# Adaptive Stochastic Natural Gradient Method for Optimizing Functions with Low Effective Dimensionality*

Teppei Yamaguchi, Kento Uchida, and Shinichi Shirakawa

Graduate School of Environment and Information Sciences,
Yokohama National University, Yokohama, Japan
yamaguchi-teppei-yc@ynu.jp, uchida-kento-nc@ynu.jp,
shirakawa-shinichi-bg@ynu.ac.jp

**Abstract.** Black-box optimization algorithms, such as evolutionary algorithms, have been recognized as useful tools for real-world applications. Several efficient probabilistic model-based evolutionary algorithms, such as the compact genetic algorithm (cGA) and the covariance matrix adaptation evolution strategy (CMA-ES), can be regarded as a stochastic natural gradient ascent on statistical manifolds. Our baseline algorithm is the adaptive stochastic natural gradient (ASNG) method which automatically adapts the learning rate based on the signal-to-noise ratio (SNR) of the approximated natural gradient. ASNG has shown effectiveness in a practical application, but the convergence speed of ASNG deteriorates on objective functions with low effective dimensionality (LED), where LED means that part of the design variables is ineffective or does not affect the objective value significantly. In this paper, we propose an element-wise adjustment method for the approximated natural gradient based on the element-wise SNR and introduce the proposed adjustment method into ASNG. The proposed method suppresses the natural gradient elements with the low SNRs, helping to accelerate the learning rate adaptation in ASNG. We incorporate the proposed method into the cGA and demonstrate the effectiveness of the proposed method on the benchmark functions of binary optimization.

**Keywords:** probabilistic model-based black-box optimization · natural gradient · low effective dimensionality · learning rate adaptation.

## 1 Introduction

A lot of problems in real-world applications, such as the simulation-based optimization in engineering and the hyperparameter optimization in machine learning, are formulated as black-box optimization problems, that is, the gradient of

* This is an author version of the paper accepted to the 16th International Conference on Parallel Problem Solving from Nature (PPSN XVI). The final authenticated version is available online at https://doi.org/10.1007/978-3-030-58112-1_50.

the objective function cannot be accessed. The population-based black-box optimization methods, including evolutionary algorithms, have succeeded in a wide range of applications. In general, the performance of evolutionary algorithms depends on the choice of the hyperparameter, for example, the population size and learning rate. Tuning such hyperparameters in real-world applications is not realistic because the number of function evaluations is usually limited, and the computational cost for function evaluations is expensive. Therefore, a robust parameter adaptation mechanism is required in practical situations.

Probabilistic model-based evolutionary algorithms [3,5,6] are promising black-box optimization methods that define a parametric probability distribution on the search space and iteratively update the parameters of the distribution to improve the objective function value of the samples generated from the distribution. Several efficient probabilistic model-based evolutionary algorithms, such as the compact genetic algorithm (cGA) [6] or the covariance matrix adaptation evolution strategy (CMA-ES) [5], can be regarded as a stochastic natural gradient ascent on statistical manifolds [9]. In this paper, we focus on the adaptive stochastic natural gradient (ASNG) method [1] that adapts the learning rate in the stochastic natural gradient methods such as the cGA. In the black-box optimization, the natural gradient has to be estimated by the Monte Carlo approximation. ASNG measures the reliability of the estimated natural gradient direction by means of the signal-to-noise ratio (SNR) and tries to keep the SNR around a constant value by controlling the learning rate. The literature [1] shows that ASNG can achieve an efficient and robust optimization in the problem of a one-shot neural architecture search for deep learning. However, the convergence speed of ASNG becomes slow on objective functions with low effective dimensionality (LED), in which part of the variables is ineffective or does not affect the objective value significantly. Particularly, when redundant variables exist, the learning rate becomes smaller than necessary because of the low SNR due to the random walk in the redundant dimensions.

Because LED often appears in high-dimensional problems, including real-world applications [4,7,8], several works tackling LED exist. REMBO [11] projects a high dimensional search space into a low dimensional subspace using a random embedding to solve efficiently high-dimensional problems with LED by Bayesian optimization. REMEDA [10] applies the same idea to the estimation of distribution algorithm. However, the random projection does not reflect the landscape information, and the number of subspace dimensions still remains as a hyperparameter, which should be carefully chosen depending on the target problems.

In this paper, we propose a method for improving ASNG for objective functions with LED. When optimizing the objective functions with LED by ASNG, the reliabilities of the elements of the estimated natural gradient differ. Precisely, the SNR of the estimated natural gradient corresponding to the nonsensitive dimensions becomes small. Our idea is to adjust the update direction of the distribution parameters by exploiting the element-wise SNR of the natural gradient. The proposed method can be regarded as using the element-wise learning rate in ASNG, that is, the larger learning rate is adopted for the large SNR elements.

We incorporate the proposed method into ASNG without breaking its theoretical principal. The experimental results demonstrate that the proposed method, termed ASNG-LED, works efficiently on binary objective functions with LED.

## 2 Preliminaries

### 2.1 Stochastic Natural Gradient for Black-Box Optimization

We consider a black-box objective function $f : \mathcal{X} \to \mathbb{R}$ to be maximized on an arbitrary search space $\mathcal{X}$. To realize the black-box optimization, we introduce the technique called *stochastic relaxation* to transform the original problem into a differentiable objective function. Note that the following transformation is the same as the one considered in the information geometric optimization (IGO) framework [9] and natural evolution strategies [12].

