Covariance Matrix Adaptation Evolution Strategy for Constrained Optimization Problem

Guidance

Professor Masao FUKUSHIMA
Associate Professor Nobuo YAMASHITA

Keiichirou HOSHIMURA

2005 Graduate Course

in

Department of Applied Mathematics and Physics

Graduate School of Informatics

Kyoto University

February 2007

Abstract

Recently engineers in many fields have faced solving complicated optimization problems. The objective functions of such problems are often nondifferentiable, or even if differentiable, their derivatives may not be calculated explicitly. Moreover, the problems are nonconvex in general, and hence, it is difficult to find the global optima. In order to overcome such difficulties, Hansen proposed the Covariance Matrix Adaptation Evolution Strategy (CMA-ES), which is an evolutionary algorithm generating a number of search points by using normal distribution. CMA-ES finds a global (or better local) minimum without using derivatives of the objective functions. However it is applicable only to the unconstrained problem.

In this paper we propose three CMA-ES type methods for the constrained optimization problem. These methods are based on the l_1-penalty method, and the differences of the methods are generating mechanism of search points. The first method generates search points by standard normal distribution. We note that the method is unapplicable to the problems whose objective functions are not defined out of the feasible region since the method sometimes generates a infeasible points. The second method generates search points by using lognormal distribution and the third method uses the projection onto the feasible region. Therefore the second and third methods always generate search points in the feasible region.

We compared these three methods by solving standard test problems. According to the results, the method based on the normal distribution is superior to the other methods for most problems. On the other hand, the method based on the projection showed better performance when many inequality constraints are active at a solution.

Contents

1 Introduction

In this paper, we consider the constrained optimization problem of the form

$$\begin{array}{lll} & \min & f(x) \\ (\mathrm{P}) & \text{s.t.} & h_j(x) = 0 \quad j = 1, \ldots, m \\ & & x_i \geq 0 \quad\;\; i \in S \end{array}$$

where $f : \Re^n \to \Re, h_j : \Re^n \to \Re$ $(j = 1, \ldots, m)$ and S is a subset of $\{1, \ldots, n\}$. We assume that neither continuity nor differentiability of f and h_j. Since a general nonlinear inequality constraints $g(x) \leq 0$ can be transformed into $x_{n+1} \geq 0$ and $g(x) + x_{n+1} = 0$ with a slack variable x_{n+1}, (P) does not lose any generality. In this paper we propose evolutional strategy algorithms for this problem (P) by using only function values of f and h_j.

Recently the engineers in many fields have faced solving optimization problems derived from complex systems. When the problem can be formulated with differentiable objective and constraint functions, we can apply efficient methods such as Newton-type methods, interior point methods and SQP method. In order to formulate such differentiable problems highly mathematical knowledge of modeling is required of users. Moreover in highly complicated systems, it is often difficult to construct mathematical models explicitly. For example, when some status of systems are evaluated by simulation, functions f or h_j of the status cannot be expressed in an explicit manner. For such a situation, we usually formulate approximate models to possess differentiability. However these approximate models may not reflect the original systems sufficiently and local minima of the models are sometimes far from local minima of the original models. Therefore we want algorithms that use only objective function values.

For such algorithms, Nelder-Mead method [8], pattern search method, Derivative Free Optimization (DFO) [2] have been proposed.

Since these methods use a local search technique, they usually converge rapidly. However these are not the method searching for a global minimum. Recently Metaheuristics [5, 7, 12] and Evolution Strategy with Covariance Matrix Adaptation(CMA-ES, [1, 3, 4]) have been much attention as global optimization methods. Metaheuristics, which include the Genetic Algorithm (GA) [7] and Particle Swarm Optimization (PSO), are the computation technique which searches near-optimal solutions in a practical time, instead of finding exact optimal solutions in very long time. The GA is based on the evolutionary mechanisms, and consists of combination and selection. The PSO imitates the swarm behavior of fish and bird, in which each particle shares certain information with other particle, and decides the next movement by the information.

The CMA-ES generates sets of search points according to the multivariate normal distribution, and finds the optimal solution by updating its mean and covariance matrix to displace the distribution.

It is a population based approach instead of point-to-point approach. Basically, the CMA-ES is similar to a usual local search, since it searches neighborhood of the current point and moves to the next point with minimum object function value. However, the CMA-ES has a potential to find a global minimum because of the mechanism of randomized search (like mutation in metaheuristics). The usual local search algorithms, such as Nelder-Mead method and the pattern search method, do not possess such mechanism. Moreover, since the covariance matrices updated

in the CMA-ES can be regarded as the inverse of the Hessian of the objective function, the CMA-ES takes into account quadratic information of objective function approximately. It is also reported in [11] that CMA-ES can find higher accuracy solution with fewer evaluations of objective values compared with GA and PSO.

Basically the CMA-ES has been considered for unconstrained optimization problem, and is not supposed applying to constrained optimization problems. In this paper we propose three CMA-ES based algorithms for solving constrained optimization problem efficiently. These methods based on the penalty method. The differences are generating mechanism of search points. The first method generates search points according to the normal distribution. The second one generates them with the lognormal distribution instead of normal distribution. The last one projects search points generated by using normal distribution onto the nonnegative orthant so that all search points are in the feasible region.

This paper is organized as follows. In Section 2 we describe the CMA-ES algorithm for unconstrained optimization problems. In Section 3 we propose three CMA-ES type algorithms for constrained optimization problems. In Section 4 we show the numerical results and discuss about them. In Section 5, we give the concluding remarks.

