Sample Reuse in the Covariance Matrix Adaptation Evolution Strategy Based on Importance Sampling

Shinichi Shirakawa
University of Tsukuba
1-1-1 Tennodai, Tsukuba, Ibaraki, Japan
shirakawa@iit.tsukuba.ac.jp

Youhei Akimoto
Shinshu University
4-17-1 Wakasato, Nagano City, Nagano, Japan
y_akimoto@shinshu-u.ac.jp

Kazuki Ouchi
Aoyama Gakuin University
5-10-1 Fuchinobe, Chuo-ku, Sagamihara-shi, Kanagawa, Japan
kzk0520.cb@gmail.com

Kouzou Ohara
Aoyama Gakuin University
5-10-1 Fuchinobe, Chuo-ku, Sagamihara-shi, Kanagawa, Japan
ohara@it.aoyama.ac.jp

ABSTRACT

Recent studies reveal that the covariance matrix adaptation evolution strategy (CMA-ES) updates the parameters based on the natural gradient. The rank-based weight is considered the result of the quantile-based transformation of the objective value and the parameters are adjusted in the direction of the natural gradient estimated by Monte-Carlo with the samples drawn from the current distribution. In this paper, we propose a sample reuse mechanism for the CMA-ES. On the basis of the importance sampling, the past samples are reused to reduce the estimation variance of the quantile and the natural gradient. We derive the formula for the rank-μ update of the covariance matrix and the mean vector update using the past samples, then incorporate it into the CMA-ES without the step-size adaptation. From the numerical experiments, we observe that the proposed approach helps to reduce the number of function evaluations on many benchmark functions, especially when the number of samples at each iteration is relatively small.

Categories and Subject Descriptors

G.1.6 [**Numerical Analysis**]: Optimization—*Global optimization*

Keywords

Importance Sampling; Natural Gradient; Covariance Matrix Adaptation Evolution Strategy; Information Geometric Optimization

1. INTRODUCTION

The covariance matrix adaptation evolution strategy (CMA-ES) [6, 7] is a comparison-based stochastic search algorithm for continuous optimization. The CMA-ES controls the mean vector and the covariance matrix of the multivariate normal distribution, from which candidate solutions are sampled. At each iteration it samples candidate solutions, then they are evaluated on the problem and their ranking is computed. The parameter of the distribution $\mathcal{N}(m, \sigma^2 C)$ are updated by using the candidate solutions and their ranking, so that the candidate solutions tend to converge towards the optimum of the problem at a geometric rate. Since it does not explicitly use the objective value, the CMA-ES is invariant to the order-preserving transformation of the objective function, $f \mapsto g \circ f$, where $f : \mathbb{R}^d \to \mathbb{R}$ is an objective function and $g : \mathbb{R} \to \mathbb{R}$ is any increasing function. Invariances to the order-preserving transformation of the objective function and to the linear transformation of the search space enable the CMA-ES to efficiently solve non-convex, non-separable and ill-conditioned problems [8].

Recent studies [2,5,10] reveal that some components of the CMA-ES, namely m-update and the rank-μ C-update, are regarded as the stochastic natural gradient ascent [4]. Ollivier et al. [10] have proposed a generalized framework of the stochastic search method for black-box optimization problems on an arbitrary search space, called information geometric optimization (IGO). The IGO framework includes not only the rank-μ update CMA-ES but also a binary optimization algorithm called population-based incremental learning (PBIL) as instantiations with different probability models. The IGO framework provides a mathematical interpretation of the weighted recombination, based on which we develop a novel technique in this paper.

In the IGO framework, the objective function (to be minimized, w.l.o.g.) is transformed into the utility function on the basis of the quantile of the objective value under the sampling distribution. The objective of the parameter update is considered to maximize the expected utility function over the sampling distribution, and the parameters are updated in its steepest ascent direction. With respect to the KL-divergence, the steepest ascent direction is known to be given by the so-called natural gradient [4], which is computed by the product of the inverse Fisher information matrix and the gradient. The natural gradient is estimated by Monte-Carlo using the candidate solutions sampled from the current distribution. Larger sample size typically results in higher accuracy (lower variance) of the natural gradient estimate, requires less iterations, but it costs more function evaluations.

GECCO '15, July 11 - 15, 2015, Madrid, Spain

DOI: http://dx.doi.org/10.1145/2739480.2754704

To accurate the natural gradient estimate without any additional function evaluations and accelerate the search, we consider *reusing* the candidate solutions of past iterations. In the CMA-ES and the IGO setting, the samples drawn at past iterations are discarded. If we just reuse the samples from past iterations without the modification of the update, it will introduce a bias to the Monte-Carlo estimate of the natural gradient since the past distributions are different from the current distribution. However, if the past distributions are close to the current distribution, the past samples can be useful to estimate the natural gradient. Reusing the past samples is natural and widely used in the evolutionary computation, for instance, the $(\mu + \lambda)$-ES, though their purpose differs from ours.

In this paper, we propose a sample reuse mechanism for the CMA-ES based on the importance sampling. We derive the formula to estimate the utility values (recombination weights) and the natural gradient by using the samples from the mixture of the current and past distributions. Theoretically it does not introduce any bias in the estimation and is expected to reduce the estimation variance. We derive the formula of the rank-μ update of the covariance matrix and the mean vector update with the sample reuse, then incorporate rank-one update into this mechanism.

The rest of the paper is organized as follows. In Section 2 we overview the CMA-ES, the IGO, and an existing sample reuse mechanism called importance mixing [13, 14]. In Section 3, we introduce the importance sampling in the IGO framework and derive the update formula for the rank-μ update CMA-ES. In Section 4, we evaluate the performance of the CMA-ES with and without the proposed sample reuse mechanism. We also compare the proposed strategies with the importance mixing. In Section 5, we summarize the contribution and discuss the future work.

2. RELATED ALGORITHMS

2.1 The CMA-ES without CSA

The covariance matrix adaptation evolution strategy (CMA-ES) [6, 7] maintains the Gaussian distribution $\mathcal{N}(m, \sigma^2 C)$, from which the candidate solutions are sampled. It repeats sampling, evaluation and update of the parameters of the distribution. At each iteration the CMA-ES generates the λ candidate solutions x_i for $i = 1, \cdots, \lambda$. Their objective values are evaluated and the ranking of each solution is computed. The parameters of the Gaussian distribution is updated by using the current candidate solutions and their ranking. The Gaussian distribution in the CMA-ES is parameterized with a mean vector $m \in \mathbb{R}^d$ and a covariance matrix $\sigma^2 C$, where $\sigma \in \mathbb{R}_>$ is called global step-size and $C \in \mathbb{R}^{d\times d}$ is a positive-definite symmetric matrix. These parameters are updated by the different update rules, the weighted recombination for m, the rank-μ and rank-one update for C, and the cumulative step-size adaptation (CSA) for σ. Here, we only explain the update rules of m and C since we consider the CMA-ES without CSA to investigate the basic effect of the proposed approach in this paper. Therefore, w.l.o.g., we assume $\sigma = 1$.