Let us consider a parametric family of probability distributions $\mathcal{P} = \{P_\theta : \theta \in \Theta \subseteq \mathbb{R}^{D_\theta}\}$ on $\mathcal{X}$. The transformed objective function is the expectation of $f$ under $P_\theta$, that is,

$$J(\theta) := \int_{x \in \mathcal{X}} f(x) p_\theta(x) \mathrm{d}x = \mathbb{E}_{P_\theta}[f(x)] \ , \tag{1}$$

where $p_\theta$ is the density function of $P_\theta$ with respect to (w.r.t.) the reference measure $\mathrm{d}x$ on $\mathcal{X}$. For any $x$, we assume that the log-likelihood $\ln p_\theta(x)$ is differentiable w.r.t. $\theta$, and there exists a sequence of the distributions approaching the Dirac Delta distribution $\boldsymbol{\delta}_x$ around $x$. Then, the maximization of $J$ has the same meaning as the original problem in the sense that $\sup_{\theta \in \Theta} J(\theta) = \mathbb{E}_{\boldsymbol{\delta}_{x^*}}[f(x)] = f(x^*) = \sup_{x \in \mathcal{X}} f(x)$, where $x^*$ is the global optimal solution.

In this paper, we focus on the special case, as also assumed in [1], that $\mathcal{P}$ is represented by an exponential family of probability distributions whose density function is given as $p_\theta(x) = h(x) \exp(\eta(\theta)^{\mathrm{T}} T(x) - A(\theta))$, where $\eta : \Theta \to \mathbb{R}^{D_\theta}$ is the normal (canonical) parameter, $T : \mathcal{X} \to \mathbb{R}^{D_\theta}$ is the sufficient statistics, and $A(\theta)$ is the normalization factor. To make it simple, we assume $h(x) = 1$, and the parameter of the distribution is represented by $\theta = \mathbb{E}_{p_\theta}[T(x)]$, which is called the expectation parameter. Then, the natural gradient of the log-likelihood is given by $\tilde{\nabla} \ln p_\theta(x) = T(x) - \theta$, and the inverse of the Fisher information matrix is $\mathrm{F}(\theta)^{-1} = \mathbb{E}_{p_\theta}[(T(x) - \theta)(T(x) - \theta)^{\mathrm{T}}]$. We note that several known families of probability distributions, such as the Gaussian distribution and the Bernoulli distribution, are included in the exponential family and can be represented by the expectation parameter.

To maximize $J$, we consider the update of $\theta$ in the metric of the Kullback-Leibler (KL) divergence. Then, the steepest direction is given by the natural gradient direction [2] w.r.t. the Fisher metric defined by $\mathrm{F}(\theta)$. Because the natural gradient cannot be obtained analytically in the black-box optimization scenario, we approximate it by Monte Carlo with independent and identically distributed (i.i.d.) samples $x_1^{(t)} \cdots x_N^{(t)}$ from $P_{\theta^{(t)}}$. Moreover, we apply the utility transformation $u(x_i^{(t)})$ based on the ranking of $x_i^{(t)}$ w.r.t. $f$-value. As a result, the estimated

natural gradient is obtained as

$$G(\theta^{(t)}) = \frac{1}{N}\sum_{i=1}^{N} u(x_i^{(t)})(T(x_i^{(t)}) - \theta^{(t)}) \ . \tag{2}$$

Introducing the learning rate $\epsilon_\theta > 0$, the update rule of $\theta^{(t)}$ reads

$$\theta^{(t+1)} = \theta^{(t)} + \epsilon_\theta G(\theta^{(t)}) \ . \tag{3}$$

### 2.2 Adaptive Stochastic Natural Gradient (ASNG)

Akimoto et al. [1] developed the adaptive stochastic natural gradient (ASNG) method by introducing a learning rate adaptation mechanism into the stochastic natural gradient method using an exponential family of probability distributions with the expectation parameters. Although the joint optimization of the differentiable and non-differentiable variables is considered in [1], ASNG can apply to a naive black-box optimization without any modification. Here, we explain the outline of ASNG in the black-box optimization scenario.

In ASNG, the learning rate is represented by

$$\epsilon_\theta = \delta_\theta / \|G(\theta^{(t)})\|_{\mathrm{F}(\theta^{(t)})} \tag{4}$$

The update rule (3) with the learning rate in (4) is similar to the trust region method under the KL divergence with the trust region radius $\delta_\theta$. The adaptation of $\delta_\theta$ in ASNG is based on the theoretical insight of the stochastic natural gradient method. According to [1, Theorem 4], the monotonic improvement of $J$ is ensured when it holds

$$D_{\mathrm{KL}}(\theta^{(t+1)}, \theta^{(t)} + \epsilon_\theta \tilde{\nabla} J(\theta^{(t)})) \leq \zeta D_{\mathrm{KL}}(\theta^{(t)} + \epsilon_\theta \tilde{\nabla} J(\theta^{(t)}), \theta^{(t)}) \tag{5}$$

for some $\zeta > 0$ and if $\epsilon_\theta < (\zeta f(x^*) + J(\theta))^{-1}$ with some other mild assumptions, where $D_{\mathrm{KL}}(\theta, \theta')$ is the KL divergence between $P_\theta$ and $P_{\theta'}$. ASNG relaxes the condition for monotonic improvement to the improvement over $\tau \propto \delta_\theta^{-1}$ iterations. If $\epsilon_\theta$ is small enough, it allows the approximation where $\theta^{(t)} \approx \theta^{(t+k)}$ and $G(\theta^{(t+k)})$ is i.i.d. for $k = 0, \cdots, \tau - 1$. Then, approximation of the KL divergence with the Fisher metric allows the transformation of the condition in (5) as

$$\left\| \sum_{k=0}^{\tau-1} \frac{\epsilon_\theta G(\theta^{(t+k)}) - \epsilon_\theta \mathbb{E}[G(\theta^{(t)})]}{\sqrt{\tau}} \right\|^2_{\mathrm{F}(\theta^{(t)})} \leq \zeta \tau \epsilon_\theta^2 \|\mathbb{E}[G(\theta^{(t)})]\|^2_{\mathrm{F}(\theta^{(t)})} \ . \tag{6}$$

