbb{E}[\|\eta G^t\|_{\theta^t}^2] &= \mathbb{E}[(G^t)^{\mathrm{T}}(\eta\mathcal{I}(\theta^t)\eta)G^t] \\ &= \mathbb{E}[(G^t)^{\mathrm{T}}\,\mathrm{diag}(\eta_1^2\mathcal{I}_1, \dots, \eta_k^2\mathcal{I}_k)G^t] \\ &= \textstyle\sum_{i=1}^{k} \eta_i^2\mathbb{E}[(G_i^t)^{\mathrm{T}}\mathcal{I}_i G_i^t] \\ &= \textstyle\sum_{i=1}^{k} \eta_i^2\,\mathrm{Tr}\left(\mathcal{I}_i\mathbb{E}[G_i^t(G_i^t)^{\mathrm{T}}]\right) \\ &= \textstyle\sum_{i=1}^{k} \eta_i^2\,\mathrm{Tr}(\mathbf{I}_i)/\mu_{\mathrm{eff}}^t = \mathrm{Tr}(\eta^2)/\mu_{\mathrm{eff}}^t \ .\end{aligned}$$

B. EXPLICIT FORMULA FOR GAUSSIAN

For the Gaussian distribution $\mathcal{N}(\boldsymbol{m}, \mathbf{C})$ with the mean vector $\boldsymbol{m}$ and the covariance matrix $\mathbf{C}$ with the parameter vector $\theta = (\boldsymbol{m}, \mathrm{vech}(\mathbf{C}))$, it is known from [1] that the Fisher information matrix of this θ is a diagonal matrix $\mathrm{diag}(\mathcal{I}_{\boldsymbol{m}}, \mathcal{I}_{\mathbf{C}})$, where $\mathcal{I}_{\boldsymbol{m}} = \mathbf{C}^{-1}$ and $\mathcal{I}_{\mathbf{C}}$ is an $n(n+1)/2$ dimensional symmetric matrix given by

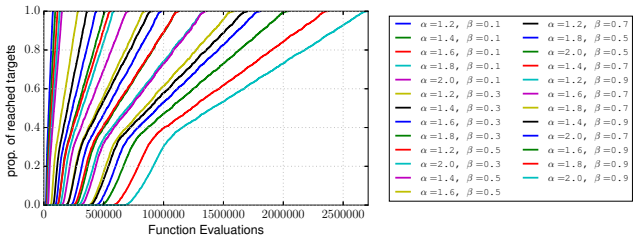
$$\mathcal{I}_{\mathbf{C}} = -\tfrac{1}{2}\left(\tfrac{\partial(\mathrm{vech}(2\mathbf{C}^{-1} - \mathrm{diag}(\mathbf{C}^{-1}))^{\mathrm{T}}}{\partial\,\mathrm{vech}(\mathbf{C})}\right) \ .$$

Let $q = \mathrm{vech}(\mathbf{M}_q)$ be the half vectorization of a symmetric matrix $\mathbf{M}_q$ of dimension n. Using the fact that $\mathrm{vech}(2\mathbf{C}^{-1} - \mathrm{diag}(\mathbf{C}^{-1}))^{\mathrm{T}}\,\mathrm{vech}(\mathbf{M}_q) = \mathrm{Tr}(\mathbf{C}^{-1}\mathbf{M}_q)$, we have

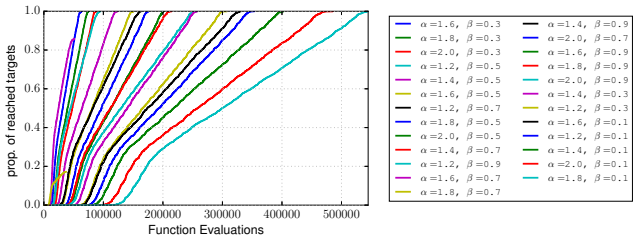
$$\mathcal{I}_{\mathbf{C}}q = -\tfrac{1}{2}\left(\tfrac{\partial\,\mathrm{Tr}(\mathbf{C}^{-1}\mathbf{M}_q)}{\partial\,\mathrm{vech}(\mathbf{C})}\right) \ .$$