2 The Covariance Matrix Adaptation Evolution Strategy for Unconstrained Optimization Problems

In this section, we explain the CMA-ES for unconstrained optimization problem which is one of Evolution Strategy (ES [10, 6, 9]). Evolution Strategy is a search algorithm based on ideas of adaptation and evolution and was first proposed by Bienetk, Rechenberg and Schwefel. It can be classified into $(1+1)$-ES, (μ, λ)-ES and $(\mu+\lambda)$-ES. In $(1+1)$-ES, one point moves to another point in each iteration. On the other hand, in (μ, λ)-ES and $(\mu+\lambda)$-ES, one set moves to another set. In particular $(\mu+\lambda)$-ES chooses μ best points from the union of the original μ points and generated λ points, and (μ, λ)-ES chooses μ points from the generated λ points only. The CMA-ES belongs to (μ, λ)-ES type Evolution Strategy. The CMA-ES first generates the set of search points by exploiting a multivariate normal distribution, and then selects search points by objective function value.

Outline of the CMA-ES is written as follows:

Step 0. Generate the set of search points $x_1^{(g+1)}, \cdots, x_\lambda^{(g+1)}$ on iteration so that $x_k^{(g+1)} \sim N\left(m^{(g)}, (\sigma^{(g)})^2 C^{(g)}\right)$ for $k = 1, \cdots, \lambda$ by using mean value $m^{(g)}$, covariance matrix $C^{(g)}$ and stepsize $\sigma^{(g)}$,
Step 1. Select the best μ search points among $x_1^{(g+1)}, \cdots, x_\lambda^{(g+1)}$ according to their objective function value.
Step 2. Calculate $m^{(g+1)}$, $C^{(g+1)}$ and $\sigma^{(g+1)}$ from the selected μ points. Set g:=g+1, and return to Step 1.

Here,

$$x_k^{(g+1)} \sim N\left(m^{(g)}, (\sigma^{(g)})^2 C^{(g)}\right) \tag{1}$$

denotes that $x_k^{(g+1)}$ is chosen from the multivariate normal distribution with mean $m^{(g)}$ and covariance matrix $(\sigma^{(g)})^2 C^{(g)}$.

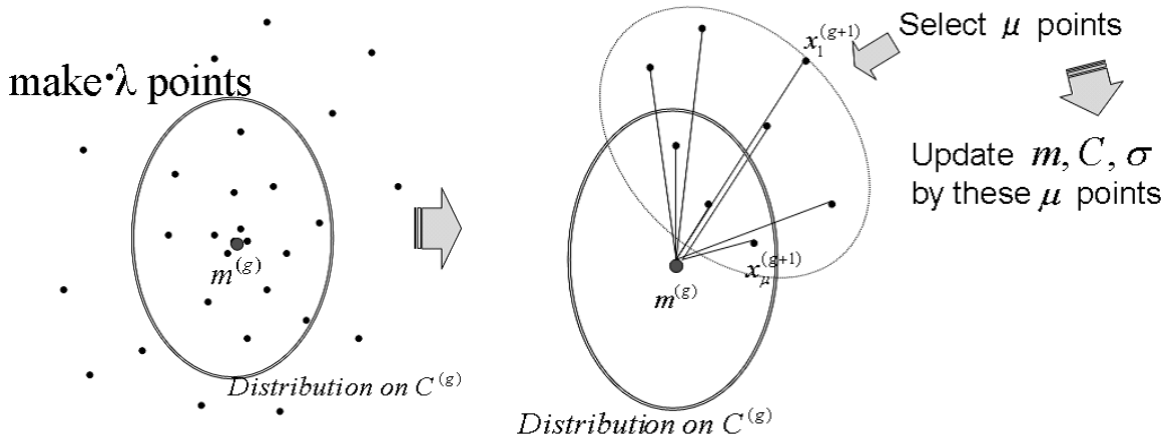

Figure 1: Outline on generation (g)

In CMA-ES, we generate set of search points $\{x_i^{(g+1)}\}(i = 1, \ldots, \lambda)$, mean $m^{(g+1)}$, covariance matrix $C^{(g+1)}$ and stepsize $\sigma^{(g+1)}$ for each iteration(Figure 1).

As a beginning of iteration g, population of new λ search points $x_1^{(g+1)}, \cdots, x_\lambda^{(g+1)}$ is generated in Step 0. Then in Step 1, best μ $(< \lambda)$ points are selected from this population. At last, we update $m^{(g+1)}$, $C^{(g+1)}$, $\sigma^{(g+1)}$ from selected μ points in Step 2. We repeat above steps until the optimal solution is found.

How to update $m^{(g+1)}, C^{(g+1)}$ and $\sigma^{(g+1)}$ has much influence to algorithm performance. A lot of update formula is proposed, but we would like to introduce [4].

At first, we describe update formula of mean value $m^{(g+1)}$. Here, we assume that λ search points $x_1^{(g+1)}, \cdots, x_\lambda^{(g+1)}$ is ranked as $f(x_1^{(g+1)}) \leq f(x_2^{(g+1)}) \leq \ldots f(x_\lambda^{(g+1)})$. We use a weighted average with weight parameter ω_i The update formula of $m^{(g+1)}$ is given by

$$m^{(g+1)} = \sum_{i=1}^{\mu} \omega_i x_i^{(g+1)} \tag{2}$$

where $\omega_i \in \Re_+,\ i = 1, \ldots, \mu$ is weight coefficients satisfying

$$\sum_{i=1}^{\mu} \omega_i = 1, \qquad \omega_1 \geq \omega_2 \geq \cdots \geq \omega_\mu \geq 0. \tag{3}$$

Next, the update formula of the covariance matrix $C^{(g+1)}$ is described. We estimate weighted $C^{(g+1)}$ from sampled population $x_1^{(g+1)}, \cdots, x_\mu^{(g+1)}$, weight coefficient ω_i and weighted average $m^{(g)}$ as follows:

$$\begin{aligned} C^{(g+1)} &= (1-\nu)C^{(g)} + \nu\left(1 - \frac{1}{\xi}\right)\sum_{i=1}^{\mu}\omega_i \left(\frac{x_i^{(g+1)} - m^{(g)}}{\sigma^{(g)}}\right)\left(\frac{x_i^{(g+1)} - m^{(g)}}{\sigma^{(g)}}\right)^T \\ &\quad + \frac{\nu}{\xi} p_c^{(g+1)} p_c^{(g+1)^T} \end{aligned} \tag{4}$$

where $0 < \nu \leq 1$ is the parameter called learning ratio, and by the ratio $(1-\nu) : \nu$, we use the information of the previous and current iterations. ξ is a choosing ratio of second term and third term.

