Yokohama Publishers
ISSN 1880-5221 ONLINE JOURNAL

Pacific Journal of Optimization
Vol. 2, No. 2, pp. 261–276, May 2006

A NEW CHAOS GENERATOR BASED ON THE AFFINE SCALING METHOD FOR GLOBAL OPTIMIZATION PROBLEMS*

Keiji Tatsumi, Yousuke Yamamoto and Tetsuzo Tanino

Abstract: In this paper, we focus on chaotic metaheuristic methods which solve continuous and discrete global optimization problems having many local minima. Those methods exploit the sensitive dependence on initial conditions of the chaotic dynamics and search for a solution extensively in the feasible region. Then, the performance of the chaotic generator used in those methods is very important to find a desirable solution. However, the conventional chaos generator based on the steepest descent method has a drawback, though being used widely. A sequence generated by the method tends to accumulate to the boundary of the feasible region. Thus, in some cases, it is difficult to obtain a satisfactory solution by the existing methods. Therefore, in this paper, we propose a new chaos generator based on the affine scaling method which can overcome the drawback. We apply the chaotic metaheuristic method with the proposed chaos generator to the minimization problem of a concave function and the quadratic assignment problem, and verify the efficiency of the proposed method through some numerical experiments.

Key words: *chaos dynamics, global optimization, quadratic assignment problem, multi-start method, meta-heuristics*

Mathematics Subject Classification: *65K05, 65P20, 90C59*

---

1 Introduction

Recently, chaotic metaheuristic methods have been investigated to solve continuous or discrete global optimization problems that have many local minima. These methods exploit the sensitive dependence on initial conditions in chaotic trajectories to search for a solution extensively in the feasible region. Although the method was originally proposed as one of neural networks called the chaotic neural network [3, 9, 15], which can be applied to a certain kind of combinatorial optimization problem such as the traveling salesman problem, the model has been extended for general optimization problems [5, 8, 10, 13].

In those methods, to generate a chaotic sequence, an original constraint problem is transformed into an unconstrained problem by using a diffeomorphic mapping from the Euclid space to the feasible region, and then the steepest descent dynamics with a sufficiently large step-size is applied to the problem. Then, it is well-known that a generated sequence by them is chaotic [4]. Those methods search for a solution not only in the steepest descent direction of the objective function but also broadly in the feasible region by making use of the chaotic behavior. Hence, those methods have the ability to find the global optimum or a desirable local minimum for the global optimization problem. However, a generated sequence tends to accumulate on the boundary of the feasible region in spite of its chaotic behavior. Thus, in some cases, it is difficult to obtain the desirable solution by the existing methods. Even for the problem having local minima on the boundary of the feasible region, those methods sometimes can not obtain a high-quality solution.

Therefore, we propose a new chaos generator based on the affine scaling method. Since the proposed method requires neither a large step-size nor a diffeomorphic mapping to generate a chaotic sequence, it can be expected to overcome the drawback. In particular, since the proposed method is suitable to especially the global optimization problem having local minima on the boundary of the feasible region, we apply it to problems of this kind and verified the performance of the proposed chaos generator. In addition, we improve the proposed method by using a switched step-size technique. Furthermore, as a chaotic metaheuristic method which uses the proposed chaos generator, we select the chaotic multi-start method (CMS), which is proposed as a metaheuristic method for the continuous global optimization problem [10]. First, we applied the CMS with the proposed chaos generator to the continuous minimization problem of a concave function. Secondly, we focus on the quadratic assignment problem (QAP), which is known to be an NP hard combinatorial problem, and apply the CMS with the proposed generator to the problem, where we extend the CMS for the combinatorial problem. For both problems, through some numerical experiments, we verify the efficiency of the CMS with the proposed chaos generator.

This paper consists of six sections. In section 2, we introduce the chaotic metaheuristic method and the conventional chaos generator based on the steepest descent method. Next, in section 3, we propose a new chaotic generator based on the affine scaling method. In section 4, we apply CMS with the proposed method to the continuous minimization problem of a concave function. In section 5, we extend CMS for QAP and apply the method with the proposed generator to QAP. Finally, we conclude in section 6.

2 Chaotic Metaheuristic Method

2.1 Steepest Descent Model

In this subsection, we introduce the chaotic metaheuristic method based on the steepest descent method, which has been exploited to solve the global optimization problem. At first, we consider the following constraint optimization problem having many local minima:

$$\text{(P1)} \quad \begin{array}{ll} \text{min.} & f(x) \\ \text{s.t.} & x \in X, \end{array}$$

where $X \subset \Re^n$ is a nonempty closed convex set and $f(x)$ is a multi-peaked and differentiable function. Now, we define $E(z) = f(s(z))$ by using an appropriate diffeomorphic mapping $s : \Re^n \to X$, and transform (P1) into the unconstrained problem:

$$\text{(P2)} \quad \text{min.} \quad E(z).$$

To solve (P1), we consider the following steepest descent dynamics (SD):

$$z(t+1) = z(t) - \alpha \nabla E(z(t)),$$

where α denotes a step-size. If we choose a small α, this dynamics means the steepest descent method and a sequence generated by SD converges to a local minimum. On the other hand, when α is sufficiently large, it is well known that this dynamics generates a chaotic sequence in the sense of Li-York [4]. Those methods search for a solution not only in the steepest descent direction of the objective function but also broadly in the feasible region by making use of the chaotic behavior. Hence, it is asserted that those methods have the ability to find the global optimum or a desirable local minimum for the global optimization problem.