Then, the limitation of $\tau \to \infty$ in the LHS leads

$$\frac{\|\mathbb{E}[G(\theta^{(t)})]\|^2_{\mathrm{F}(\theta^{(t)})}}{\mathrm{Tr}(\mathrm{Cov}[G(\theta^{(t)})]\mathrm{F}(\theta^{(t)}))} \geq \frac{1}{\zeta\tau} \in \Omega(\delta_\theta) \ . \tag{7}$$

The LHS in (7) is the SNR of the estimated natural gradient measured w.r.t Fisher metric. The above discussion indicates that the SNR should be greater than a constant value proportional to $\delta_\theta$.

To estimate the lower bound of the SNR, the following accumulations were proposed in [1]:

$$s^{(t+1)} = (1-\beta)s^{(t)} + \sqrt{\beta(2-\beta)}\mathrm{F}(\theta^{(t)})^{\frac{1}{2}}G(\theta^{(t)}) \tag{8}$$

$$\gamma^{(t+1)} = (1-\beta)^2\gamma^{(t)} + \beta(2-\beta)\|G(\theta^{(t)})\|^2_{\mathrm{F}(\theta^{(t)})} \ , \tag{9}$$

where $s^{(0)} = \mathbf{0}$ and $\gamma^{(0)} = 0$. Finally, introducing hyperparameters $\alpha > 1$ and $\beta \propto \delta_\theta$, the adaptation of $\delta_\theta$ is written as

$$\delta_\theta \leftarrow \delta_\theta \exp\left(\beta\left(\frac{\|s^{(t+1)}\|^2}{\alpha} - \gamma^{(t+1)}\right)\right) \ . \tag{10}$$

This adaptation tries to maintain $\|s^{(t+1)}\|^2/\gamma^{(t+1)}$ around $\alpha$, which makes the lower bound of the SNR proportional to $\delta_\theta$. Note that the condition in (7) is satisfied under this design principle.

## 3 ASNG for Low Effective Dimensionality

ASNG increases the learning rate when the estimated SNR becomes large and decreases it when the SNR becomes small. We can intuitively regard the SNR as a measurement of the reliability of the natural gradient direction. If there are low-effective or redundant variables in the objective function, the variance of the natural gradient elements corresponding to such variables becomes large, resulting in the estimated SNR of $G(\theta^{(t)})$ and the learning rate becoming smaller than necessary. Therefore, the performance of ASNG deteriorates on objective functions with LED. To prevent this, we propose a natural gradient adjustment method for ASNG by using the element-wise SNR.

### 3.1 Estimation of Element-wise SNR

Let us denote $i$-th element of $G(\theta^{(t)})$ as $G_i^{(t)}$ for short. By using a one-hot vector $h_{(i)}$ whose $i$-th element is one and other elements are zero, we define the SNR of $G_i^{(t)}$ on the Fisher metric as follows:

$$\frac{\|\mathbb{E}[h_{(i)} \circ G(\theta^{(t)})]\|^2_{\mathrm{F}(\theta^{(t)})}}{\mathrm{Tr}(\mathrm{Cov}[h_{(i)} \circ G(\theta^{(t)})]\mathrm{F}(\theta^{(t)}))} = \frac{(\mathbb{E}[G_i^{(t)}])^2}{\mathrm{Var}[G_i^{(t)}]} \ . \tag{11}$$

To estimate the element-wise SNR, we accumulate $\hat{s}^{(t)}$ and $\hat{\gamma}^{(t)}$ similar to (8) and (9) as

$$\hat{s}^{(t+1)} = (1-\hat{\beta})\hat{s}^{(t)} + \sqrt{\hat{\beta}(2-\hat{\beta})}G(\theta^{(t)}) \tag{12}$$

$$\hat{\gamma}^{(t+1)} = (1-\hat{\beta})^2\hat{\gamma}^{(t)} + \hat{\beta}(2-\hat{\beta})G(\theta^{(t)}) \circ G(\theta^{(t)}) \ , \tag{13}$$

where $\hat{\beta} \in (0,1)$ is a smoothing factor and $\circ$ means the element-wise product. When the learning rate $\epsilon_\theta$ is small enough, we can consider $\theta^{(t+k)}$ stays around

the $\theta^{(t)}$ for $\tau \propto \epsilon_\theta^{-1}$ updates after $t$ iterations. Therefore, the expectations of $i$-th elements of $\hat{s}^{(t+1)}$ and $\hat{\gamma}^{(t+1)}$ are approximated under mild condition as

$$\mathbb{E}[\hat{s}_i^{(t+1)}] \approx \left(\sum_{k=0}^{\tau-1}(1-\hat{\beta})^k\right)\sqrt{\hat{\beta}(2-\hat{\beta})}\mathbb{E}[G_i^{(t)}] \overset{\tau\to\infty}{\longrightarrow} \sqrt{\frac{2-\hat{\beta}}{\hat{\beta}}}\mathbb{E}[G_i^{(t)}] \ , \tag{14}$$

$$\mathbb{E}[\hat{\gamma}_i^{(t+1)}] \approx \left(\sum_{k=0}^{\tau-1}(1-\hat{\beta})^{2k}\right)\hat{\beta}(2-\hat{\beta})\mathbb{E}[(G_i^{(t)})^2] \overset{\tau\to\infty}{\longrightarrow} \mathbb{E}[(G_i^{(t)})^2] \ . \tag{15}$$