Then, we have

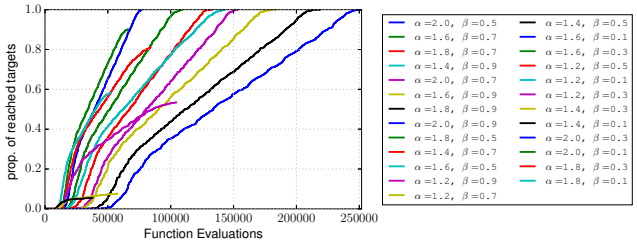
$$\begin{aligned}q^{\mathrm{T}}\mathcal{I}_{\mathbf{C}}q &= -\tfrac{1}{2}\textstyle\sum_{k=1}^{n(n+1)/2}[q]_k\tfrac{\partial\,\mathrm{Tr}(\mathbf{C}^{-1}\mathbf{M}_q)}{\partial[\mathrm{vech}(\mathbf{C})]_k} \\ &= \tfrac{1}{2}\textstyle\sum_{k=1}^{n(n+1)/2}[q]_k\,\mathrm{Tr}\left(\tfrac{\partial\mathbf{C}}{\partial[\mathrm{vech}(\mathbf{C})]_k}\mathbf{C}^{-1}\mathbf{M}_q\mathbf{C}^{-1}\right) \\ &= \tfrac{1}{2}\textstyle\sum_{j=1}^{n}\sum_{i=j}^{n}[\mathbf{M}_q]_{i,j}(2-\delta_{i,j})[\mathbf{C}^{-1}\mathbf{M}_q\mathbf{C}^{-1}]_{i,j} \\ &= \tfrac{1}{2}\textstyle\sum_{j=1}^{n}\sum_{i=1}^{n}[\mathbf{M}_q]_{i,j}[\mathbf{C}^{-1}\mathbf{M}_q\mathbf{C}^{-1}]_{i,j} \\ &= \tfrac{1}{2}\,\mathrm{Tr}(\mathbf{M}_q\mathbf{C}^{-1}\mathbf{M}_q\mathbf{C}^{-1}) \ .\end{aligned}$$



(a) $\eta_m = 0.1$



(b) $\eta_m = 0.5$



(c) $\eta_m = 1$

Figure 5: Parameter survey on 20 dimensional noiseless Ellipsoid function. The legends are sorted in the ascending order w.r.t. the number of function evaluations to reach the proportion of reached targets of 1. For the lines that do not reach the top of the figure, the legends are placed below and sorted in the descending order w.r.t. the proportion of reached targets.

From the first line to the second line we used the formula $\frac{\partial\mathbf{C}^{-1}}{\partial\theta} = -\mathbf{C}^{-1}\frac{\partial\mathbf{C}}{\partial\theta}\mathbf{C}^{-1}$. Therefore, we have for any $p = (p_{\boldsymbol{m}}, p_{\mathbf{C}}) \in \mathbb{R}^n \times \mathbb{R}^{n(n+1)/2}$,

$$p^{\mathrm{T}}\mathcal{I}p = p_{\boldsymbol{m}}^{\mathrm{T}}\mathbf{C}^{-1}p_{\boldsymbol{m}} + \frac{1}{2}\,\mathrm{Tr}((\mathrm{vech}^{-1}(p_{\mathbf{C}})\mathbf{C}^{-1})^2) \ .$$

C. PARAMETER SURVEY

Figure 5 shows the empirical cumulative density function on the 20 dimensional noiseless Ellipsoid function. The target values are set to $10^{1-9(i-1)/(N_{\mathrm{target}}-1)}$ for $i \in [\![1, N_{\mathrm{target}}]\!]$, and the number of trials is $N_{\mathrm{trial}} = 20$. The values for $\alpha = 1.2, 1.4, \dots, 2.0$, the values for $\beta = 0.1, 0.3, \dots, 0.9$, and the other settings are the same as in Section 4. In this figure we can see the dependency of the reasonable parameter setting for α and β on the learning rate $\eta_{\boldsymbol{m}}$ of the mean vector on the noiseless 20 dimensional Ellipsoid function. The smaller α and β are, the faster the noiseless function value converges when $\eta_{\boldsymbol{m}} = 0.1$. On the other hands, a small β tends to interfere the convergence towards the optimum as $\eta_{\boldsymbol{m}}$ becomes greater. Overall, the combination of $\alpha = \sqrt{2}$ and $\beta = \min(\eta_{\boldsymbol{m}}, 0.9)$ seems to be a reasonable setting.