Although this model is often used in many chaotic metaheuristic methods [3, 7, 9, 10, 13, 15], a sequence $\{x(t)\}$ generated by this dynamics often overconcentrates around the boundary of the feasible region even if the corresponding sequence $\{z(t)\}$ is widely scattered in $\Re^n$. Independently of the shape of the objective function, the tendency is caused by the diffeomorphic mapping and a large step-size, which are required to generate a chaotic sequence. For example, consider the following problem:

$$\begin{array}{ll} \text{min.} & (x+4.2)(x+2.2)(x+1.2)(x-1.7)(x-2.8)(x-3.865) \\ \text{s.t.} & -4.3 \leq x \leq 4.3. \end{array}$$

We apply the SD model to this problem by using the hyperbolic tangent $s(u) = 4.3(1 - \exp(-u)) \ / \ (1 + \exp(-u))$ as a diffeomorphic mapping. Figure 1 shows points x generated from some initial points by SD when step-size α is varied. It shows that the generated

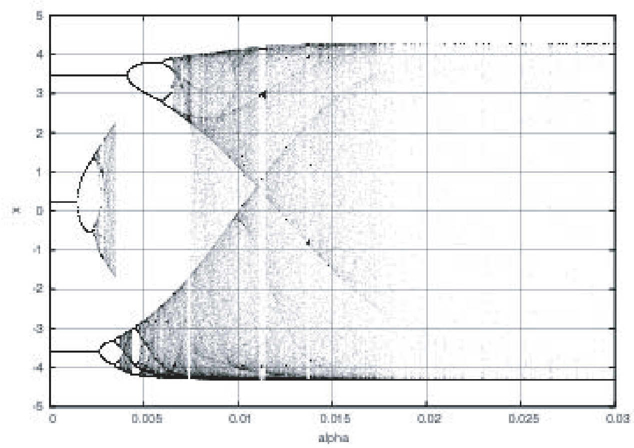

Figure 1: Steepest Descent model: SD

sequence tends to overconcentrate around the boundary of the feasible region. Note that when α is too large, almost all points in the sequence accumulate to the point $x = 4.3$ or $x = -4.3$. Hence, for some problems, chaotic metaheuristic methods with SD fail to find a desirable solution. Even for the problem having many local minima on the boundary of the feasible region, those methods sometimes can not obtain a satisfactory solution.

Therefore, we propose a new chaos generator based on the affine scaling method (AS) which can be expected to overcome the above drawback. In this paper, we focus on the chaotic multi-start method (CMS) for the continuous global optimization problem, which was reported to have better performance than conventional chaotic metaheuristic methods such as the chaotic annealing method [10]. In the next subsection, we summarize the idea of CMS.

2.2 Chaotic Multi-start Method

The CMS is a multi-start method where local searches (LS) are executed from points found by a chaotic global search (GS). In this method, the chaotic GS procedure is executed until the algorithm terminates. If the GS procedure finds a promising area which can be expected to include a good local minimum, then a LS procedure is started from it. If the obtained solution by LS is better than the tentative one, then it is updated by the obtained solution. Figure 2 shows the concept of CMS. The starting conditions of LS procedure should be selected appropriately for each problem. Although there are two ways of implementing CMS: parallel and sequential implementations, in this paper we use the latter, where if once a LS procedure starts, the GS procedure is paused for the completion of the LS.

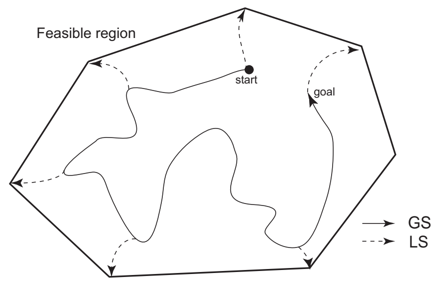

Figure 2: Chaotic multi-start method: CMS

3 Affine Scaling Method

In this section, we introduce the affine scaling method (AS) in brief and show how it can be exploited to generate a chaotic sequence. The AS is one of the simplest interior point methods which was originally proposed to solve the linear programming problem. It can be also applied to the nonlinear programming problem by linearization of its objective function at a current point as follows. Now, we consider the following problem:

$$\begin{array}{lll} \text{(P3)} & \text{min.} & h(x) \\ & \text{s.t.} & Ax = b, \quad x \geq 0, \end{array}$$

where $h(x)$ is differentiable and $A \in \Re^{m \times n}$, $x \in \Re^n$, $b \in \Re^m$. Then x is modified by the following dynamics:

$$x(t+1) = x(t) - \beta \frac{X(t)\Pi^{\perp}_{AX(t)} X(t) \nabla h(x(t))}{\|\Pi^{\perp}_{AX(t)} X(t) \nabla h(x(t))\|}, \tag{1}$$

where $X(t) := \text{diag}\{x_1(t), \ldots, x_n(t)\}$, $\Pi^{\perp}_{AX(t)}$ denotes the projection matrix onto Ker $AX(t)$ and β is a step-size. Although this method is sometimes called the short-step affine scaling method, in this paper, we call it AS. If $\beta = 1$, x is always obtained within the feasible region. On the other hand, the following update can be available:

$$x(t+1) = x(t) - \beta \frac{X(t)\Pi^{\perp}_{AX(t)} X(t) \nabla h(x(t)}{\|(\Pi^{\perp}_{AX(t)} X(t) \nabla h(x(t)))\|_{\infty}}, \tag{2}$$

where $\|x\|_\infty := \max_i |x_i|$. This method is called the long-step affine scaling method (LAS). If β is sufficiently close to 1, this method can find a solution within a smaller number of iterations than AS [14].