Let $\mathrm{rk}(x_i^t)$ be the ranking of x_i^t among the λ candidate solutions with respect to f. The weight is assigned to each solution according to its ranking. Each weight defined in [7] is

$$w_i = \frac{\max(0, \ln(\frac{\lambda+1}{2}) - \ln(i))}{\sum_{j=1}^{\lambda} \max(0, \ln(\frac{\lambda+1}{2}) - \ln(j))} \tag{1}$$

and the weight $w_{\mathrm{rk}(x_i^t)}$ is assigned to each solution x_i^t.

The mean vector m is updated by

$$m^{t+1} = m^t + c_m \sum_{i=1}^{\lambda} w_{\mathrm{rk}(x_i^t)}(x_i^t - m^t), \tag{2}$$

where c_m is the learning rate for m and usually $c_m = 1$. The update rule of the covariance matrix C consists of two components: rank-one update and rank-μ update. The rank-one update accelerates the update of the covariance matrix using the evolution path which is the cumulation of the consecutive steps. The evolution path for the rank-one update, p_c, is updated with the following formula

$$p_c^{t+1} = (1 - c_c)p_c^t + \sqrt{c_c(2 - c_c)\mu_{\mathrm{eff}}} \sum_{i=1}^{\lambda} w_{\mathrm{rk}(x_i^t)}(x_i^t - m^t), \tag{3}$$

where c_c is the cumulation parameter for the evolution path and $\mu_{\mathrm{eff}} = (\sum_{i=1}^{\lambda} w_i^2)^{-1}$. Then, the update rule of the covariance matrix C with the rank-one and rank-μ update is

$$\begin{aligned} C^{t+1} = C^t &+ \underbrace{c_1(p_c^{t+1}(p_c^{t+1})^{\mathrm{T}} - C^t)}_{\text{rank-one update}} \\ &+ \underbrace{c_\mu \sum_{i=1}^{\lambda} w_{\mathrm{rk}(x_i)} \left((x_i^t - m^t)(x_i^t - m^t)^{\mathrm{T}} - C^t\right)}_{\text{rank-}\mu\text{ update}}, \end{aligned} \tag{4}$$

where c_1 and c_μ are the learning rates of the rank-one and rank-μ updates for C, respectively. The rank-one update enlarges the eigenvalue of C corresponding to the direction of the evolution path, and the rank-μ update is considered the natural gradient ascent of the expectation of the transformed objective function as described in Section 2.2. The CMA-ES described above with $c_1 = 0$ is called the pure rank-μ update CMA-ES.

2.2 Information Geometric Optimization

Given a probability model $p_\theta(x)$ on $\mathbb{X}$ with the parameter $\theta \in \Theta$, the IGO [10] transforms the original minimization problem of $f : \mathbb{X} \to \mathbb{R}$ to the maximization of $J_{\theta^t} : \Theta \to \mathbb{R}$ at each iteration and performs the so-called natural gradient ascent. The function J_{θ^t} depends on the parameter θ^t at each iteration. It is the expected value of the utility $W_{\theta^t}^f(x)$ that is a nonlinear and non-increasing transformation of the objective function f.

The utility function[1], $W_\theta^f(x)$, is defined by using the quantile of $f(x)$ under $x \sim p_\theta$. Let $w : [0, 1] \to \mathbb{R}$ be a non-increasing function, $q_\theta^{\leqslant}(x) = P_\theta[y : f(y) \leqslant f(x)]$ represent the probability of sampling a better point than x from p_θ, and $W_\theta^f(x) = w(q_\theta^{\leqslant}(x))$. Then, $J_{\theta^t}(\theta)$ is defined as the expectation of the utility over the distribution p_θ, namely,

$$J_{\theta^t}(\theta) = \int W_{\theta^t}^f(x) p_\theta(x) \mathrm{d}x.$$

The natural gradient, $\tilde{\nabla} J_{\theta^t}(\theta)$, is given by the product of the inverse of the Fisher information matrix $F(\theta)$ and the vanilla gradient $\nabla J_{\theta^t}(\theta)$, i.e., the vector of the partial derivative w.r.t. each parameter, which can be written as

$$\tilde{\nabla} J_{\theta^t}(\theta) = \int W_{\theta^t}^f(x) \tilde{\nabla} l(\theta; x) p_\theta(x) \mathrm{d}x. \tag{5}$$

[1]This definition assumes the probability of getting the same objective value among different samples is 0. This assumption holds only if the Lebesgue measure of the level set of the objective function is 0, and it is no problem in most practical cases. The rigorous formulation that allows the same objective value is explained in [10].

Here, $\tilde{\nabla} l(\theta; x) = F^{-1}(\theta)\nabla l(\theta; x)$ is the natural gradient of the log-likelihood $l(\theta; x) = \ln p_\theta(x)$.

In practice, the integral in (5) cannot be computed analytically since the objective function is black-box, and have to be estimated by Monte-Carlo using the samples drawn from $p_{\theta^t}(x)$. Replacing the expectation in (5) with the arithmetic average over the samples $x_i \sim p_{\theta^t}$, we have

$$\tilde{\nabla} J_{\theta^t}(\theta^t) \approx \frac{1}{\lambda}\sum_{i=1}^{\lambda} W^f_{\theta^t}(x_i)\tilde{\nabla} l(\theta^t; x_i) \ .$$

We also need to estimate the utility $W^f_{\theta^t}(x_i) = w(q^{\leqslant}_{\theta^t}(x_i))$ for each x_i. Since $q^{\leqslant}_{\theta^t}(x_i)$ is the probability of sampling a better point than x_i from p_{θ^t}, it can be approximated by using the number of better point than x_i among the λ samples drawn from p_{θ^t}, namely $(\mathrm{rk}(x_i) - 1/2)/\lambda$. Then, the utility $W^f_{\theta^t}(x_i)$ is approximated by $\hat{w}(x_i) = w((\mathrm{rk}(x_i) - 1/2)/\lambda)$. With this approximation, we have an estimate of (5) at θ^t as $\tilde{\nabla} J_{\theta^t}(\theta^t) \approx \lambda^{-1}\sum_{i=1}^{\lambda} \hat{w}_i \tilde{\nabla} l(\theta^t; x_i)$. Then, we obtain the parameter update

$$\theta^{t+1} = \theta^t + \eta \sum_{i=1}^{\lambda} \frac{\hat{w}_i}{\lambda} \tilde{\nabla} l(\theta^t; x_i) \ , \tag{6}$$

where η denotes the learning rate and $\hat{w}_i = \hat{w}(x_i)$ for simplicity.