The second term is the rank $\max(\mu, n)$, and which has the covariance matrix information from rank $\max(\mu, n)$ elements. This term exists for calculating more reliable covariance matrices by using as much information as possible. If μ is too large, we can get not only more reliable information but also the drawback as taking great deal of time for calculating. In this formula, the reason why we use not $m^{(g+1)}$ but $m^{(g)}$ for the update is given below. The covariance become smaller than $m^{(g)}$ if we use $m^{(g+1)}$ [3]. For the wide range searching we should adopt $m^{(g)}$ and make covariance matrix larger than previous iterations.

The third term $p_c^{(g+1)} p_c^{(g+1)^T}$ is rank one, and hence it has the information of covariance by rank one element. This term is available if μ is small. The reliable information from the rank one $p_c^{(g+1)}$ is given as below. From $p_c^{(g+1)}$, the information of mean trajectory in each iteration is available as a correlated information for update [3, 4]. $p_c^{(g+1)}$ is given as follows:

$$p_c^{(g+1)} = (1 - c_c) p_c^{(g)} + \zeta \frac{m^{(g+1)} - m^{(g)}}{\sigma^{(g)}} \tag{5}$$

where $c_c \leq 1$, and ζ is a constant selected as $p_c^{(g+1)} \sim N(0, C)$ [4, 3]. By using this correlated information, we can update covariance matrix reliably if population size λ is small.

Finally, we describe the update formula of stepsize $\sigma^{(g)}$. To control $\sigma^{(g)}$ we use correlation of mean trajectory by $p_\sigma^{(g)}$ given as below. $p_\sigma^{(g)}$ has similar structure to (4) and is described as

$$p_\sigma^{(g+1)} = (1 - c_\sigma) p_\sigma^{(g)} + \eta C^{(g)^{-1/2}} \frac{m^{(g+1)} - m^{(g)}}{\sigma^{(g)}} \tag{6}$$

where $c_\sigma < 1$ and η is selected as $p_{\sigma^{(g+1)}} \sim N(0, I)$. By using this, update formula of $\sigma^{(g)}$ is proposed as

$$\sigma^{(g+1)} = \sigma^{(g)} \exp\left(c_\sigma \left(\frac{\parallel p_\sigma^{(g+1)} \parallel}{\sqrt{n}} - 1 \right) \right). \tag{7}$$

The larger mean trajectory $m^{(g+1)} - m^{(g)}$ and $p_\sigma^{(g+1)}$ are, the larger $\sigma^{(g+1)}$ is. The smaller mean trajectory $m^{(g+1)} - m^{(g)}$ and $p_\sigma^{(g+1)}$ are, the smaller $\sigma^{(g+1)}$ is. In conclusion stepsize $\sigma^{(g+1)}$ increase and decrease by mean trajectory correlation. Therefore stepsize $\sigma^{(g+1)}$ is large at first, and become small after method begin to converge.

From above update formulae, the CMA-ES algorithm can be described as follows:

CMA-ES Algorithm

Step 0.Parameters setting
Input $c^{(0)}, m^{(0)}$. Set parameters $\lambda, \mu, \omega_{i=1\cdots\mu}, c_\sigma, d_\sigma, c_c, \mu_{cov}$ and c_{cov} to their default values.
Step 1.Initialization
Set $p_\sigma^{(0)} = 0, p_c\ (0) = 0$. Choose step size $\sigma^{(0)} \in \Re+$.
Step 2.Termination criterion If termination criterion met, then stop:
Step 3. New population sampling
$x_k^{(g+1)} \sim N\left(m^{(g)}, (\sigma^{(g)})^2 C^{(g)}\right) \quad for\ k = 1, \cdots, \lambda$
Step 4.Update Update mean value $m^{(g+1)}$ by (2) and (3).
Update stepsize $\sigma^{(g+1)}$ by (6) and (7).
Update Covariance matrix by (4) and (5).
Go to Step 2.

An advantage of the CMA-ES is that this can be applied to nondifferentiable problems. Hence, a CMA-ES can solve problems that the steepest descent method or the quasi Newton method can not be applied to. Another advantage is that the CMA-ES shows the better performance than metaheuristics like GA, PSO, etc., since the CMA-ES can take second order derivative into account approximately by using covariance matrix.

The disadvantage is that if we apply the CMA-ES to easy problems solved immediately by Newton type method, then the CMA-ES takes higher cost for calculating. Compared with Newton method which can obtain search direction by derivative, to obtain search direction, the CMA-ES have to generate search points and evaluate them for all problems. In addition, as the dimension become larger, the CMA-ES takes larger time for estimating $n \times n$ covariance matrix C. The covariance matrix estimation is most influential on performance of a CMA-ES. Therefore applying CMA-ES to large dimension problem shows poor performance.

We note that the CMA-ES shows poor performance for the problem whose optimal solution is on the boundary of constraints. To observe this disadvantage, we applied the CMA-ES to following two problems, and showed results in Table 1.

Problem A

$$f(x) = x_1^2 + x_2^2$$

Problem B

$$f(x) = \begin{cases} x_1^2 + x_2^2 & x_1 \geq 0, x_2 \geq 0 \\ x_1^2 + |x_2| & x_1 \geq 0, x_2 < 0 \\ |x_1| + x_2^2 & x_1 < 0, x_2 \geq 0 \\ |x_1| + |x_2| & x_1 < 0, x_2 < 0 \end{cases}$$

In Problem A, function f is smooth at the origin, while in Problem B f is nonsmooth.