A solution or a sequence obtained by (2) mainly depends on the objective function $h(x)$. If $h(x)$ is a linear function, this method can find the optimal solution [14]. If $h(x)$ is a convex quadratic function, it can also find the optimal solution by adding some procedures such as the trust region method [1]. However, if eigenvalues of Hessian matrix of $h(x)$ are sufficiently large, the sequence generated by AS without any additional procedures is chaotic, where the optimum solution is a snap-back repeller. Furthermore, when $h(x)$ is a concave function or the problem (P3) has many local minima on the boundary of the feasible region, by adding an appropriate barrier function such that the eigenvalues of Hessian matrix $\nabla h(x)$ at some local minima are sufficient large, a generated sequence can be expected to be chaotic around some local minima. In the case that $h(x)$ is a multi-peaked function, we can obtain a chaotic sequence around some local minima, with multiplying the objective function by an appropriate positive constant.

In this paper, we propose a new chaos generator by making use of this property of AS. Since this method requires neither a large step-size nor a diffeomorphic mapping, it can be expected to overcome the drawback of SD model and generates a chaotic sequence which reflects the shape of the objective function. In particular, since for the problem having many local minima on the boundary of the feasible region, this model can be considered to be suitable, in this paper, we focus on the proposed generator for problems of this kind. Hence, in section 4, we apply CMS with the proposed generator to the minimization problem of a concave function which has bounded constraints. Next, in section 5, we consider the quadratic assignment problem, which is an NP-hard combinatorial problem. In order to solve the QAP, we apply CMS with the proposed generator to the relaxed problem of the original QAP, which has all local minima at vertices of the relaxed feasible region.

4 Minimization Problem of the Continuous Concave Function

Many conventional chaotic metaheuristic methods with SD generator have been applied to the global optimization problem with a rectangular constraint to verify their performance [7, 13, 15], because it is easy to select the diffeomorphic mapping, that is, the sigmoid function or the hyperbolic tangent. Thus, in this section, to compare the proposed generator with a conventional one, we consider the following minimization problem of a concave function with a rectangular constraint:

$$\begin{array}{lll} \text{(CP)} & \text{min.} & g(x) = \dfrac{1}{2}x^\top W x + b^\top x \\ & \text{s.t.} & 0 \le x_i \le 1,\ i = 1, 2, ..., n, \end{array}$$

where $W \in \Re^{n\times n},\ x,\ b \in \Re^n$ and $g(x)$ is concave. In general, this problem has many local minima at vertices in the feasible region.

4.1 Chaos Generator Based on the AS

Now, the following interior penalty function:

$$p(x) = c_p \ \times \sum_{i=1}^{n} \left(\frac{1}{\cos^2\left(\pi\left(x_i - 0.5\right)\right)} - 1 \right),$$

is added to the objective function $g(x)$, where c_p is a positive constant. Then, we have

$$\begin{array}{ll} \text{min.} & g_p(x) := g(x) + p(x) \\ \text{s.t.} & 0 \le x_i \le 1,\ i = 1, 2, ..., n, \end{array}$$

and $x(t)$ is updated by

$$x(t+1) = x(t) - \beta \frac{\Pi(x(t)) \nabla g_p(x(t))}{S(x(t)))}, \tag{3}$$

where

$$\begin{aligned} \rho(u) &:= \frac{u^2(1-u)^2}{u^2 + (1-u)^2}, \\ \Pi(x) &:= \operatorname{diag}\{\rho(x_1), \dots, \rho(x_n)\}, \\ S(x) &:= \sqrt{\sum_{i=1}^{n} \left(\frac{1}{x_i^2} + \frac{1}{(1-x_i)^2} \right) (\rho(x_i))^2 (d_i(x))^2 + \delta}, \end{aligned}$$

and δ denotes a sufficiently small constant which prevents $S(x)$ from being zero. In this model, if c_p is selected appropriately, AS can be expected to generate a chaotic trajectory around the neighborhoods of local minima.

For example, consider the problem of which objective function is $g(x) := -5x(x - 1.2)$ subject to $0 \le x \le 1$. We apply AS to the problem with an appropriate c_p. Figure 3 shows the points generated by AS when step-size β is varied from 0 to 1. This figure shows the

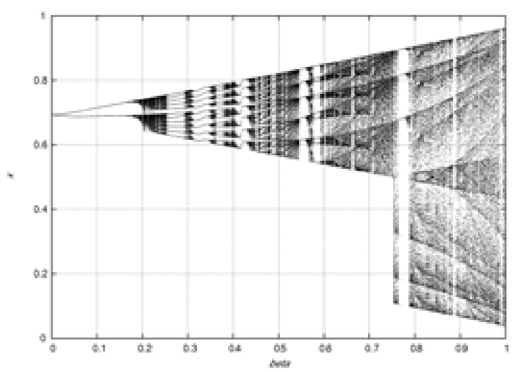

Figure 3: Trajectory by AS model

chaotic behavior of x. When β is sufficiently close to 1, the generated sequence are widely scattered in the feasible region. Moreover, we can see that the sequence does not accumulate to points close to the boundary, $x = 1$ and $x = 0$.

However, if eigenvalues of Hessian matrix of objective function $g_p(x)$ are too large around a local minimum, it is sometimes observed that a generated sequence is trapped in a basin which includes an undesirable minimum in spite of its chaotic behavior. Therefore, in the next subsection, we improve AS model.