The pure rank-μ update CMA-ES ($c_1 = 0$ in (4)) is considered an instantiation of the IGO algorithm. Given the multivariate Gaussian distribution $\mathcal{N}(m, C)$ parameterized by $\theta = (m, C)$, the parameter update (6) in the IGO algorithm is equivalent to (2) and (4) if $c_m = c_\mu, c_1 = 0$, and any function w such that $w((i-1/2)/\lambda) = \lambda w_i$ for w_i defined in (1).

2.3 Importance Mixing

An idea of reusing previously generated solutions has been introduced in [13, 14], called importance mixing. It creates the set of points that can be considered as taken from the current distribution p_{θ^t}. Provided a population of λ points drawn from $p_{\theta^{t-1}}$, it accepts each point x in the population with probability $\min\{1, (1-\alpha)p_{\theta^t}(x)/p_{\theta^{t-1}}(x)\}$ into new population. Let $\lambda' \leqslant \lambda$ be the number of accepted points. Then, sample a point x from p_{θ^t} and accept it with probability $\max\{\alpha, 1 - p_{\theta^{t-1}}(x)/p_{\theta^t}(x)\}$ into the new population and repeat this procedure until $\lambda - \lambda'$ points are accepted. Then, the new population is considered as being distributed as p_{θ^t}. Since this idea is introduced to reduce the number of points for which the function value is evaluated at each iteration, the estimation variance of the natural gradient does not necessarily lessen. In Section 4, we compare the proposed strategies with the importance mixing. The parameter α called the minimal refresh rate is set to $\alpha = 0$ to reduce the function evaluations as much as possible in the experiments.

3. SAMPLE REUSE IN THE CMA-ES

In this section, we propose a sample reuse mechanism for the CMA-ES based on the importance sampling that is a technique to estimate the expectation over a probability distribution by using samples drawn from a different probability distribution. We mix the current and past samples as drawn from the mixture of the current and past distributions, and estimate the natural gradient by Monte-Carlo with the samples from the mixture.

3.1 Importance Sampling

Suppose that we have $K+1$ probability distributions p^k ($k = 0, 1, \ldots, K$) and λ independent samples x^k_i ($i = 1, \ldots, \lambda$) from each probability distribution p^k. Our objective here is to estimate the expectation $\int g(x)p^0(x)\mathrm{d}x$ by using $\lambda(K+1)$ samples x^k_i. In the CMA-ES, p^0 and p^k corresponds to the current distribution and the distribution of previous kth iteration. The simplest way is to apply the Monte-Carlo estimate by using only the samples x^0_i from p^0,

$$\frac{1}{\lambda}\sum_{i=1}^{\lambda} g(x^0_i) \tag{7}$$

implying that we discard all the other samples x^k_i ($k \geqslant 1$) despite that they may help to estimate the quantity more accurately. To utilize all the possible samples, we employ the idea of importance sampling.

Let $\bar{p}$ be the mixture $\bar{p}(x) = (K+1)^{-1}\sum_{k=0}^{K} p^k(x)$. One can rewrite the expectation as

$$\int g(x)p^0(x)\mathrm{d}x = \int g(x)\frac{p^0(x)}{\bar{p}(x)}\bar{p}(x)\mathrm{d}x \tag{8}$$

Considering the set of points x^k_i to be sampled from $\bar{p}$, we can approximate the RHS of the above equality by the average

$$\frac{1}{\lambda(K+1)}\sum_{k=0}^{K}\sum_{i=1}^{\lambda} g(x^k_i)\frac{p^0(x^k_i)}{\bar{p}(x^k_i)} \ . \tag{9}$$

It is an unbiased estimator of (8) shown in [11, 15], i.e., the expected value of (9) equals to (8). In this way, we can utilize all the possible samples without introducing any bias and expect that this leads to more accurate estimation than the simple Monte-Carlo using only x^0_i. This is the key idea of our work. Note that the case $K = 0$ recovers (7).[2]

3.2 Sample Reuse in the IGO

We introduce (9) in the IGO framework to estimate the natural gradient $\tilde{\nabla} J_{\theta^t}(\theta^t)$ using the samples x^{t-k}_i ($i = 1, \ldots, \lambda$) from the current and past K distributions, p^{t-k} ($k = 0, 1, \ldots, K$). Let $\bar{p}(x) = (K+1)^{-1}\sum_{k=0}^{K} p^{t-k}(x)$.

First, we need to estimate $W^f_{\theta^t}(x^{t-k}_i) = w(q^{\leqslant}_{\theta^t}(x^{t-k}_i))$. By definition, we can write $q^{\leqslant}_{\theta^t}(x^{t-k}_i)$ in the form of (8) with $g(x) = \mathbb{I}\{f(x) \leqslant f(x^{t-k}_i)\}$, where $\mathbb{I}\{\cdot\}$ indicates the indicator function. Then, we can apply the formula (9) to estimate $q^{\leqslant}_{\theta^t}(x^{t-k}_i)$ and we obtain

$$\sum_{l=0}^{K}\sum_{j=1}^{\lambda} \frac{\mathbb{I}\{f(x^{t-l}_j) \leqslant f(x^{t-k}_i)\}}{\lambda(K+1)}\frac{p^t(x^{t-l}_j)}{\bar{p}(x^{t-l}_j)} =: \bar{q}(x^{t-k}_i) \ . \tag{10}$$

Given w, the estimate $\hat{w}^{t-k}_i$ of the weight value $W^f_{\theta^t}(x^{t-k}_i)$ for each x^{t-k}_i is computed as

$$\hat{w}^{t-k}_i = w\left(\bar{q}(x^{t-k}_i)\right) \ . \tag{11}$$

Next we estimate the natural gradient (5) using (11). The natural gradient (5) at θ^t is of the form (8) with $g(x) = W^f_{\theta^t}(x)\tilde{\nabla} l(\theta^t; x)$.

[2] One can create a different unbiased estimator that uses all the possible samples. Instead of considering the mixture $\bar{p}$, one computes $K+1$ unbiased estimator $\lambda^{-1}\sum_{i=1}^{\lambda} g(x^k_i)p^0(x^k_i)/p^k(x^k_i)$ and averages them, resulting in

$$\lambda^{-1}(K+1)^{-1}\sum_{k=0}^{K}\sum_{i=1}^{\lambda} g(x^k_i)p_0(x^k_i)/p_k(x^k_i).$$

This is also an unbiased estimator of (8). We can prove that (9) has a smaller estimation variance than this estimator, which is also claimed by [11, 12]. Therefore, we employ (9).