Table 1: Evaluation on Problem A & Problem B

	Problem A	Problem B
Average number of function evaluation	287.4	504.6

It can be seen from the Table 1 that Problem B needs larger average number of function evaluation than Problem A. The reason of above result is that generating mean of search points

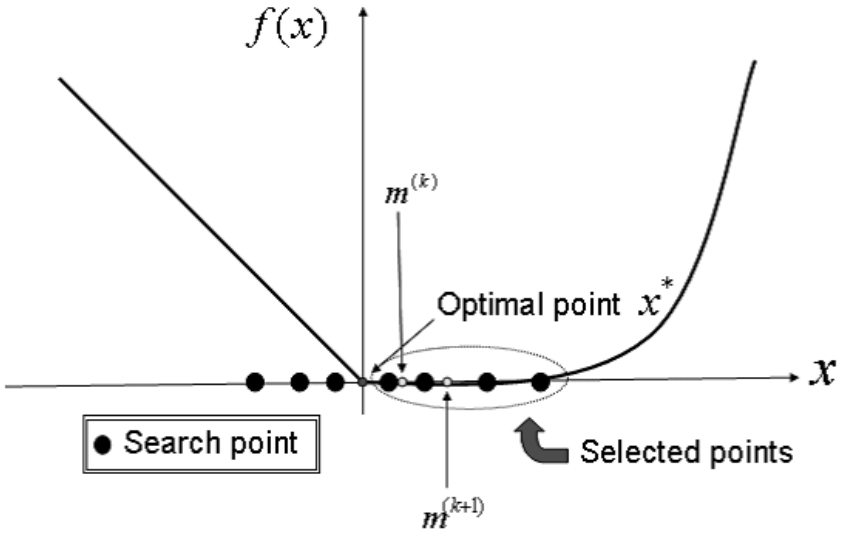

Figure 2: Search points and their averages on non-smooth function

makes bias if the optimal solution exists on the nonsmooth point. For example we describe this with $x_2 = 0$.

Figure 2 shows that CMA-ES generated 7 search points from mean $m^{(g)}$, and selected lower 4 points by objective function value. In this case, the optimal solution is $x_1 = 0$. Therefore the mean value $m^{(g+1)}$ should approach optimal solution $x_1 = 0$ nearer than $m^{(g)}$. Nevertheless generated $m^{(g+1)}$ moves away from $x_1 = 0$ actually. As a result, the method converges slower than the CMA-ES for smooth function.

3 Covariance Matrix Adaptation Evolution Strategy for Constrained Problem

Originally, CMA-ES is proposed for unconstrained optimization problem. In this section, we extend the CMA-ES for constrained optimization problem by using the penalty method. We propose the normal distribution method, the lognormal distribution method and the projection method.

3.1 Normal distribution method

We define the extended function $P_1(x; \rho)$ obtained by using l_1-penalty function to problem (P) as follows:

$$P_1(x; \rho) = f(x) + \rho \left\{ \sum_{j=1}^{m} |h_j(x)| + \sum_{i=1}^{n} \max(0, -x_i) \right\} \tag{8}$$

where ρ is a positive penalty parameter. The extended function $P(\cdot, \rho)$ is identical with the objective function f in feasible region because the penalty term of extended function becomes 0 in feasible region. We note that the extended function $P_1(\cdot, \rho)$ is nondifferentiable function.

An unconstrained optimization problem P1(ρ) using the extended function $P(x;\rho)$ is formulated as follows:

$$\begin{aligned} \text{P1}(\rho) \qquad & \min \quad P_1(x;\rho) \\ & \text{s.t.} \quad x \in R^n \end{aligned} \tag{9}$$

If ρ^k is sufficiently large, the optimal solution of P1(ρ^k) is identical with the optimal solution of (P). The penalty method is to generate an approximate solution of P1(ρ^k) sequentially, as changes penalty parameter as $\rho^k \to \infty$. For finding an approximate solution of P1(ρ^k), we apply the normally CMA-ES with normal distribution.

The normal distribution method

Step 0. Select penalty parameter $\rho_0 \in (0, \infty)$. Set mean $m^{(0)} \in \Re^n$, covariance matrix $C^{(0)} \in \Re^{n \times n}$ and stepsize $\sigma^{(0)} \in \Re$.
Step 1. Find an (approximate) optimal solution $\bar{x}^{(k)}$ of the subproblem P1(ρ^k) by applying the CMA-ES with $m^{(k)}$ and $C^{(k)}$
Step 2. If termination criterion is satisfied by $\bar{x}^{(k)}$, then stop.
Step 3. Select $\rho^{(k+1)} \in (\rho^{(k)}, \infty)$. Update $m^{(k+1)} = \bar{x}^{(k)}$ and $C^{(k+1)}$, set $k = k+1$, and go to Step 1.

By setting $m^{(k+1)} = \bar{x}^{(k)}$ we obtain search points from the optimal solution of last iteration. And in Step 3 we use $C^{(g)}$ as $C^{(k+1)}$. Here, $C^{(g)}$ is obtained by last iteration of Step 1 in the CMA-ES. As a result we expect faster convergence, since the CMA-ES can generate good search points by using the information of previous distribution's shape.

We can not apply this method to the problem whose objective function is not defined in $x_i < 0, i \in S$ since the CMA-ES may generate search points in the infeasible region.

3.2 Lognormal distribution method

Since the points of $x_i < 0$ are generated in the CMA-ES, the normal distribution method can be applied to problems whose objective function is not defined in $x_i < 0$ with $i \in S$. Therefore we propose generating search points by lognormal distribution instead of the normal distribution in the CMA-ES. We call this method CMA-ES with lognormal distribution. The figure of the probability density function of lognormal distribution is represented as Figure 3. On this distribution, the probability in negative region is 0.

For the simplicity of explanation we set $S = \{1, \ldots, n\}$ and describe the proposed method in detail. The CMA-ES with lognormal distribution generates search points $x^{(g),1}, \ldots, x^{(g),\lambda}$ as

$$x^{(g),j} \sim N_L(\mu_L, C_L) \quad j = 1, \ldots, \lambda \tag{10}$$

where $N_L(\mu_L, C_L)$ is lognormal distribution with mean μ_L, covariance matrix C_L.