4.2 Switched Step-size AS

In this subsection, we improve AS model by switching two step-size $\beta_L = 1$ and $\beta_H > 1$, provided that the next point is found in the feasible region. At first, we update a current point $x(t)$ by (3) with the large step-size β_H and obtain the candidate point $x^H(t+1)$. If the point satisfies all the constraint, the next point is determined by $x(t+1) := x^H(t+1)$, otherwise updated by (3) with β_L. We call this method the switched step-size AS method (SAS).

Moreover, we relaxes the condition under which the large step-size β_H is available. All the elements of $x(t)$ are updated independently as follows. At first, all the elements $x_i(t)$ whose candidate $x_i^H(t+1)$ satisfies the constraint $0 \leq x_i^H(t+1) \leq 1$ are updated by $x_i^H(t+1)$, where each candidate $x_i^H(t+1)$ is obtained by

$$x_i(t+1) = x_i(t) - \beta \, \frac{\rho(x_i(t))}{S(x(t))} \, \frac{\partial g_p(x(t))}{\partial x_i}, \tag{4}$$

with β_H. Then, the other elements $x_i(t)$ are updated by (4) with β_L. We call this method the relaxed switched AS method (RSAS). In addition, we call RSAS with the long-step AS method, RSLAS. These chaos generators can be easily combined with the CMS mentioned in the previous section.

4.3 Numerical Experiment

In this subsection, we report the results of numerical experiments, where we compared CMS's with the proposed and conventional generators. All computer programs were coded in C and executed on a PC (CPU: AthronXP 1800+, memory: 1.5G). For all problems, we conducted preparatory experiments to find suitable values of parameters in each method.

At first, through some numerical experiments, we observed that sequences generated by AS generator were chaotic by measuring their approximate maximal Lyapunov exponents [11]. It is well known that if the maximal Lyapunov exponent of the sequence is positive, the sequence can be considered to be chaotic. Next, we randomly generated some minimization problem of the concave function represented by (CP) and applied CMS's with SD, SAS, RSAS and RSLAS generators to those problems, where a LS procedure started from every point found by the chaotic GS procedure in order to verify the efficiency of each method. As a LS procedure, we used SD method with a sufficiently small step-size for all CMS's. The step-size α was 0.1 in SD, β was $10\sqrt{n}$ in SAS, and β_L and β_H were $10\sqrt{n}$ and $100\sqrt{n}$ in RSAS and RSLAS, respectively, where n denotes the dimension of the decision space. Each method was executed from five different initial points.

Tables 1 and 2 show the results of four methods for 50 and 150 dimensional problems, which show the worst, best, average and standard derivation of objective function values, and the max and average norm of $\Delta x(t) := x(t+1) - x(t)$. Note that S.D. means the standard derivation. It can be seen that RSAS and RSLAS are superior to SD in terms of the searching ability. On the other hand, although $\|\Delta x\|$ in SD is relatively large, it cannot find desirable solutions. This is mainly because points generated by SD tend to accumulate to the boundary of the feasible region. Form these results, it is concluded that RSAS and RSLAS are effective as a chaos generator to solve these problems.

Table 1: Concave problem: dimension 50

(a)The best, worst and average of objective values

	SAS	RSAS	RSLAS	SD
f_{worst}	-55.817	-86.107	-98.118	-81.124
f_{best}	-120.02	-267.691	-270.477	-181.753
f_{avg}	-99.872	-195.178	-203.139	-133.803
S.D.	14.457	35.824	34.057	17.617

(b)The max and average of $\|\Delta x\|$

	SAS	RSAS	RSLAS	SD
norm_{max}	0.9545	2.2277	2.18764	5.80284
norm_{avg}	0.15904	0.6721	0.62376	0.29396

Table 2: Concave problem: dimension 150

(a)The best, worst and average of objective values

	SAS	RSAS	RSLAS	SD
f_{worst}	-196.544	-240.775	-197.483	216.565
f_{best}	-280.823	-684.817	-641.511	-266.672
f_{avg}	-245.159	-506.913	-455.276	-37.504
S.D.	24.702	84.944	88.914	89.35

(b) The max and average of $\|\Delta x\|$

	SAS	RSAS	RSLAS	SD
norm_{max}	0.98144	3.42124	3.55252	10.31688
norm_{avg}	0.13964	0.98012	1.20232	0.55168

5 Quadratic Assignment Problem

In this section, we extend CMS for the quadratic assignment problem (QAP) and apply the extended method with AS and SD chaos generators to the problem.

5.1 Formulation of QAP

The QAP is formulated as follows [6]:

$$
\begin{aligned}
\text{(QAP)}\ \ \text{min.}\quad & \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\sum_{l=1}^{n} f_{ij} d_{kl} x_{ik} x_{jl} \\
\text{s.t.}\quad & \sum_{i=1}^{n} x_{ik} = 1,\ k \in \mathcal{N}, && (5) \\
& \sum_{l=1}^{n} x_{jl} = 1,\ j \in \mathcal{N}, && (6) \\
& x \in \{0,1\}^{\mathcal{N}\times\mathcal{N}},
\end{aligned}
$$

where $\mathcal{N} = \{1, \ldots, n\}$ is a set of n facilities or n locations, d_{kl} is the distance between locations k and $l \in \mathcal{N}$, and f_{ij} is the flow between facilities i and $j \in \mathcal{N}$. Each facility should be assigned to exactly one location. If decision variable x_{ij} is 1, then facility i is assigned to the j. The aim is the minimization of the whole cost given by the quadratic objective function of (QAP).