Moreover, the variance of $\hat{s}_i^{(t+1)}$ is approximated under mild condition as

$$\mathrm{Var}[\hat{s}_i^{(t+1)}] \approx \left(\sum_{k=0}^{\tau-1}(1-\hat{\beta})^{2k}\right)\hat{\beta}(2-\hat{\beta})\mathrm{Var}[G_i^{(t)}] \overset{\tau\to\infty}{\longrightarrow} \mathrm{Var}[G_i^{(t)}] \ . \tag{16}$$

Therefore, the SNR of the $i$-th element of $G(\theta^{(t)})$ can be approximated as

$$\frac{(\mathbb{E}[G_i^{(t)}])^2}{\mathrm{Var}[G_i^{(t)}]} \approx \frac{\mathbb{E}[(\hat{s}_i^{(t+1)})^2]/\mathbb{E}[\hat{\gamma}_i^{(t+1)}] - 1}{2\hat{\beta}^{-1} - 1 - \mathbb{E}[(\hat{s}_i^{(t+1)})^2]/\mathbb{E}[\hat{\gamma}_i^{(t+1)}]} \approx \frac{\xi_i^{(t+1)}}{2\hat{\beta}^{-1} - 2 - \xi_i^{(t+1)}} \tag{17}$$

where $\xi_i^{(t+1)} := (\hat{s}_i^{(t+1)})^2/\hat{\gamma}_i^{(t+1)} - 1$.

### 3.2 Natural Gradient Adjustment Using the Element-wise SNR

The small value of the element-wise SNR estimated by (17) indicates that the corresponding natural gradient element is unreliable and has a large variance. We control the element-wise strength of $G(\theta^{(t)})$ using the element-wise estimation of the SNR; that is, we suppresses the strength of the natural gradient elements with the low SNRs. Because the denominator in (17) can be approximated by $2\hat{\beta}^{-1}$ when $\hat{\beta}$ is significantly small, the estimation of the SNR is approximately proportional to $\xi_i^{(t)}$. Therefore, we employ a $D_\theta$-dimensional adjustment vector $\sigma^{(t)}$ using $\xi_i^{(t)}$ and define $\sigma^{(t)}$ by the following sigmoid function as

$$\sigma_i^{(t)} = \frac{1}{1+\exp\left(-\omega\xi_i^{(t)}\right)} \ , \tag{18}$$

where $\omega$ ($> 0$) is the gain parameter. Then, we adjust the natural gradient direction by

$$H(\theta^{(t)}) = \sigma^{(t)} \circ G(\theta^{(t)}) \ . \tag{19}$$

When we set $\omega = \infty$, the sigmoid function coincides with the step function, and the natural gradient update with $H(\theta^{(t)})$ behaves as a dimensionality reduction method. We use the adjusted natural gradient $H(\theta^{(t)})$ in the proposed method instead of $G(\theta^{(t)})$, that is, the natural gradient $G(\theta^{(t)})$ in (3) and (4) is replaced by $H(\theta^{(t)})$.

### 3.3 Combination with ASNG

Let us assume $\mathrm{F}(\theta^{(t)})$ is given by a diagonal matrix; for example, the Bernoulli distribution satisfies this assumption. In such a case, we show the proposed method can be combined with ASNG without breaking the theoretical principle described in Section 2.2.

From the condition of monotonic improvement in (5), the search efficiency of the proposed method with ASNG is guaranteed in a similar way with that of ASNG. Let us assume $\hat{\beta}$ and $\epsilon_\theta$ are sufficiently small so that we can consider $\sigma^{(t)}$ and $\theta^{(t)}$ stay at the same point for $\tau \propto \delta_\theta^{-1}$ iterations. Then, we consider the relaxation of (5) in the same way as described in Section 2.2, which is given by

$$\left\| \sum_{k=0}^{\tau-1} \frac{\epsilon_\theta H(\theta^{(t+k)}) - \epsilon_\theta \mathbb{E}[G(\theta^{(t)})]}{\sqrt{\tau}} \right\|^2_{\mathrm{F}(\theta^{(t)})} \leq \zeta \tau \epsilon_\theta^2 \|\mathbb{E}[G(\theta^{(t)})]\|^2_{\mathrm{F}(\theta^{(t)})} \ . \tag{20}$$