Lognormal distribution has two advantages. The first one is that generated search points always satisfy nonnegative constraints. There are practical problems whose object function is not defined in negative region, but this method can also apply to these problems. The second one is that generated points become dense between the mean point and boundary of negative region (Figure 4). As a result, this method can search around boundary in detail, and we can expect this method converges faster.

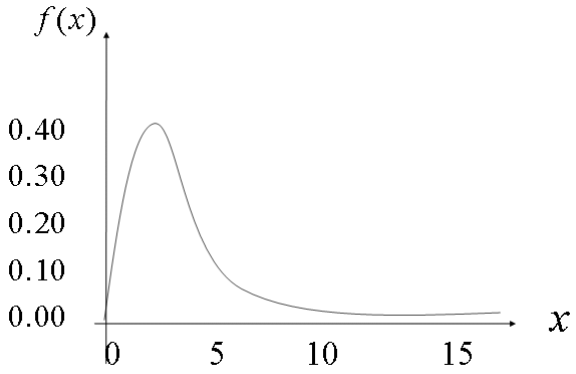

Figure 3: Shape of the probability density function of lognormal distribution

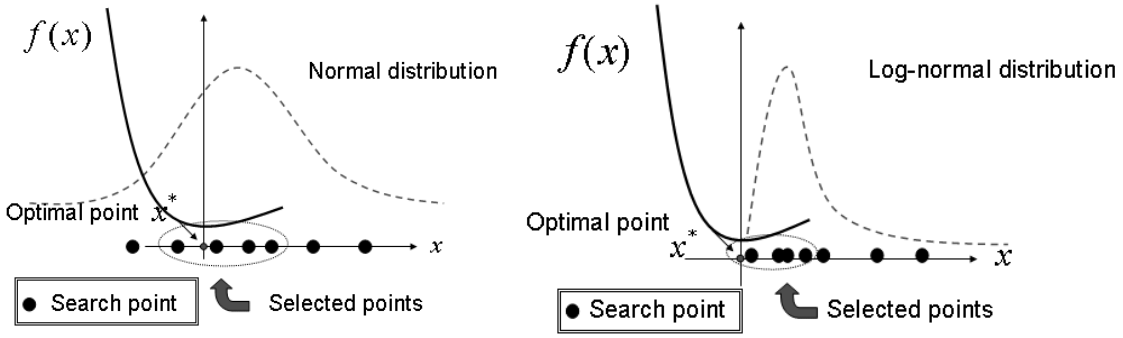

Figure 4: Generating search points by normal distribution and lognormal distribution

It is difficult to update the mean μ_L and the covariance matrix C_L of lognormal distribution from the set of search points. We apply variable transformation to search points and use the update formula of mean value (2) and covariance matrix (4) in normal distribution. For example, we describe variable transformation in the case $n = 2$ (Figure 5). If we set $z^j = (\log(x_1^{(g),j}), \log(x_2^{(g),j}))$, then z^j is normally distributed. Adversely, we assume that z^j is generated by normal distribution and $x^{(g),j} = (e^{z_1}, e^{z_2})$ is given by z^j. Then, it can be considered that $x^{(g),j}$ obtained by z^j is generated by lognormal distribution. Finally we update the mean value of normal distribution and covariance matrix of z^j by using (2) and (4).

By the using the above variable transformation, we can describe the algorithm with lognormal distribution. We set $x_i = e^{z_i}$ and use the function $\hat{P}(z;\rho)$ defined by

$$\hat{P}(z;\rho) = P_1(e^{z_1}, \ldots, e^{z_n}; \rho). \tag{11}$$

Then, problem P1(ρ) can reformulated as the following unconstrained minimization problem

$$\begin{aligned} \mathrm{PLN}(\rho) \qquad & \min \quad \hat{P}(z;\rho) \\ & \text{s.t.} \quad z \in R^n. \end{aligned} \tag{12}$$

The proposed method is to find an optimal solution of PLN(ρ) sequentially as $\rho \to \infty$.

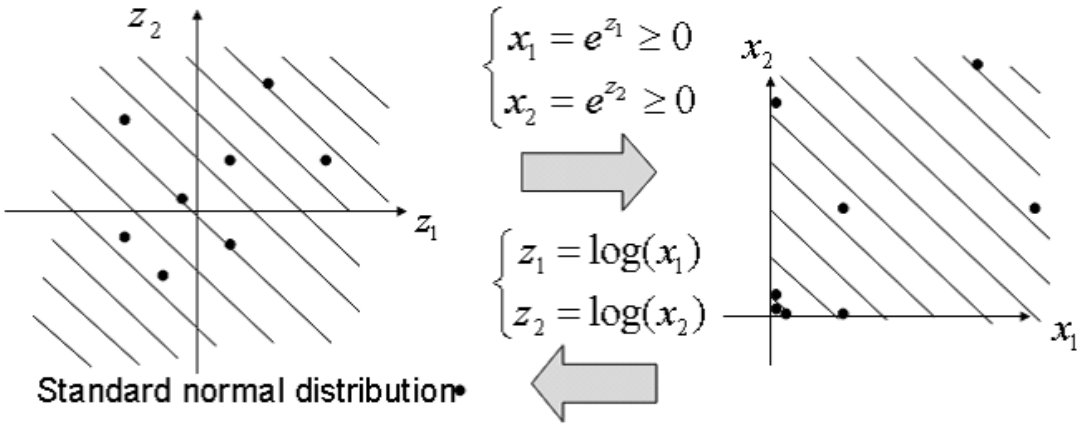

Figure 5: Generating points of lognormal distribution by variable transformation

The method based on lognormal distribution

Step 0. Select penalty parameter $\rho_0 \in (0, \infty)$. Set mean $m^{(0)} \in \Re^n$, covariance matrix $C^{(0)} \in \Re^{n \times n}$ and stepsize $\sigma^{(0)} \in \Re$.
Step 1. Find an (approximate) optimal solution $\bar{x}^{(k)}$ of subproblem PLN(ρ^k) by applying the CMA-ES to with $m^{(k)}$ and $C^{(k)}$.
Step 2. If termization criterion is satisfied by $x^{(k)} = \exp(\bar{z}^{(k)})$, then stop.
Step 3. Select $\rho^{(k+1)} \in (\rho^{(k)}, \infty)$. Update $m^{(k+1)} = \bar{z}^{(k)}$, $C^{(k+1)}$, set $k = k + 1$, and go to Step 1.