With an approximation of $W^f_{\theta^t}(x_i^{t-k})$ for each i and k defined in (11), we can apply the formula (9) and obtain the estimate of $\tilde{\nabla} J_{\theta^t}(\theta^t)$ as

$$\frac{1}{\lambda(K+1)} \sum_{k=0}^{K} \sum_{i=1}^{\lambda} \hat{r}(x_i^{t-k}) \tilde{\nabla} l(\theta^t; x_i^{t-k}) \ , \tag{12}$$

where $\hat{r}(x_i^k) = \hat{w}_i^k p_{\theta^t}(x_i^k)/\bar{p}(x_i^k)$. We remark that the proposed estimator (12) as well as the original estimator (the second term on the RHS of (6)) is biased; its expectation differs from (5) since the same samples are used to estimate $W^f_{\theta^t}$ and $\tilde{\nabla} J_{\theta^t}$, but they are consistent; the bias vanishes and both estimates converge to (5) w.p.1 as $\lambda \to \infty$. See Remark 2 of [1] for the impact of the bias.

3.3 Sample Reuse in the CMA-ES

We introduce the importance sampling to the CMA-ES described in Section 2.1. More precisely, we replace the mean vector update (2) and the rank-μ update of the covariance matrix in (4) with novel ones using the samples from the current and past K iteration.

As discussed in Section 2.2, the pure rank-μ update CMA-ES is considered as an instantiation of the IGO algorithm with the multivariate Gaussian distribution. Therefore, we can simply replace the second term on the RHS of (2) and the rank-μ update of the covariance matrix in (4) with the novel natural gradient estimate (12). All we need is to choose the function w. We choose $w(s) = -2\ln(2s)\mathbb{I}\{s \leqslant 1/2\}$, since w_i defined in (1) is considered as an approximation of $w((i-1/2)/\lambda)$ for large λ. The weight value $\hat{w}_i^{t-k}$ in (12) is given by (11) with this w. Then, we obtain the update rules for the mean vector and the covariance matrix as

$$m^{t+1} = m^t + c_m \sum_{k=0}^{K} \sum_{i=1}^{\lambda} \frac{\hat{r}(x_i^{t-k})}{\lambda(K+1)} (x_i^{t-k} - m^t), \tag{13}$$

$$C^{t+1} = C^t + c_\mu \sum_{k=0}^{K} \sum_{i=1}^{\lambda} \frac{\hat{r}(x_i^{t-k})}{\lambda(K+1)} \cdot ((x_i^{t-k} - m^t)(x_i^{t-k} - m^t)^{\mathrm{T}} - C^t). \tag{14}$$

Note that if $K = 0$, (13) and (14) is equivalent to (2) and (4) with $c_1 = 0$, except that $\hat{w}_i^t/\lambda$ is different from w_i defined in (1). In the preliminarily experiments we have observed that the sum of $\hat{w}_i^t/\lambda$ tends to be smaller than one when $K = 0$, whereas the sum of w_i is always one. Its impact is observed in the next section.

We combine the rank-one update to (14). We simply add the second term on the RHS of (4) to (14). To update the evolution path, we solely uses the current samples and use $w_{\mathrm{rk}(x_i^t)}$ as the weight for each x_i^t rather than $\hat{w}_i^t$. It means, the evolution path is updated as it is done in (3). Since the evolution path itself accumulates the past information, the rank-one update is considered utilizing the past samples. Therefore, the proposed algorithm exploits the past information in different ways to update the covariance matrix, and we expect the synergy.

To implement the proposed method, one needs to keep the mean vectors and the covariance matrices of the past K distributions to compute the likelihood ratio $p_{\theta^t}(x_i^{t-k})/\bar{p}(x_i^{t-k})$, which requires $O(Kd^2)$ additional memory space. For efficient and numerically stable computation, we keep the log-likelihood of each point, $l_i^{k,l} = \ln p_{\theta^{t-k}}(x_i^{t-l})$, for each $i = 1, \ldots, \lambda$ and $k, l = 0, \ldots, K$, which requires $O(\lambda K^2)$ additional space. The likelihood ratio is then computed as $p_{\theta^t}(x_i^{t-k})/\bar{p}(x_i^{t-k}) = (K+1)(\sum_{l=0}^{K} \exp(l_i^{l,k} - l_i^{0,k}))^{-1}$. Then the computational complexity at each iteration is $O(\lambda K d^2)$.

4. EXPERIMENTS AND RESULTS

In this section, we evaluate the proposed approach on a standard benchmark functions. We consider the following four variants of the CMA-ES using the past solutions:

(A) Reuse-m, C The parameters are updated using (13) and (14).

(B) Reuse-C The covariance matrix is updated using (14), while the mean vector is updated by (2), only current samples are used and the rankings are computed among the current samples as in the standard CMA-ES.

(C) Reuse-m, C + rank-one The rank-one update is incorporated into Algorithm (A), as is described in Section 3.3.

(D) Reuse-C + rank-one The rank-one update is incorporated into Algorithm (B).

For these four variants, we observe the effect of K and the population size λ. Moreover, we compare them with the pure rank-μ update CMA-ES with and without importance mixing and the hybrid of the rank-one and rank-μ update.

4.1 Experimental Procedure

Our benchmark set consists of Sphere, Rosenbrock, Ellipsoid, Cigar, Ackley, Bohachevsky, Schaffer, and Rastrigin functions[3]. The global optimum is located at $[1, \cdots, 1]^{\mathrm{T}}$ for Rosenbrock function and $[0, \cdots, 0]^{\mathrm{T}}$ for the other functions, where the optimal value is 0. The problem dimension d of each benchmark function is set to 20 and 40.

The initial mean vector m^0 of the Gaussian distribution is drawn uniform randomly from $[a, b]^d$ for each run, and the covariance matrix is initialized by $C^0 = \sigma^2 I$, where $\sigma = (b-a)/2$. The range is set to $[1, 5]^d$ for Sphere, Ellipsoid, Cigar, and Rastrigin functions, $[-2, 2]^d$ for Rosenbrock function, $[1, 30]^d$ for Ackley function, $[1, 15]^d$ for Bohachevsky function, and $[10, 100]^d$ for Schaffer functions, according to [3]. The evolution path p_c is initialized to the zero vector.

Each run is terminated and regarded as success when the best objective value reaches smaller than 10^{-10}. On the other hand, the run is terminated and regarded as failure after $d \times 10^6$ function evaluations, or when the minimal eigenvalue of the covariance matrix reaches 10^{-30}, except for Schaffer function, where the minimal eigenvalue is set to 10^{-60}. For each setting we conduct 50 independent runs.