When taking the limit of $\tau$ as $\tau$ goes to infinity, the LHS in (20) can be transformed and bounded from upper as

$$\epsilon_\theta^2 \sum_{i=1}^{D_\theta} \mathrm{F}_{ii}(\theta^{(t)}) \left( \mathrm{Var}[H_i(\theta^{(t)})] + \left( \frac{1-\sigma_i^{(t)}}{\sigma_i^{(t)}} \right)^2 \mathbb{E}[H_i(\theta^{(t)})]^2 \right) \tag{21}$$

$$\leq \epsilon_\theta^2 \mathrm{Tr}(\mathrm{Cov}[H(\theta^{(t)})]\mathrm{F}(\theta^{(t)})) + \epsilon_\theta^2 \|\mathbb{E}[H(\theta^{(t)})]\|^2_{\mathrm{F}(\theta^{(t)})} K^{(t)} \ , \tag{22}$$

where $K^{(t)} = \sum_{i=1}^{D_\theta} ((1-\sigma_i^{(t)})/\sigma_i^{(t)})^2$. Moreover, because $\sigma_i^{(t)} \leq 1$, we get

$$\|\mathbb{E}[\epsilon_\theta H(\theta^{(t)})]\|^2_{\mathrm{F}(\theta^{(t)})} \leq \|\mathbb{E}[\epsilon_\theta G(\theta^{(t)})]\|^2_{\mathrm{F}(\theta^{(t)})} \ . \tag{23}$$

As a result, we get the sufficient condition of (20) under the limitation of $\tau$ as

$$\frac{\|\mathbb{E}[H(\theta^{(t)})]\|^2_{\mathrm{F}(\theta^{(t)})}}{\mathrm{Tr}(\mathrm{Cov}[H(\theta^{(t)})]\mathrm{F}(\theta^{(t)}))} \geq \frac{1}{\zeta\tau - K^{(t)}} \geq \frac{1}{\zeta\tau} \in \Omega(\delta_\theta) \ , \tag{24}$$

where we assume $\tau$ satisfies $\tau > K^{(t)}/\zeta$. We note $K^{(t)}$ can be transformed as

$$K^{(t)} = \sum_{i=1}^{D_\theta} \left( \frac{1-\sigma_i^{(t)}}{\sigma_i^{(t)}} \right)^2 = \sum_{i=1}^{D_\theta} \exp\left( -2\omega \left( \frac{(\hat{s}_i^{(t)})^2}{\hat{\gamma}_i^{(t)}} - 1 \right) \right) \ . \tag{25}$$

Here, we consider the case that the accumulation of $\hat{s}_i^{(t)}$ and $\hat{\gamma}_i^{(t)}$ works ideally. Then, replacing $(\hat{s}_i^{(t)})^2$ and $\hat{\gamma}_i^{(t)}$ with their expectations given by (14), (15) and (16) approximates $K^{(t)}$ as

$$K^{(t)} \approx \sum_{i=1}^{D_\theta} \exp\left( -\frac{4\omega(1-\hat{\beta})}{\hat{\beta}} \frac{(\mathbb{E}[G_i^{(t-1)}])^2}{(\mathbb{E}[G_i^{(t-1)}])^2 + \mathrm{Var}[G_i^{(t-1)}]} \right) \leq D_\theta \ . \tag{26}$$

**Algorithm 1:** ASNG-LED

**Require:** $\theta^{(0)}$ {initial distribution parameter}
**Require:** $\alpha = 1.5$, $\delta_\theta^{\mathrm{init}} = 1$, $N = 2$, $\hat{\beta} = D_\theta^{-1}$
$t = 0$, $\Delta = 1$, $\gamma = 0$, $s = \mathbf{0}$, $\hat{\gamma}_i = 0$, $\hat{s}_i = 0$
**repeat**
  set $\delta_\theta = \delta_\theta^{\mathrm{init}}/\Delta$ and $\beta = \delta_\theta / D_\theta^{1/2}$
  compute $G_\theta(\theta^t)$ and $\sigma^{(t)}$ using (2) and (18)
  set $H(\theta^{(t)}) = \sigma^{(t)} \circ G(\theta^{(t)})$
  **for** $i = 1$ to $D_\theta$ **do**
    **if** $G_i^{(t)} \neq 0$ **then**
      $\hat{s}_i \leftarrow (1-\hat{\beta})\hat{s}_i + \sqrt{\hat{\beta}(2-\hat{\beta})} G_i^{(t)}/|G_i^{(t)}|$
      $\hat{\gamma}_i \leftarrow (1-\hat{\beta})^2 \hat{\gamma}_i + \hat{\beta}(2-\hat{\beta})$
    **end if**
  **end for**
  compute $\epsilon_\theta$ and $\theta^{(t+1)}$ by (4) and (3) replacing $G(\theta^{(t)})$ with $H(\theta^{(t)})$
  $s \leftarrow (1-\beta)s + \sqrt{\beta(2-\beta)} \mathrm{F}(\theta^{(t)})^{\frac{1}{2}} H(\theta^{(t)}) / \|H(\theta^{(t)})\|_{\mathrm{F}(\theta^{(t)})}$
  $\gamma \leftarrow (1-\beta)^2 \gamma + \beta(2-\beta)$
  $\Delta \leftarrow \Delta \exp(\beta(\gamma - \|s\|^2/\alpha))$
  $t \leftarrow t + 1$
**until** termination conditions are met

Thus, we can expect that $K^{(t)}$ becomes not so large and (24) is established when considering $\tau > D_\theta/\zeta$. Moreover, we can expect that $K^{(t)}$ becomes small when $\hat{\beta}$ is set as the small value.

Motivated from the above discussion, we modify the accumulation rule in (8) and (9) by replacing $G(\theta^{(t)})$ with $H(\theta^{(t)})$. Namely, the learning rate adaptation tries to make the lower bound of the SNR of $H(\theta^{(t)})$ proportional to $\delta_\theta$.