Similarly, in the algorithm of normal distribution method, we use $C^{(g)}$ as $C^{(k+1)}$ in Step 3. Here, $C^{(g)}$ is generated from the last iteration of Step 1 in the CMA-ES.

3.3 Projection method

In the CMA-ES we generate search points by the normal distribution. Here, the generated search point is projected onto the feasible region (Figure 6). Therefore, they are always in nonnegative orthant as lognormal distribution method.

By using projection, generated search point are always in nonnegative region. Therefore, search points always generated in positive region as lognormal distribution method.

We call the CMA-ES in which Step 3 is exchanged for the following Step 3' as CMA-ES with projection.

CMA-ES with projection
Step 3' New population sampling with projection

$$\hat{x}_k^{(g+1)} \sim N\left(m^{(g)}, (\sigma^{(g)})^2 C^{(g)}\right) \quad for\ k = 1, \cdots, \lambda$$

$$x_i^{(g+1)} = \begin{cases} \max\{0, \hat{x}_i{}^{(g+1)}\} & i \in S \\ x_i^{(g+1)} & \text{otherwise} \end{cases}$$

Note that the projection onto nonnegative region takes little calculation time.

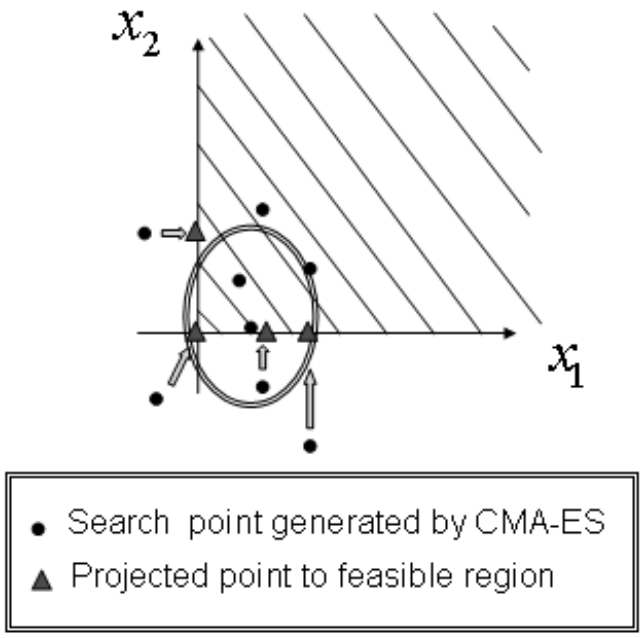

Figure 6: Selecting search points by CMA-ES by projection

In the followings we describe the method using CMA-ES with projection.

Projection method

Step 0. Select penalty parameter $\rho_0 \in (0, \infty)$. Set mean $m^{(0)} \in \Re^n$, covariance matrix $C^{(0)} \in \Re^{n \times n}$ and stepsize $\sigma^{(0)} \in \Re$.

Step 1. Find an (approximate) optimal solution $\bar{x}^{(k)}$ of by applying the CMA-ES to subproblem with $m^{(k)}$, $C^{(k)}$.

Step 2. If the termination criterion is satisfied by $x^{(k)} = \exp(\bar{z}^{(k)})$, then stop.

Step 3. Select $\rho^{(k+1)} \in (\rho^{(k)}, \infty)$. Update $m^{(k+1)} = \bar{z}^{(k)}$ and $C^{(k+1)}$, set $k = k+1$, and go to Step 1.

Similarly, in previous normal distribution method, we use $C^{(g)}$ as $C^{(k+1)}$ in Step 3. Here, $C^{(g)}$ is obtained by the last iteration of Step 1 in the CMA-ES.

In CMA-ES with projection, we consider the projection only onto the nonnegative constraints since it is easy to calculate. If the objective function or equality constraints are complicated, projection method may show poor performance. On the contrary, if it is more expensive to calculate the value of objective function than the projection onto more complicated constraints, we can expect that the total calculating time decreases. As a result, how to project is important for the performance of this method. For instance, we can consider the projection not only onto nonnegative constraints but also onto the feasible set involving equality constraints. We can also consider the projection measured by the norm other than Euclidean norm. If we consider the projection with the norm $\| x \| C^{(g)}$ defined by $x^T C^{(g)} x$, then the projected points will be influenced by projected points will be influenced by the shape of function.

4 Numerical Experiments

In this section we compare three proposed algorithms by means of numerical experiments. Test problems are chosen from CUTEr, GAMS, and [5]. Each method is coded by MATLAB 6.5 and run on a machine with 2.80GHz CPU and 1G memory.

We set initial parameters as $C^{(0)} = I$ and $\sigma^{(0)} = 0.5$. If the initial mean value $m^{(0)}$ are given by the test problem, we use it. If the initial mean value $m^{(0)}$ are not given by the test problem, we choose $m^{(0)}$ from the constant distribution $(0, 1)$.