Different population sizes (sample size) λ and K are tested. For Sphere, Rosenbrock, Ellipsoid, and Cigar functions, $\lambda \in \{4 + \lfloor 3\ln d \rfloor, 0.2d, 0.4d, 0.8d, d, 2d, 4d, 8d, 16d\}$. For the other multimodal functions, $\lambda = 2\lfloor 2^{i/2-1} \lambda_{\mathrm{base}} \rfloor$ for $i = 0, 1, \cdots, 7$ with $\lambda_{\mathrm{base}} = 2\ln d, d, 2d, 10d$ for Ackley, Bohachevsky, Schaffer, and Rastrigin functions, respectively. In preliminary experiments we have observed the failure with high probability when $\lambda \leqslant \lambda_{\mathrm{base}}$. The number of iterations for sample reuse is $K = 0, 1, 3, 5, 7, 9$. As is mentioned in Section 3.3, $K = 0$ leads to parameter updates slightly different from the standard mean vector update and the rank-μ update.

Following [7], we set the learning rates as follows:

$$c_m = 1, \quad c_c = \frac{4 + \mu_{\mathrm{eff}}/d}{\lambda + 4 + 2\mu_{\mathrm{eff}}/\lambda}, \quad c_1 = \frac{2}{(d+1.3)^2 + \mu_{\mathrm{eff}}}, \quad c_\mu = \min\left(1.0 - c_1, \frac{2(\mu_{\mathrm{eff}} - 2 + 1/\mu_{\mathrm{eff}})}{(d+2)^2 + 2\mu_{\mathrm{eff}}/2}\right). \tag{15}$$

[3]The definition of each benchmark function can be found in [3, 9].

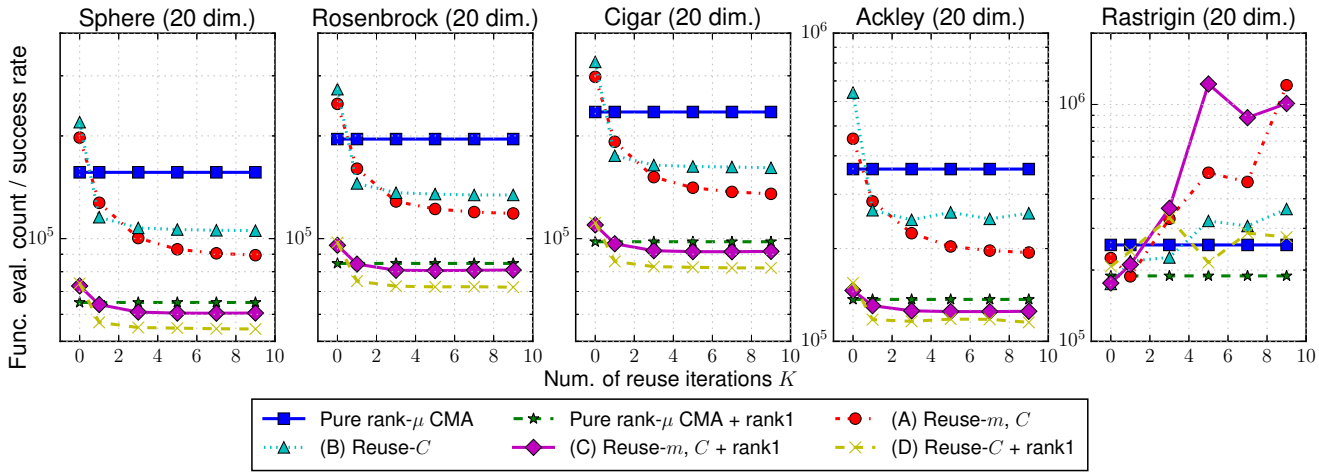

Figure 1: Average number of function evaluations divided by the success probability are shown on 20 dimensional problems. The default population size is used for Sphere, Rosenbrock, and Cigar functions and $\lambda = 2\lfloor 2\lambda_{\text{base}} \rfloor$ for Ackley and Rastrigin functions.

Note that for Algorithms (A) and (B), the learning rate for the rank-one update, c_1, is set to 0 and we do not update the evolution path.

4.2 Results and Discussion

Figure 1 shows the average number of function evaluations over the successful runs divided by the success probability on 20 dimensional functions for six algorithms: the pure rank-μ update CMA-ES ($c_1 = 0$ in Section 2.1), the hybrid update CMA-ES (described in Section 2.1), and Algorithms (A)–(D). The default population size $\lambda = 4 + \lfloor 3 \ln d \rfloor$ is used for Sphere, Rosenbrock, and Cigar functions and $\lambda = 2\lfloor 2\lambda_{\text{base}} \rfloor$ for Ackley and Rastrigin functions. The result on Ellipsoid function is very similar to that on Cigar function, and the results on Bohachevsky and Schaffer functions are similar to that on Ackley function. Moreover, we have observed a similar trends on K on the 40 dimensional functions. E.g., with the default population size, the pure rank-μ update CMA-ES and Algorithm (A) with $K = 5$ spend 7.5×10^5 and 3.8×10^5 function evaluations to solve 40 dimensional Sphere function, respectively. Their results are omitted due to the space limitation.

Figures 2 and 3 show the average number of function evaluations over the successful runs divided by the success probability on 20 dimensional problems. The performance of the pure rank-μ update CMA-ES, the pure rank-μ update CMA-ES with importance mixing, and the pure rank-μ update CMA-ES with sample reuse (Algorithm (A): Reuse-m, C with $K = 0, 1, 3, 5$) for different population sizes are shown in Figure 2. The results of the hybrid update CMA-ES, and the CMA-ES with sample reuse (Algorithm (D): Reuse-C with rank-one with $K = 0, 1, 3, 5$) for different population sizes are shown in Figure 3.

Figure 4 illustrates the transition of the sum of the products of the weight and the likelihood ratio $s_t^k = (1/\lambda) \sum_i \hat{r}(x_i^k)$ for the current ($k = t$) and past samples ($k = t-1, \ldots, t-3$) in a typical run. The results of Algorithm (A) and (B) with $K = 3$ and the default population size on 20 dimensional Sphere function are shown. Smaller values for $t - k$ iteration indicate either that the samples from $t - k$ iteration are away from the current distribution and the likelihood ratios are small, or that the samples from $t - k$ iteration are low in the ranking and small or zero weight values are assigned to them. Remark also that the sum of the coefficients does not amount to 1, as we have mentioned in Section 3.3.

Figure 5 displays the typical behaviors of Algorithm (A) with the default population size on 20 dimensional Sphere, Rosenbrock, and Ellipsoid functions and Algorithm (D) with $\lambda = 160$ on 20 dimensional Rosenbrock function. The results of both $K = 0$ and $K = 5$ are monitored.