### 3.4 Implementation of ASNG-LED

Referring to [1], we replace $\mathrm{F}(\theta^{(t)})^{\frac{1}{2}} H(\theta^{(t)})$ and $\|H(\theta^{(t)})\|^2_{\mathrm{F}(\theta^{(t)})}$ in the accumulations of $s^{(t+1)}$ and $\gamma^{(t+1)}$ with $\mathrm{F}(\theta^{(t)})^{\frac{1}{2}} H(\theta^{(t)}) / \|H(\theta^{(t)})\|_{\mathrm{F}(\theta^{(t)})}$ and 1 for the stable updates. In the same manner, $\hat{s}_i^{(t+1)}$ and $\hat{\gamma}_i^{(t+1)}$ accumulate $G_i^{(t)}/|G_i^{(t)}|$ and 1 instead of $G_i^{(t)}$ and $|G_i^{(t)}|^2$. Because of this modification, $\hat{s}_i^{(t+1)}$ and $\hat{\gamma}_i^{(t+1)}$ are not updated when $G_i^{(t)} = 0$. We set the sample size $N$ as two and apply the ranking-based utility transformation introduced in [1], where $u(x_1^{(t)}) = 1$ and $u(x_2^{(t)}) = -1$ when $f(x_1^{(t)}) > f(x_2^{(t)})$ (and vice versa), and $u(x_1^{(t)}) = u(x_2^{(t)}) = 0$ when $f(x_1^{(t)}) = f(x_2^{(t)})$. This utility transformation can be generalized for an arbitrary sample size as follows: $u(x_i^{(t)}) = 1$ for best $\lceil N/4 \rceil$ samples, $u(x_i^{(t)}) = -1$ for worst $\lceil N/4 \rceil$ samples and $u(x_i^{(t)}) = 0$ otherwise. The proposed method implemented with ASNG, termed ASNG-LED, is summarized in Algorithm 1.

**Table 1.** The benchmark functions used in the experiment. They are $D$-dimensional functions of which only $d$ dimensions affect the function value.

| Name | LeadingOnes | OneMax | BinVal |
|---|---|---|---|
| Definition | $\sum_{i=1}^{d} \prod_{j=1}^{i} x_j$ | $\sum_{i=1}^{d} x_i$ | $\sum_{i=1}^{d} 2^{i-1} x_i$ |

# 4 Experiment

## 4.1 Experimental Setting

We evaluate ASNG-LED on several benchmark functions on $D$-dimensional binary domain. To simply demonstrate that ASNG-LED works well on functions with LED, we construct the benchmark functions with LED by injecting the redundant variables that do not affect the function value at all. The modified benchmark functions listed in Table 1 have $d$ ($\leq D$) effective dimensions and $D - d$ redundant (ineffective) dimensions. The optimal solutions on these functions are given by vectors whose first $d$ elements are given by 1. These functions become the same as the widely used binary benchmark functions when $D = d$. In LeadingOnes, the effective variables change during the optimization. All the variables of the OneMax function with $D = d$ have an equal contribution to the objective value, whereas the effects of the variables in BinVal significantly differ in each dimension. When $D > d$, these functions have redundant dimensions, and we particularly expect that ASNG-LED works well in such a situation.

In ASNG-LED, we set the strategy parameters as $\alpha = 1.5$, $\hat{\beta} = D_\theta^{-1}$, and $N = 2$. The gain parameter in (18) is set to $\omega = 1$. In our preliminary experiment, we observed that the impact of the setting of $\omega$ is not so significant if it is set to around one. We use the multivariate Bernoulli distribution, whose probability mass function is defined as $p_\theta(x) = \prod_{i=1}^{D_\theta} \theta_i^{x_i}(1-\theta_i)^{1-x_i}$, where $D_\theta = D$, to apply ASNG to binary optimization problems. The natural gradient of the log-likelihood and the Fisher information matrix are given by $\tilde{\nabla} \ln p_\theta(x) = x - \theta$ and a diagonal matrix with diagonal elements equal to $\mathrm{F}_{ii}(\theta) = \theta_i^{-1}(1-\theta_i)^{-1}$, respectively. We compare the proposed method to ASNG and the cGA. We use the default parameter setting proposed in [1] and set a sample size of two. The cGA can be regarded as a stochastic natural gradient with a sample size $N = 2$, which is derived by applying the family of Bernoulli distributions to the IGO framework. In other words, the cGA is an algorithm without the learning rate adaptation in ASNG. In our experiments, the learning rate of the cGA is fixed as $\epsilon_\theta = D^{-1}$. Moreover, we incorporate the margin of $D^{-1}$ into all methods as done in [1], i.e., the range of $\theta_i$ is restricted to $[D^{-1}, 1 - D^{-1}]$, to leave the probability of generating arbitrary binary vectors. We ran 50 independent trials for each method, where all algorithm settings succeeded in finding an optimal solution in all trials.

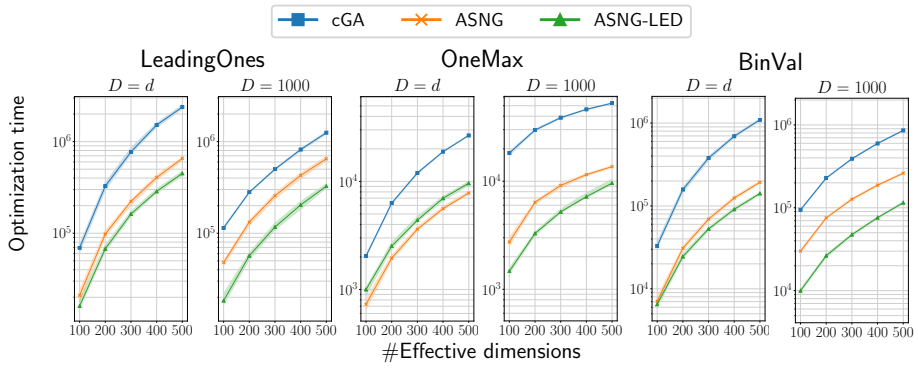


**Fig. 1.** Comparison of the optimization time (the number of function evaluations) on the benchmark functions with redundant dimensions ($D = 1000$) and without redundant dimensions ($D = d$). The median values and inter-quartile ranges over 50 trials are displayed.