In Step 3 of each method we use $C^{(g)}$ as $C^{(k+1)}$, where $C^{(g)}$ is obtained by the last iteration of Step 1 in the CMA-ES. Penalty parameter is updated as

$$\rho^{(k+1)} = 10\rho^{(k)}.$$

In the projection method, we calculate the projection onto the nonnegative orthant

$$x_j^{(g),i} := \max\{0, x_j^{(g),i}\}, \quad j \in S, i = 1, \ldots, \lambda.$$

The termination criterion of the main loop and the inner loop of the CMA-ES are given as follows. We use the following termination criterion for the inner loop of the CMA-ES

$$\left| P^k(x^{(g),1}) - \min_{j=1,\ldots,k-1} P(x_i^{(g),1}) \right| < 10^{-5} \tag{13}$$

where $P^k(x)$ is replaced by $P_1(x; \rho^k)$ in (8). That is, we consider the differential between the extended function value at $x^{(k)}$ and that at the best point which was obtained by the previous iteration. If this differential becomes nearly 0, we consider that the method has converged. If the minimum of the extended function meets constraints, then it must solve the original problem. Therefore, stop condition of main loop is set as follows:

$$\sum_{i \in S} \max(0, -x_i^{(k)}) + \sum_{j=1}^{m} |h(x_j^{(k)})| < 10^{-8} \tag{14}$$

The test problems and their properties are shown in Table 2. Here, ackley is originally unconstrained problem, but we added constraints $x_i \geq 0$ $(i = 1, \ldots, 20)$ in this experiments. The values of n, m, and l in Table 2 represents the rank of x, the number of inequality constraints, the number of equality contraints, respectively, and Act is the number of inequality constraints which are active on the global minimum.

The test problems are classified as A, B, C.

Group A All inequality constraints are inactive on the global minimum.

Group B There are some active inequality constraints on the global minimum, but not all inequality constraints are active.

Group C All inequality constraints are active on global minimum.

Each problem was solved ten times and their results are shown in Tables 3, 4, and 5.

In these tables, "#S" represents the number of trials in which we obtain the global optima and "#F" represents average number of function evaluations, when the methods stop. Here let

Table 2: Problem groups

Problem group	Function name	(n, m, l)	Act
A	hatflda	(4,0,4)	0
	hs001	(2,1,0)	0
	supersim	(2,1,2)	0
	logros	(2,2,0)	0
	tame	(2,2,1)	0
	try-b	(2,2,1)	0
B	extrasim	(2,1,1)	1
	bt13	(5,1,1)	1
	harker	(25,6,7)	8
	lotschd	(12,12,7)	5
C	ackley	(20,20,0)	20
	griewank	(10,10,0)	10

Table 3: Results for problems in Group A

	Normal				Lognormal				Projection			
Problem	#S	Best	Worst	#F	#S	Best	Worst	#F	#S	Best	Worst	#F
hatflda	10	1.29e-8	3.35e-6	668	9	3.74e-9	1.87e-1	867.2	10	7.99e-9	1.89e-4	684.8
hs001	10	2.48e-9	7.82e-8	795.0	10	3.52e-9	6.96e-8	796.5	10	9.54e-10	4.35e-8	805.0
supersim	10	6.67e-1	6.67e-1	592.8	10	6.67e-1	6.67e-1	598.8	10	6.67e-1	6.67e-1	582
logros	10	2.24e-9	2.04e-6	940.8	0	6.93e-1	6.93e-1	47.5	2	1.07e-8	6.93e-1	527.4
tame	10	2.93e-9	6.05e-4	884.4	5	2.45e-55	1.09e-2	4359.6	10	2.83e-9	7.16e-5	939.6
try-b	0	5.21e-2	8.53e-1	4002.0	0	4.66e-67	1.0	1339.2	0	5.83e-2	1.0	3415.8

$\hat{x}$ be the optimal solution found by the algorithm and x^* be the global optimum. Then, if the inequality

$$\frac{f(\hat{x}) - f(x^*)}{\max\{1, f(x^*)\}} \leq 0.01 \tag{15}$$

is satisfied, we consider that $\hat{x}$ converges to x^*. "Best" shows the minimum value of the objective function among ten trials and "Worst" shows the maximum value.

From Table 3, lognormal distribution shows a little poorer performance. On the contrary normal distribution method and projection method show better performance on most of problems in Group A. We can interpret this result as follows. Projection method is identical with normal distribution method near optimal solution, since the inequality constraints are not active on the optimal solution. In the result of the lognormal distribution method, the global minima are obtained only a few times for logros problem, try-b problem and tame problem. Actually, these problems have local minimum near the global minima. For logros and try-b, number of function evaluation is fewer than the other methods to find (local) minimum. This is because most of inequality constraints are active on local minimum on these problems.

From Table 4 we obtain the following result on Group B. In most of problems, the normal distribution method is superior to the lognormal distribution method and projection method. In extrasim problem, the lognormal distribution method is inferior to other two methods, because

Table 4: Results for problems in Group B

	Normal				Lognormal				Projection			
Problem	#S	Best	Worst	#F	#S	Best	Worst	#F	#S	Best	Worst	
extrasim	10	1	1	1578.6	1	1	1.66	3745.8	10	1	1	15
bt13	8	2.68e-3	1.78e-2	10093.6	0	1.37e-2	6.73	25000.0	4	0	3.97	18
harker	1	-9.86e+2	-9.77e+2	64179.6	0	-9.85e+2	-1.03e-14	90595.8	0	-9.83e+2	-8.58e+2	93
lotschd	9	2.23e+3	2.34e+3	16613.3	5	2.23e+3	2.80e+3	95679.5	9	2.23e+3	2.76e+3	28

Table 5: Results for problems in Group C

	Normal				Lognormal				Projection			
Problem	#S	Best	Worst	#F	#S	Best	Worst	#F	#S	Best	Worst	#F
ackley	3	7.55e-4	2.17	6567.6	10	3.45e-5	1.31e-4	524.4	10	8.45e-8	1.42	2304
griewank	10	1.86e-7	1.86e-5	2310.0	10	1.61e-12	3.25e-7	264	10	7.37e-9	3.45e-5	630

this method tends to converge to local minima. In bt13, harker, and lotschd, projection function need more a function evaluation than normal distribution method. Since these problems have the complicated equalities, if the search points approximately satisfy the equality constraints, then the projected points often violate the equality constraints. As a result, the projected search points are seldom selected and evaluations of these points become futile. From this reason, projection method (projected to nonnegative constraints) is not suitable method to the problems whose equality constraints are complicated.