4.2.1 *Impact of K*

We observe from Figure 1 that the performances of Algorithms (A)–(D) improve more or less monotonically as K increases for all but Rastrigin function. The algorithms with sample reuse (A and B or C and D) outperform the one without sample reuse (the pure rank-μ update CMA-ES or the hybrid update CMA-ES, respectively) already with $K = 1$, where the current and previous populations are used. We see the improvement over K up to 7 in Algorithm (A), whereas the improvement is observed for K up to 3 in Algorithms (B)–(D). The reason that the performance is saturated for large K is because the likelihood ratios for old samples tend to be small since past distributions are away from the current distribution and newer samples tend to have better function values, resulting that very small weights are assigned to very old samples. On Rastrigin function, a large K value tends to lead to low success rate. This defect is less visible when λ becomes larger as it is seen in Figures 2 and 3.

From Figure 5, the speedup by the sample reuse ($K = 5$ over $K = 0$) is seen not only at the convergence stage but also at the adaptation stage.

4.2.2 *Impact of λ*

When the population size is relatively large, we observe from Figures 2 and 3 that the sample reuse is not helpful very much to reduce the number of function evaluations. For large λ, we expect that the natural gradient estimate in (2) and (4) is relatively accurate and the effect of the sample reuse is less emphasized. Moreover, since the learning rate c_μ defined in (15) becomes larger as λ increases, leading to a big change of the probability distribution, the likelihood ratios for the past samples will be small. Therefore, the sample reuse does not help when λ is relatively large. We can observe the same effect of large λ for Algorithms (B) and (C) as well.

On Rosenbrock function, the sample reuse turns out to lead to failure in the case of $\lambda > 10^2$. As we see in Figure 5, the mean vector needs to move from the origin towards the global optimum.

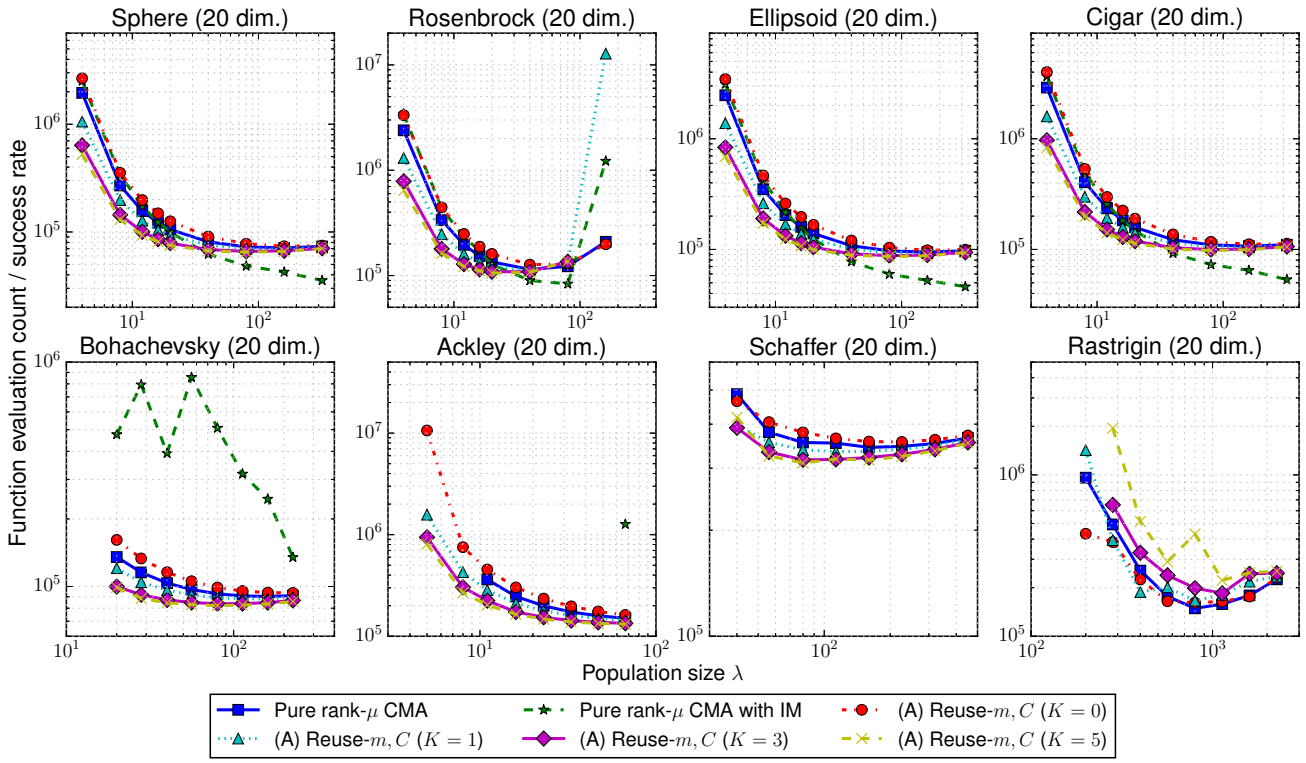

Figure 2: Average number of function evaluations divided by the success probability versus population size λ of the pure rank-μ update CMA-ES without and with importance mixing (IM), and the rank-μ update CMA-ES with sample reuse (Algorithm (A) for $K = 0, 1, 3, 5$) on 20 dimensional problems. Missing data implies the failure.

In the case of $\lambda = 160$ and $K = 5$, we observe that the eigenvalues of the covariance matrix tends to be smaller compared to the case of $K = 0$ during the movement of the mean vectors. Smaller eigenvalues of the covariance matrix leads to producing shorter steps, $x - m$, resulting in slow movement of the mean vectors.[4]

4.2.3 *Sample reuse in the pure rank-μ update*

Comparing Algorithms (A) and (B) in Figure 1, we find that Algorithm (B) is better for $K = 1$ but Algorithm (A) reaches better performance for larger K. The reason is observed in Figure 4, where the sums of the coefficients, $\hat{r}(x_t^k)$, which appears in (13) and (14), are more or less equal for $k = t, \ldots, t-3$ in Algorithm (A), whereas the sums for $k = t$ and $t-1$ amount to higher values than the ones for $k = t-2$ and $t-3$ in Algorithm (B). This indicates that the past samples are used to estimate the natural gradient in Algorithm (A) as well as the current samples, whereas mostly the current and previous samples are used and the older samples has less impact in Algorithm (B). In Algorithm (A), the mean vector is updated by using the proposed update (13), where the natural gradient is estimated with smaller variance than the one estimated in the original update (2). Then, the mean vector less fluctuates and the likelihood ratios are likely to be relatively large for the past samples.

[4]Empirically, we observe that a larger population size itself leads to smaller eigenvalues. Since more samples are used in the case of $K = 5$, compared to the case of $K = 0$, the reason of adapting a smaller covariance matrix may be the same reason.