### 4.2 Experimental Result and Discussion

**Optimization time:** We compare the search efficiency of ASNG-LED and baseline algorithms on the functions with and without redundant dimensions. We vary the number of effective dimensions as $d = 100, 200, 300, 400$, and 500, while the numbers of dimensions are set as $D = d$ and $D = 1000$ for each benchmark function. We call the former the benchmark functions without redundant dimensions and the latter the benchmark functions with redundant dimensions.

Figure 1 shows the relation between the number of effective dimensions and the optimization time, which is the number of iterations needed to find one of the optimal solutions. We observe that ASNG-LED can reduce the optimization time on the functions with redundant dimensions compared with ASNG. Also, the optimization time of ASNG-LED does not increase significantly on the functions with redundant dimensions compared with on the ones without redundant dimensions. The performance improvement against ASNG is highlighted in the case of the functions with redundant dimensions. We believe the reason is that ASNG-LED suppressed the strength of the estimated natural gradient corresponding to the redundant dimensions, allowing the adaptation mechanism for the learning rate to work effectively.

Focusing on LeadingOnes and BinVal, ASNG-LED outperforms ASNG on functions both with and without redundant dimensions. This is because part of the dimensions on these functions does not contribute to the function value significantly or at all during the optimization. More precisely, in LeadingOnes, any element following the first 0 in a binary string does not affect the function value at all. On BinVal, the weights for each bit are greatly different, and the lower parts of dimensions do not change the function value significantly. Because of such properties, ASNG-LED successfully adjusts the natural gradient estimates

and reduces the undesirable effect of the nonsensitive dimensions for the learning rate adaptation.

For the results of OneMax without redundant dimensions, we observe that ASNG-LED requires more optimization time than ASNG. OneMax is a linear function that has the same weight for each dimension. Therefore, adjusting the element-wise strength of the update direction does not provide a positive effect on the learning rate adaptation on OneMax. We note, however, the optimization time on OneMax is greatly shorter than the one on LeadingOnes and BinVal. Meanwhile, ASNG-LED performs better on OneMax when there is a large number of redundant dimensions. This is because the increased number of redundant dimensions makes the SNR value smaller in the original ASNG, and the performance of ASNG is degraded. In contrast, the accumulated value $\|s\|^2$ for learning rate adaptation in the proposed method does not become so small because of the modification of the accumulation rule.

We note that the performances of the cGA differ between the functions with and without redundant dimensions. This performance difference is caused by the different learning rate and margin settings of the distribution parameters described in Section 4.1.

**Optimization process on functions with and without redundant dimensions:** We apply each algorithm to both $D = 1000$ dimensional benchmark functions without redundant dimensions ($d = D$) and ones with $d = 500$ effective dimensions. Figure 2 shows the transitions of the normalized optimality gap $\phi(t)$, which is defined as $\phi(t) := 1 - f_{\mathrm{best}}(t)/f_{\mathrm{opt}}$, where $f_{\mathrm{opt}}$ and $f_{\mathrm{best}}(t)$ are the function values for an optimal solution and for the best so far solution at $t$-th iteration, respectively. From Figure 2, we can observe that ASNG-LED converges faster than ASNG except for OneMax without redundant dimensions ($D = d = 1000$). This implies that the adaptation mechanism of the proposed method works more efficiently than that of ASNG on the functions with redundant or insignificant dimensions.

To demonstrate how the modification in the proposed method accelerates the learning rate adaptation, we show the transitions of the trust-region radius $\delta_\theta$ on both OneMax without and with redundant dimensions in Figure 3. We observe that the transitions of the trust-region radius $\delta_\theta$ of ASNG and ASNG-LED are almost the same in OneMax without redundant dimensions, while $\delta_\theta$ of ASNG-LED is larger than $\delta_\theta$ of ASNG in OneMax with redundant dimensions. Because the exact natural gradient elements corresponding to the redundant dimensions are zero, the estimates of such elements behave like a random walk. Therefore, $\|s\|^2$ becomes small against $\gamma$ if the redundant dimensions exist, resulting in an unnecessarily small learning rate in ASNG. On the other hand, ASNG-LED tries to suppress the influence of redundant dimensions on the SNR value to avoid unnecessarily small learning rate on the adaptation mechanism.