Finally we discuss the result of Group C (Table 5). The lognormal distribution method and projection method are superior to normal distribution method. Particularly in ackley function the lognormal distribution method and projection method always find a global minimum in spite of existance of several local minima. Judging from this fact, we can consider those two methods tend to find a solutions where many inequality constraints become active. Therefore, if the problem's optimal solution is on the boundary of the feasible set (e.g. the concave objective function), it is expected that lognormal distribution method and projection method show great performance to these problems.

From the above numerical results, we give the flowchart which indicates the choice of three methods (Figure 7).

5 Concluding remarks

For solving constrained optimization problems, we have proposed three CMA-ES based methods, normal distribution method, lognormal distribution method, and projection method. Moreover by applying these methods to several test problems, we have revealed the numerical properties of each method.

As future issues, it would be interesting to consider update rules for stepsize or covariance matrices corresponding to each proposed method. Particularly, the numerical results obtained by lognormal distribution methods were not so good as we have expected. This may be due to the fact that, in updating the covariance matrices, we have employed the variable transformation $z_i = \log(x_i)$ to reduce the computational cost, and have not used genuine covariant matrices

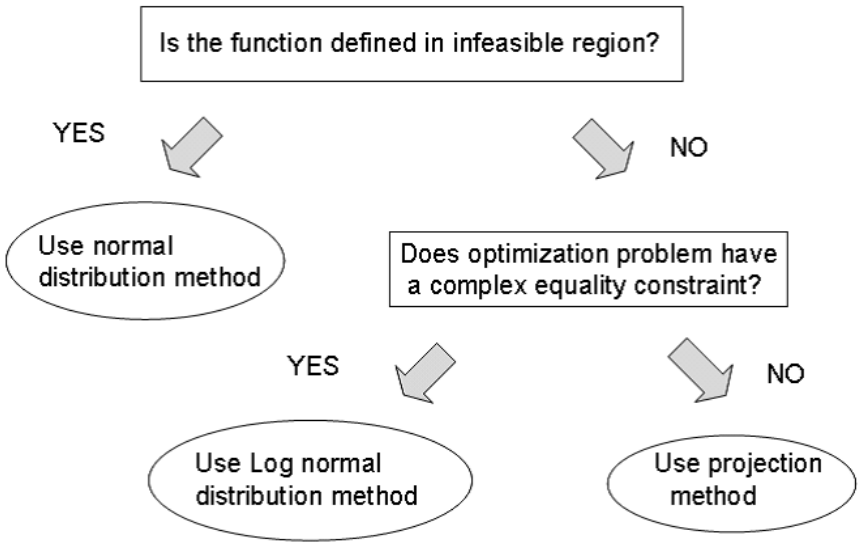

Figure 7: Flowchart for the choice of three methods.

inherent in lognormal distribution. For the projection method, we considered only the projection onto the nonnegative orthant. However the projection has a room for improvement. For example it is also possible to consider the projection onto the constraint region itself, or to define it by using a norm measured by covariance matrices. If such a projection is introduced, then the number of function evaluation may decrease, while the computational cost for the projection will be more expensive.

Acknowledgment

First of all, I would like to express my sincere thanks and appreciation to Associate Professor Nobuo Yamashita for his continual guidance, extreme patience, and thoughtful assist. I would also like to express Professor Masao Fukushima for his precise and helpful comments. I also greatly appreciate to Assistant Professor Shunsuke Hayashi's helpful assist, his humor and his lectures of ramen. Finally, I would like to thank all the members of in Fukushima's Laboratory.

References

[1] H. G. Beyer and D. V. Arnold: *Qualms regarding the optimality of cumulative path length control in CSA/CMA-evolution strategies.* Evolutionary Computation, Vol. 11(1), 2003, pp. 19–28.

[2] A. R. Conn, K. Scheinberg and Ph. L. Toint: *On the convergence of derivative-free methods for unconstrained optimization.* Invited presentation at the Powellfest, Cambridge, July, 1996.

[3] N. Hansen: *The CMA evolution strategy: a tutorial.* 2005, http://www.bionik.tu-berlin.de/user/ niko/cmatutorial.pdf

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

[5] A. Hedar: *Studies on metaheuristics for continuous global optimization problems.* Ph. D. thesis, Kyoto University, 2004

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

[7] F. Herrera, M. Lozano and J. L. Verdegay: *Tackling real-coded genetic algorithms: operators and tools for behavioural analysis.* Artificial Intelligence Review, Vol. 12, 1998, pp. 265–319.

[8] C. T. Kelley: *Detection and remediation of stagnation in the Nelder-mead algorithm using a sufficient decrease condition.* SIAM Journal on Optimization, Vol. 10, 1999, pp. 43–55.

[9] E. Mezura-Montes and C. A. Coello Coello: *On the usefulness of the evolution strategies' self-adaptation mechanism to handle constraints in global optimization.* Technical Report EVOCINV-01-2003, Evolutionary Computation Group at CINVESTAV, January 2003

[10] A. Ostermeier and N. Hansen: *An evolution strategy with coordinate system invariant adaptation of arbitrary normal mutation distributions within the concept of mutative strategy parameter control.* GECCO-99 Proceedings of the Genetic and Evolutionary Computation Conference, 1999, pp. 902–909.

[11] C. Spieth, R. Worzischek and F. Streichert: *Comparing evolutionary algorithms on the problem of network inference.* GECCO 2006 Proceedings of the Genetic and Evolutionary Computation Conference, 2006, pp. 305–306

[12] E. G. Talbi: *A taxonomy of hybrid metaheuristics.* Joernal of Heuristics, Vol. 8, 2002, pp. 541–564