This problem is the discrete optimization problem, while the CMS is a solver for the continuous optimization problem. Hence, we extend the method by relaxing the original problem, where the 0-1 constraint $x \in \{0,1\}^{\mathcal{N}\times\mathcal{N}}$ is replaced with the hypercube constraint $x \in [0,1]^{\mathcal{N}\times\mathcal{N}}$. Then, we use two different relaxed problems for LS and chaotic GS procedures. Furthermore, in order to compare the LAS generator with the conventional SD generator, we consider the following three methods:

Chaotic Multi-Start method – SD generator (CMS-SD)

Global Search: SD and Local Search: LAS

Chaotic Multi-Start method – LAS generator (CMS-LAS)

Global Search: LAS and Local Search: LAS

Chaotic Multi-Start method – Switched step-size LAS generator (CMS-SLAS)

Global Search: SLAS and Local Search: LAS

Thus, we deal with three relaxed problem: a problem for the LS procedure with LAS method and two problems for GS procedures with LAS and SD chaos generators, respectively.

5.2 LS Procedure with LAS Method

A LS procedure is a descent method to find a local minimum by exploiting the LAS method. By relaxing the 0-1 constraint and adding a penalty function to the objective function, we have the following continuous optimization problem:

$$\begin{array}{lll} \text{(RQAP1)} & \text{min.} & F(y) = \frac{1}{2} y^\top Q y + \frac{1}{2} y^\top C(e-y) \\ & \text{s.t.} & Ay = b,\ y \geq 0, \end{array}$$

where a decision vector $y \in \Re^{n^2}$ and $Q \in \Re^{n^2 \times n^2}$ are defined by

$$\begin{aligned} y_{n(i-1)+j} &:= x_{ij}, \quad i, j \in \mathcal{N}, \\ q_{n(i-1)+j, n(k-1)+l} &:= f_{ij} d_{kl}, \quad i, j, k, l \in \mathcal{N}, \end{aligned}$$

and $e := (1, \ldots, 1)^\top \in \Re^{n^2}$, $Ay = b$ denotes the assignment constraints (5) and (6). Note that if $Ay = b$ and $y \geq 0$, then $y \leq e$. $\frac{1}{2} y^\top C(e-y)$ is a penalty term for the 0-1 constraint, where $C := \text{diag}\{c_1, \ldots, c_{n^2}\}$ and $c_1, \ldots, c_{n^2}$ are positive constants. If $c_1, \ldots, c_{n^2}$ are sufficiently large such that

$$\sum_{j=1, j\neq i}^{n^2} q_{ij} < c_i,\ i \in \mathcal{N}, \tag{7}$$

then (RQAP1) has only 0-1 valued solutions [12]. Moreover, (RQAP1) and (QAP) have the same objective function value at each 0-1 valued solution.

Now, we apply the LAS to (RQAP1), then the update dynamics is given by

$$y(t+1) = y(t) - \beta \frac{Y(t)\Pi^{\perp}_{AY(t)} Y(t) \nabla F(y(t))}{\|\Pi^{\perp}_{AY(t)} Y(t) \nabla F(y(t))\|_\infty}, \tag{8}$$

where $Y(t) := \mathrm{diag}\{y_1(t),\ldots,y_{n^2}(t)\}$, and $\Pi^{\perp}_{AY(t)}$ denotes the projection matrix onto Ker $AY(t)$. If β is selected from $(0,1]$, then $y(t+1)$ obtained by (8) always satisfies all the constraints of (RQAP1). Hence, additionally, if condition (7) is satisfied, the sequence generated by (8) converges to a local minimum for (QAP).

LS Procedure

Step 0 Starting point y_{ST} is given. Set $y(0) := y_{ST}$ and $s := 0$. Select $\beta \in (0,1]$.

Step 1 Update $y(s+1)$ by (8).

Step 2 If the termination criterion is satisfied, then find the local minimum y_{LS} corresponding to $y(s+1)$. Otherwise, let $s := s+1$ and go to **Step 1**.

Note that the LS procedure can be used as a single metaheuristic method. In this paper, we use this method not only as a LS procedure in CMS but also a single solver called a simple LS (SLS), where penalty parameters c_i, $i = 1, \cdots, n^2$ are gradually increased to be satisfy (7) from 0. In a later subsection for numerical experiments, we compare three CMS methods and this SLS.

5.3 GS Procedure with LAS Generator

In order to generate a chaotic trajectory, similarly to the LS, the 0-1 constraint of (QAP) is relaxed and the barrier function $B(y)$:

$$B(y) = \sum_{i=1}^{n^2} \{\exp(-\delta(y_i - \gamma_l)) + \exp(\delta(y_i - \gamma_h))\}$$

is added to the objective function, where δ is a positive constant and γ_l and γ_h are nonnegative constants such that $0 \leq \gamma_l << 0.5 << \gamma_h \leq 1$. Then, we have

$$\begin{array}{lll} \text{(RQAP2)} & \text{min.} & E(y) = \frac{1}{2} y^{\mathrm{T}} Q y + B(y) \\ & \text{s.t.} & Ay = b, \ y \geq 0. \end{array}$$

We apply LAS to (RQAP2) and have the dynamics :

$$y(t+1) = y(t) - \beta \frac{Y(t)\Pi^{\perp}_{AY(t)} Y(t) \nabla E(y(t))}{\|\Pi^{\perp}_{AY(t)} Y(t) \nabla E(y(t))\|_\infty}. \tag{9}$$

Then, a sequence generated by the GS procedure with (9) can be expected to be chaotic in the sense of Li-York, similarly to the case we mentioned in section 3. We approximately calculated the Lyapunov exponents [11] of the sequences obtained by (9) and verified that obtained sequences were chaotic.