4.2.4 *Sample reuse in the hybrid update*

Each variant of the algorithms is improved by introducing the rank-one update as seen in Figure 1. While the sample reuse for m update results in better performance when the covariance matrix is solely updated by the rank-μ update, the standard m update leads to faster convergence when the rank-one update is incorporated. As discussed in Section 4.2.3, the proposed m update (13) tends to have less fluctuation in a subspace that are less sensitive in function value such as the first axis for Cigar function. Then the evolution path will not be long in this subspace, while it should be in the example of Cigar function so that the covariance matrix learns the principle axis. The standard m update does not disturb the rank-one update and the speedup is achieved by the sample reuse in the rank-μ update.

Comparing the performance on Rastrigin function shown in Figures 2 and 3, we notice that Algorithm (D) is less suffered from a large value of K than Algorithm (A). This may be because the past samples have less effect in Algorithm (D) for the reason partially observed in Figure 4.

Figure 3 shows that smallest λ leads to the least performance difference between different K values in Algorithm (D) for the unimodal functions, while the difference between different K values becomes pronounced for smaller λ in Algorithm (A), shown in Figure 2. The learning rate c_1 for the rank-one update has a relatively large value for a small population size compared with the learning rate c_μ for the rank-μ update, then the impact of the rank-one update is dominative.

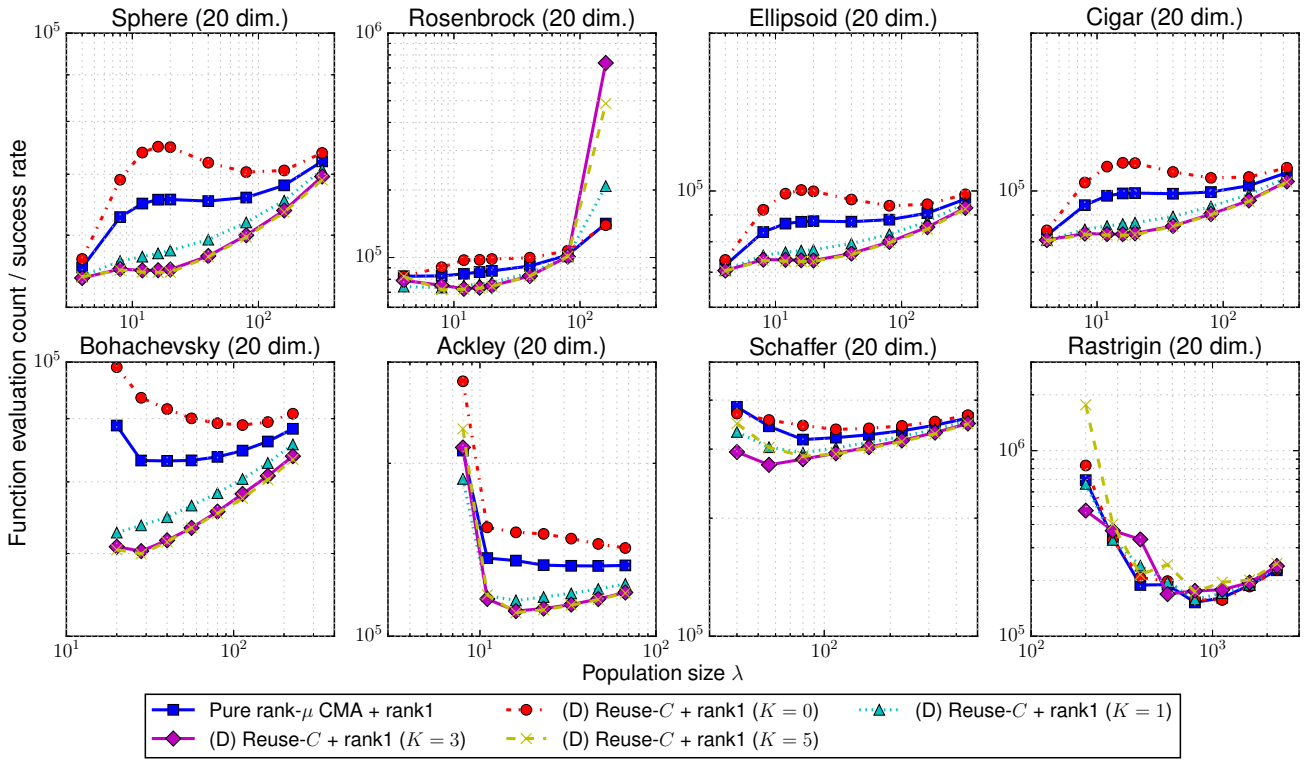

Figure 3: Average number of function evaluations divided by the success probability versus population size λ of the pure rank-μ update CMA-ES with rank-one update, and the CMA-ES with sample reuse (Algorithm (D) for $K = 0, 1, 3, 5$) on 20 dimensional problems.

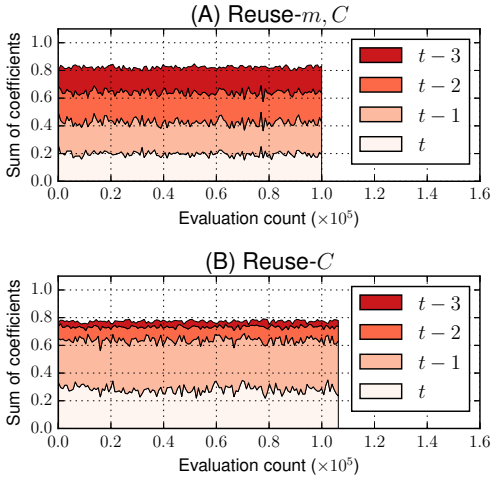

Figure 4: Transition of the sum of the coefficients (the products of the weight and the likelihood ratio) for the current and past samples, $s_t^k = 1/\lambda \sum_i \hat{r}(x_i^k), k = t, \ldots, t-3$, on 20 dimensional Sphere function for Algorithm (A) and (B) with $K = 3$ and the default population size. The moving average with window size 11 is computed for smoothing.

4.2.5 *Comparison with importance mixing*

When introducing the importance mixing in the pure rank-μ update CMA-ES, the performance on the unimodal functions are improved. The improvement is emphasized when λ is relatively large as is seen in Figure 2. The performance is better than Algorithm (A) for $\lambda \geqslant 320$. On the other hand, it is likely to be trapped by local minima on the multimodal functions. The motivation of the importance mixing is to reduce the number of function evaluations by reusing the sample and not to estimate the natural gradient more precisely. In fact, the estimate of the natural gradient will be less accurate since the number of samples drawn from the current distribution is less than λ.

Combining the rank-one update and the importance mixing, it ceases working properly when the refresh rate $\alpha = 0$. By tuning α, the unimodal functions can be solved. However, the α needs to be tuned dependently of λ. Moreover, it does not work well on the multimodal functions.