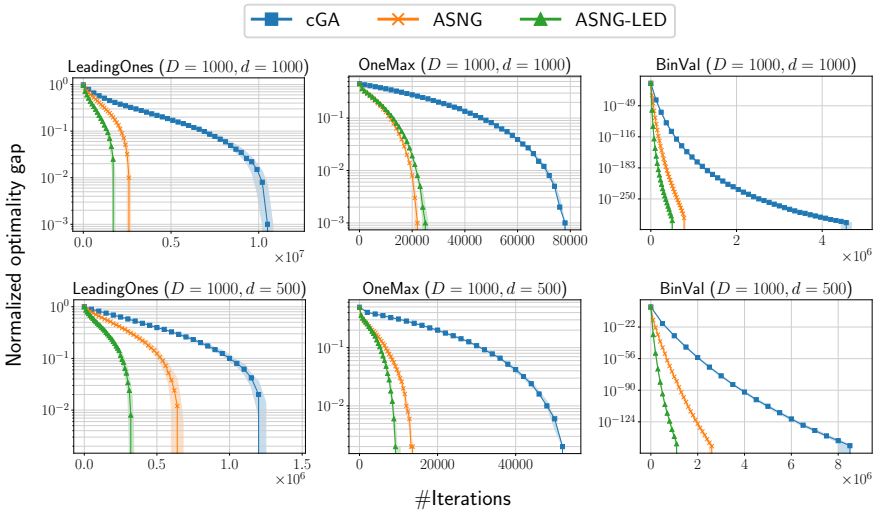


**Fig. 2.** Transitions of the optimality gap on the benchmark functions without redundant dimensions ($D = d = 1000$) and with redundant dimensions ($d = 500$). The median values and inter-quartile ranges over 50 trials are displayed.

## 5 Conclusion

We have proposed a method to improve the performance of ASNG on objective functions with LED. The proposed ASNG-LED adjusts the estimated natural gradient based on the elemental-wise SNR estimation. We confirmed the proposed adjustment can be combined with ASNG without breaking the theoretical aspect. We implemented ASNG-LED by applying the Bernoulli distribution and evaluated the performance on several benchmark functions on binary domain. The experimental results showed that ASNG-LED could accelerate the learning rate adaptation in ASNG and outperform the original ASNG on the functions with LED. In future work, ASNG-LED with the Gaussian distribution for continuous optimization should be implemented and evaluated. Also, the effectiveness of ASNG-LED should be verified on more realistic problems such as hyperparameter optimization and feature selection. The limitation of ASNG-LED is that it assumes the irrelevant directions are aligned with the axes, i.e., the performance of ASNG-LED will degrade by the rotation of the coordinate system. Making ASNG-LED rotational invariant is another important future work.

**Acknowledgments.** This work is partially supported by the SECOM Science and Technology Foundation and JSPS KAKENHI Grant Number JP20H04240.

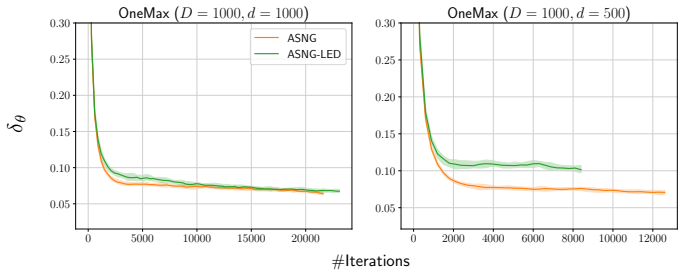


**Fig. 3.** Transition of the trust-region radius $\delta_\theta$ in OneMax with and without redundant dimensions in ASNG and ASNG-LED. The median values and inter-quartile ranges over 50 trials are displayed.

## References


1. Akimoto, Y., Shirakawa, S., Yoshinari, N., Uchida, K., Saito, S., Nishida, K.: Adaptive stochastic natural gradient method for one-shot neural architecture search. In: Proceedings of the 36th International Conference on Machine Learning (ICML). pp. 171–180 (2019)
2. Amari, S.: Natural gradient works efficiently in learning. Neural Computation **10**, 251–276 (1998)
3. Baluja, S., Caruana, R.: Removing the genetics from the standard genetic algorithm. In: Proceedings of the 12th International Conference on Machine Learning (ICML). pp. 38–46 (1995)
4. Bergstra, J., Bengio, Y.: Random search for hyper-parameter optimization. Journal of Machine Learning Research **13**, 281–305 (2012)
5. Hansen, N., Müller, S.D., Koumoutsakos, P.: Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). Evolutionary Computation **11**(1), 1–18 (2003)
6. Harik, G.R., Lobo, F.G., Goldberg, D.E.: The compact genetic algorithm. IEEE Transactions on Evolutionary Computation **3**, 287–297 (1999)
7. Hutter, F., Hoos, H., , Leyton-Brown, K.: An efficient approach for assessing hyperparameter importance. In: Proceedings of the 31st International Conference on Machine Learning (ICML). pp. 754–762 (2014)
8. Lukaczyk, T., Constantine, P., Palacios, F., Alonso, J.: Active subspaces for shape optimization. In: Proceedings of the 10th AIAA Multidisciplinary Design Optimization Conference. pp. 1–18 (2014)
9. Ollivier, Y., Arnold, L., Auger, A., Hansen, N.: Information-geometric optimization algorithms: A unifying picture via invariance principles. Journal of Machine Learning Research **18**, 1–65 (2017)
10. Sanyang, M.L., Kabán, A.: REMEDA: Random embedding EDA for optimising functions with intrinsic dimension. In: Proceedings of the 14th International Conference on Parallel Problem Solving from Nature. pp. 859–868 (2016)
11. Wang, Z., Hutter, F., Zoghi, M., Matheson, D., de Freitas, N.: Bayesian optimization in a billion dimensions via random embeddings. Journal of Artificial Intelligence Research **55**, 361–387 (2016)

12. Wierstra, D., Schaul, T., Glasmachers, T., Sun, Y., Peters, J., Schmidhuber, J.: Natural evolution strategies. Journal of Machine Learning Research **15**, 949–980 (2014)