Moreover, we improve the GS procedure by switching two step-sizes, β_h and β_l ($\beta_h > 1$, $0 < \beta_l \leq 1$), as mentioned in section 4.2.

5.4 GS Procedure with SD Generator

For the GS procedure with SD method, the following relaxed problem is used instead of (RQAP2):

$$\begin{array}{lll} \text{(RQAP3)} & \text{min.} & E(y) = \frac{1}{2} y^{\mathrm{T}} Q y + B(y) + \frac{1}{2} c_{SD} \|Ay - b\|^2 \\ & \text{s.t.} & 0 \leq y \leq e, \end{array}$$

where a penalty function $\frac{1}{2}c_{SD}\|Ax-b\|^2$ is added so that local minima for (RQAP3) always satisfy all constraints of (RQAP1). Because in CMS-SD, a starting point of LS found by this GS procedure have to satisfy the constraints of (RQAP1). Although a large c_{SD} ensures that the obtained starting point is always feasible for (RQAP1), it makes difficult to search for a solution extensively in the GS procedure. Thus, c_{SD} is selected to be not so large. If an obtained starting point is infeasible for (RQAP1), a LS procedure starts from the projection of the point onto the feasible region of (RQAP1), which requires to solve a quadratic programming problem. In addition, note that this model has the rectangular constraint $0 \leq y \leq e$ which is different from constraints in (RQAP1) and (RQAP2). Because for an optimization problem with the rectangular constraint, the sigmoid function can be easily used as a diffeomorphic mapping.

5.5 Starting Condition of LS

As the starting condition of the LS procedure, we use the following three conditions:

1 The objective function value $E(y(t))$ at a current point in GS procedure is less than $E_{th}(t)$.

2 Distances between a current point and the preceding two starting points of LS procedure are greater than $d_{th}(>0)$.

3 The objective function value $E(y(t))$ at a current point is less than $E(y(t-1))$ and $E(y(t+1))$, those at the preceding and the next points.

If all these conditions are satisfied at a point obtained by the GS procedure, we start a LS procedure.

Condition 1 means that if the objective function value $E(y(t))$ of (QAP2) at a current point in the GS procedure is less than a criterion value $E_{th}(t)$, a desirable local minimum can be expected to be found near the point. It is a natural concept that $E_{th}(t)$ is set to be the least objective function value obtained by the GS procedure within t iterations. However, since in many cases, the condition is too tight to search for a solution widely, we relaxed it by using the following rule:

$$E_{th}(t) = f_{\min}(t) + \lambda(t)(E_{\max}(t) - f_{\min}(t)), \quad (10)$$

$$\lambda(t) = \min\{1,\ \lambda_0 + (t - t_{last})\lambda_1\}, \quad (11)$$

where $f_{\min}(t)$ is the least objective function value of (QAP) obtained by the GS procedure within t iterations, $E_{\max}(t)$ is the greatest objective function value of (RQAP2) obtained within t iterations, and t_{last} denotes the last iteration of the GS at which the LS procedure started. In (10), E_{th} is regularized by $f_{\min}(t)$, $E_{\max}(t)$ and a relaxation parameter $\lambda(t) \in (0,1]$. Furthermore, by (11), we tighten the criterion with positive constants, λ_0 and λ_1, at some iterations just after a LS procedure started. The condition 2 implies that if distances between the candidate of a starting point and the preceding two starting points of LS procedures are not sufficiently large, a LS procedure form the candidate point may find again the same local minimum. The condition 3 denotes that until the GS procedure finds a point sufficiently close to a local minimum, a LS procedure does not start. It is illustrated in Figure 4. Each parameter in these conditions is selected in preparatory experiments: $\lambda_0 = 0.2$, $\lambda_1 = 0.02$ and $d_{th} = 1.0$.

Now, CMS-SLAS can be summarized as follows.

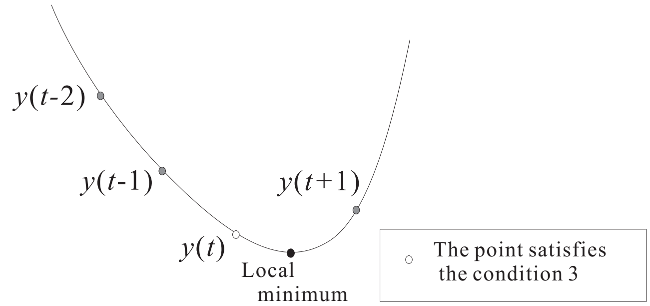

Figure 4: Condition 3

Algorithm: CMS-SLAS

Step 0 Set $t := 0$ and select $y(0)$ in the feasible region. Execute **LS** from $y_{ST} := y(0)$ and obtain y_{LS}. Let $y_{\min} := y_{LS}$.

Step 1 Update $\bar{y}(t+1)$ by (9) with β_h.

Step 2 If $\bar{y}(t+1)$ is feasible, let $y(t+1) := \bar{y}(t+1)$. Otherwise, update $y(t+1)$ by (9) with β_l.

Step 3 If starting conditions of LS are satisfied, execute **LS** with $y_{ST} := y(t)$. If $F(y_{LS}) < F(y_{\min})$, then $y_{\min} := y_{LS}$.

Step 4 If the termination criterion is satisfied, then go to **Step 5**. Otherwise, let $t := t+1$ and go to **Step 1**.

Step 5 Let $y_{ST} := y(t+1)$ and execute **LS**. If $F(y_{LS}) < F(y_{\min})$, then $y_{\min} := y_{LS}$ and find the closest 0-1 valued solution $x \in \{0,1\}^n$ to $y_{\min}$.