5. CONCLUSION

We have proposed the sample reuse technique based on the importance sampling to improve the accuracy of the estimate of the natural gradient in the CMA-ES. The quantile based utility function and then the natural gradient are estimated using the mixture of the current and last K populations. This has been realized thanks to the mathematical formulation of the rank-based selection proposed in the IGO framework. From the experiments, we have found that the sample reuse is effective both for the mean vector and the covariance matrix update in the pure rank-μ update CMA-ES. On the other hands, when the rank-one update is employed with rank-μ update (i.e., hybrid update), the sample reuse for the mean vector update disturbs the evolution path. The best performance is achieved by the variant that updates the mean vector without the sample reuse and combines the rank-one update and the rank-μ update using the sample reuse, where the reasonable choice for K seems to be $K \leqslant 3$.

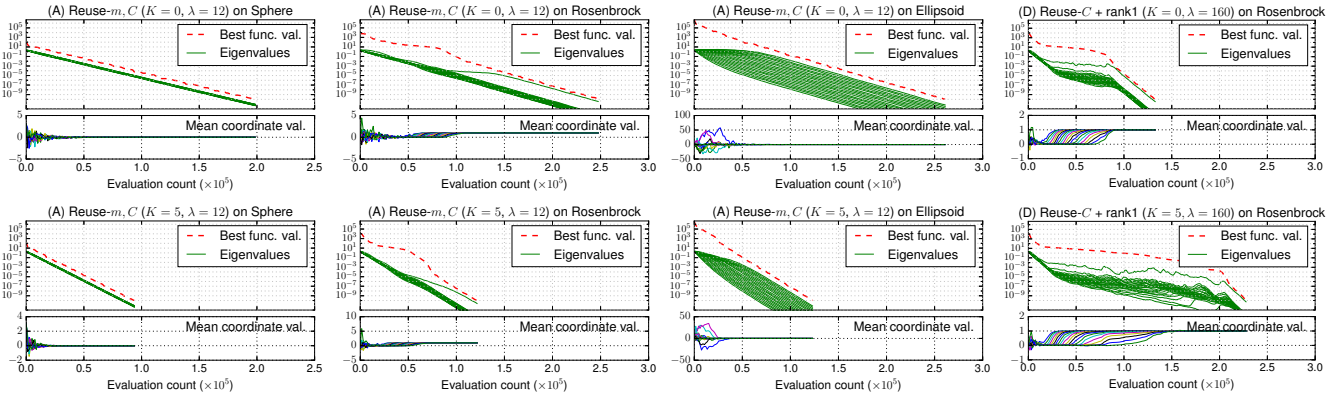

Figure 5: The best objective function value, eigenvalues of C, and each coordinate of m with $K = 0$ (top) and $K = 5$ (bottom) are shown for Algorithm (A) with the default population size on 20 dimensional Sphere (1st column), Rosenbrock (2nd column), and Ellipsoid (3rd column) functions and Algorithm (D) with $\lambda = 160$ on 20 dimensional Rosenbrock (4th column) function.

Larger K values tend to lead to failure on Rastrigin function. We have investigated the effect of K on 20 and 40 dimensional problems and observed a similar trend. More study on the effect of K should be done on higher dimensional problems.

The step-size adaptation is not introduced in the paper. In the CMA-ES, the step-size adaptation improves the efficiency and the robustness of the algorithm drastically. For instance, it can solve 20 dimensional Ellipsoid function within 2×10^4 function evaluations with the default population size, whereas the proposed algorithm needs at least 7×10^4 function evaluations. However, since this leads to a fast change of the distribution, the likelihood ratio for the past populations will be very small and the past population cannot contribute the natural gradient estimate. Therefore, we expect that the proposed sample reuse mechanism, as it is, is not effective when the step-size adaptation is employed. We may need a modification to effectively reuse the past samples.

6. REFERENCES

[1] Y. Akimoto, A. Auger, and N. Hansen. Convergence of the Continuous Time Trajectories of Isotropic Evolution Strategies on Monotonic C^2-composite Functions. In *Problem Solving from Nature-PPSN XII*, pages 42–51, 2012.

[2] Y. Akimoto, Y. Nagata, I. Ono, and S. Kobayashi. Bidirectional Relation between CMA Evolution Strategies and Natural Evolution Strategies. In *Parallel Problem Solving from Nature –PPSN XI*, volume 6238 of *LNCS*, pages 154–163. Springer, 2010.

[3] Y. Akimoto, J. Sakuma, I. Ono, and S. Kobayashi. Functionally Specialized CMA-ES : A Modification of CMA-ES Based on the Specialization of the Functions of Covariance Matrix Adaptation and Step Size Adaptation. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '08)*, pages 479–486, 2008.

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

[5] T. Glasmachers, T. Schaul, Y. Sun, D. Wierstra, and J. Schmidhuber. Exponential Natural Evolution Strategies. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '10)*, pages 393–400, 2010.

[6] N. Hansen. The CMA Evolution Strategy: A Comparing Review. In J. A. Lozano, P. Larrañaga, I. n. Inza, and E. Bengoetxea, editors, *Towards a New Evolutionary Computation*, volume 192 of *Studies in Fuzziness and Soft Computing*, pages 75–102. Springer, 2006.

[7] 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 Designing Metaheustics*, Natural Computing Series, pages 145–180. Springer, 2014.

[8] N. Hansen, A. Auger, R. Ros, S. Finck, and P. Pošík. Comparing Results of 31 Algorithms from the Black-Box Optimization Benchmarking BBOB-2009. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '10) Companion*, pages 1689–1696, 2010.

[9] N. Hansen and A. Ostermeier. Completely Derandomized Self-Adaptation in Evolution Strategies. *Evolutionary Computation*, 9(2):159–195, 2001.

[10] Y. Ollivier, L. Arnold, A. Auger, and N. Hansen. Information-Geometric Optimization Algorithms: A Unifying Picture via Invariance Principles. *eprint arXiv:1106.3708v2*, 2011.

[11] C. R. Shelton. Policy Improvement for POMDPs Using Normalized Importance Sampling. Technical Report AI Momo 2001-002, MIT AI Lab., 2001.

[12] C. R. Shelton. Policy Improvement for POMDPs Using Normalized Importance Sampling. In *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence (UAI '01)*, pages 496–503, 2001.

[13] Y. Sun, D. Wierstra, T. Schaul, and J. Schmidhuber. Efficient Natural Evolution Strategies. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '09)*, pages 539–546, 2009.

[14] Y. Sun, D. Wierstra, T. Schaul, and J. Schmidhuber. Stochastic Search Using the Natural Gradient. In *Proceedings of the 26th International Conference on Machine Learning (ICML '09)*, pages 1161–1168, 2009.

[15] E. Veach and L. J. Guibas. Optimally Combining Sampling Techniques for Monte Carlo Rendering. In *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '95)*, pages 419–428, 1995.