5.6 Comparison of the Scattered Index of Generated Sequences

In this and next subsections, we reports the results of numerical experiments, which were conducted under the same computer environment as in subsection 4.3.

In this subsection, we compare sequences obtained by SD, LAS and SLAS generators. Then, although it is worth to compare them from the viewpoint of the efficiency to solve the QAP, the accurate estimation is extremely difficult. Thus, in this paper, we simply focus on the magnitude of the spread of points of the sequence in the feasible region and consider an index calculated by the following simple iterative technique. We call it the scattered index. At first, let us consider a set whose element is the initial point of a sequence. If the next generated point is not contained in the ε-neighborhood of every point in the set, where the ε-neighborhood of the point x denotes a set $\{y \mid \|x-y\| < \varepsilon\}$ for a positive constant ε, then the point is added to the set. Finally, the cardinality of the obtained set is regarded as the scattered index of the sequence.

In numerical experiments, we compared those indices of sequences obtained by three generators for problems Chr12a and Chr20c, which are benchmark problems from Quadratic Assignment Problem Library (QAPLIB) [2]. Tables 3 and 4 show the average index in 30 trials, when the size of the neighborhood ε was varied. From tables, we can see that LAS and SLAS generated more widely scattered points than SD, and that the average index of the sequences generated by SLAS is slightly larger than that by LAS. This results show that the proposed LAS generator and the switched step-size technique work efficiently.

Table 3: Comparison of the scattered index of each method: Chr12a

ε	0.1	0.5	1.0	2.0
CMS-SD	3.1	3.0	3.0	2.0
CMS-LAS	91.5	26.3	9.4	3.1
CMS-SLAS	95.3	30.1	10.6	3.3

Table 4: Comparison of the scattered index of each method : Chr20c

ε	0.1	0.5	1.0	2.0	5.0
CMS-SD	3.0	3.0	3.0	2.0	2.0
CMS-LAS	96.9	29.1	12.4	4.8	1.0
CMS-SLAS	98.1	29.7	12.1	4.9	1.0

5.7 Application to QAP

In this subsection, through numerical experiments, we compare the SLS and CMS's with three chaos generators (SD, LAS and SLAS) for the QAP. We applied these four methods to four benchmark problems from QAPLIB, Chr12a, Chr20c, Tai40b and Lipa60b, and compared them. Chr12a, Chr20c, Tai40b and Lipa60b problems have 12×12, 20×20, 40×40 and 60×60-dimensional decision vectors, respectively. All parameters in four methods were selected in preparatory experiments as follows: $\beta = 0.9$ in SLS, $c_{SD} = 100$ in CMS-SD, $\beta_h = 1.2$ in CMS-SLAS, and $\delta = 50$, $\gamma_l = 0.05$, $\gamma_h = 0.85$ and $\beta_l = 0.9$ in CMS-SD, CMS-LAS and CMS-SLAS,

Tables 5 and 6 show results of four methods for Chr12a and Chr20c, and Tables 7 and 8 show results of two methods for Tai40b and Lipa60b. F_{ave} denotes the average of objective values obtained by each method, where the number of trials for SLS was 10000, that for CMS-SD was 100 and that for CMS-LAS and CMS-SLAS was 500.

For Chr12a, SLS and CMS-SLAS are superior to other methods in terms of searching for the best solution, while CMS-LAS and CMS-SLAS obtained the good average of objective function values. After considering all the factors, CMS-SLAS has the best performance. For Chr20c, the result is similar to Chr12a. On the whole, CMS-SLAS outperforms other methods. For Tai40b and Lipa60, since each trial requires enormous CPU time, especially in CMS-SD, we selected two better methods, CMS-SLAS and SLS, and compared them. Tables 7 and 8 show that CMS-SLAS found the better solution than SLS, while the former requires more CPU time than the latter. Therefore, we compared SLS with CMS-SLAS within the same CPU time. Table 9 shows how many trials one method finds the better solution than the other within the same CPU time (1.20(sec) for Chr12a, 16.00(sec) for Chr20c, 1650(sec) for Tai40c and 1800(sec) for Lipa60b). It can be seen that the number of trials in which CMS-SLAS is superior to SLS becomes larger as the size of problem increases.

These results indicates that the proposed method is superior to the conventional method in generating a desirable chaotic sequence to solve the QAP, especially in the case of the large-scale problem.

Table 5: Result for Chr12a

	# of LS	CPU time (sec)	F_{ave}	F_{best}	F_{worst}
SLS	-	0.22	17974.6	9552	29944
CMS-SD	2.9	0.65	18595.8	12360	29996
CMS-LAS	5.2	1.20	16177.2	9562	27556
CMS-SLAS	5.3	1.20	16049.6	9916	24842

Table 6: Result for Chr20c

	# of LS	CPU time (sec)	F_{ave}	F_{best}	F_{worst}
SLS	-	3.23	30949.6	17752	51466
CMS-SD	3.0	14.45	25403.2	20876	28688
CMS-LAS	5.1	15.26	26651.0	16868	45288
CMS-SLAS	5.1	15.17	26081.6	16540	41640

Table 7: Result for Tai40b

	# of LS	CPU time (sec) ($\times 10^2$)	F_{ave} ($\times 10^8$)	F_{best} ($\times 10^8$)	F_{worst} ($\times 10^8$)
SLS	-	3.66	8.93	7.67	10.24
CMS-SLAS	6.4	17.55	8.23	7.47	8.94

6 Conclusion

In this paper, we have pointed out a drawback of the conventional chaos generator based on the steepest descent method, which has been used in many chaotic metaheuristic methods for the global optimization. In order to overcome the drawback, we have proposed a new chaos generator exploiting the affine scaling method. In addition, we have improved it by using the switched step-size technique. Since the proposed method is suitable to the problem having many local minima on the boundary of the feasible region, we have applied the chaotic multi-start method with the proposed generator to problems of this kind, the minimization problem of a concave function and the quadratic assignment problem, QAP. For the former problem, we have confirmed that the proposed method can effectively find better solutions than the conventional method. For the QAP, we have extended the chaos multi-start method, which is a solver for the continuous global optimization problem, and applied the extended method with the proposed chaos generator to the problem. Through some numerical experiments, we have verified that the sequence generated by the proposed

Table 8: Result for Lipa60b

	# of LS	CPU time (sec) ($\times 10^3$)	F_{ave} ($\times 10^6$)	F_{best} ($\times 10^6$)	F_{worst} ($\times 10^6$)
SLS	-	3.5	3.07	3.05	3.10
CMS-SLAS	6.4	18.1	3.02	3.01	3.03

Table 9: Comparison CMS-SLAS with SLS within the same CPU time

	Chr12a	Chr20c	Tai40b	Lipa60b
SLS	4	2	1	0
CMS-SLAS	6	8	9	10

method was more widely scattered in the feasible region than that by the conventional method for the QAP. Moreover, we have seen that the proposed methods could effectively find better solutions than the conventional method for the problem. Those results indicate good performance of the proposed chaos generator.

However, in this paper, since we have mainly compared the efficiency of chaos generators, the performance of the extended CMS for the QAP has not been verified enough and solutions obtained in numerical experiments were not so satisfactory. Hence, the further investigation is required. At the same time, we should compare the CMS with the proposed generator with other chaotic methods or other metaheuristic methods such as the stochastic method or the tabu search.

References

[1] J. Bonnans and M. Bouhtou, The trust region affine interior point algorithm for convex and nonconvex quadratic programming, *RAIRO Oper. Res.* 29 (1995) 195–217.

[2] R. E. Burkard, S. E. Karisch and F. Rendl, QAPLIB-A quadratic assignment problem library, *J. Global Optim.* 10 (1997) 391–403.

[3] L. Chen and K. Aihara, Chaos and asymptotical stability in discrete-time neural networks, *Phys. D* 104 (1997) 286–325.

[4] M. Hata, Euler's finite difference scheme and chaos in $\Re^n$, in *Proc. Japan. Acad. Ser. A* 58 ,1982, pp. 178–181.

[5] S. Ishii and M. Sato, Constrained neural approaches to quadratic assignment problems, *Neural Networks* 11 (1998) 1073–1082.

[6] T. C. Koopmans and M. J. Beckmann, Assignment problems and the location of economics activities, *Econometrica* 25 (1957) 53–76.

[7] T. Kwok and K. A. Smith, Experimental analysis of chaotic neural network models for combinatorial optimization under a unifying framework, *Neural Networks* 13 (2000) 731–744.

[8] K. Masuda and E. Aiyoshi, Solution to combinatorial problems by using chaotic global optimization method on a simplex, in *Proc. the 41st SICE Annual Conference*, 2002, pp. 1313–1318.

[9] H. Nozawa, A neural network model as a globally coupled map and applications based on chaos, *Chaos* 2(3) (1992) 377–386.

[10] Y. Obita, K. Tatsumi and T. Tanino, A global optimization method using chaotic dynamics with local search processes, in *Proc. the third International Conference on Optimization and Control with Applications*, 2004, pp. 79.

[11] S. Sato, M. Sano and Y. Sawada, Practical methods of measuring the generalized dimension and the Lyapunov exponent in high dimensional chaotic systems, *Progr. Theoret. Phys.* 77(1) (1987) 1–5.

[12] K. Tatsumi, Y. Yagi and T. Tanino, Improved projection Hopfield network for the quadratic assignment problem, in *Proc. the 41st SICE Annual Conference*, 2002, pp. 2295–2300.

[13] I. Tokuda, K. Onodera, R. Tokunaga, K. Aihara and T. Nagashima, Global bifurcation scenario for chaotic annealing dynamical System which solves optimization problem and analysis on its optimization capability, (in Japanese) *Journal on IEICE* (A) J80-A(6) (1997) 936–948.

[14] T. Tsuchiya and M. Muramatsu, Global convergence of a long-step affine scaling algorithm for degenerate linear programming problems, *SIAM J. Optim.* 5(3) (1995) 525–551.

[15] L. Wang and K. Smith, On chaotic simulated annealing, *IEEE Trans. Neural Networks* 9(4) (1998) 716–718.

Manuscript received 24 October 2005
revised 23 January 2006, 24 February 2006
accepted for publication 6 February 2006

KEIJI TATSUMI
Division of Electrical, Electronic and Information Engineering,
Graduate School of Engineering, Osaka University,
Yamada-Oka 2-1, Suita, Osaka 565-0871, Japan
E-mail address: tatsumi@eei.eng.osaka-u.ac.jp

YOUSUKE YAMAMOTO
Division of Electrical, Electronic and Information Engineering,
Graduate School of Engineering, Osaka University,
Yamada-Oka 2-1, Suita, Osaka 565-0871, Japan
E-mail address: yamamoto@sa.eie.eng.osaka-u.ac.jp

TETSUZO TANINO
Division of Electrical, Electronic and Information Engineering,
Graduate School of Engineering, Osaka University,
Yamada-Oka 2-1, Suita, Osaka 565-0871, Japan
E-mail address: tanino@eei.eng.osaka-u.ac.jp