Available online at www.sciencedirect.com

SCIENCE @ DIRECT®

Chaos, Solitons and Fractals 21 (2004) 933–941

CHAOS
SOLITONS & FRACTALS

www.elsevier.com/locate/chaos

Application of chaos in simulated annealing

Ji Mingjun *, Tang Huanwen

Department of Applied Mathematics, Dalian University of Technology, Dalian 116024, People's Republic of China

Accepted 8 December 2003

Abstract

Simulated annealing (SA) has been applied with success to many numerical and combinatorial optimization problems in recent years. SA has a rather slow convergence rate, however, on some function optimization problems. In this paper, by introducing chaotic systems to simulated annealing, we propose a optimization algorithm named chaos simulated annealing (CSA). The distinctions between CSA and SA are chaotic initialization and chaotic sequences replacing the Gaussian distribution. Simulation results of typical complex function optimization show that CSA improves the convergence and is efficient, applicable and easy to implement. In addition, we discuss the advantages of CSA, and show the reasons why CSA performs better than SA.

1. Introduction

Since Lorenz [1] found the first canonical chaotic attractor in 1963, chaos has been extensively studied within the engineering scientific and mathematical communities [2]. Chaos has been found to be useful, or has great potential to be useful, in many disciplines such as high-performance circuits and devices, collapse prevention of power systems, information processing, etc. [2–4]. Chaos is apparently an irregular motion, seemingly unpredictable random behavior exhibited by a deterministic nonlinear system under deterministic conditions. Roughly speaking, a nonlinear system is said to be chaotic if it exhibits sensitive dependance on initial conditions and has an infinite number of different periodic responses. It may happen that small differences in the initial conditions produce very great ones in the final output. A small error in the former will produce an enormous error in the latter [5].

Although chaos has received growing attention, few papers study application of chaos in optimization methods [6,7]. In this paper, we propose a optimization algorithm named chaos simulated annealing (CSA). The key idea of CSA is to take full advantages of ergodic property and stochastic property of chaotic system and replace the Gaussian distribution by chaotic sequences in simulated annealing (SA).

Simulated annealing was firstly proposed by Kirkpatrick et al. [8] as a method for solving combinatorial optimization problems. The name of the algorithm derived from an analogy between the simulation of the annealing of solid first proposed by Metropolis et al. [9], and the strategy of solving combinatorial optimization problems. Annealing refers to a process of cooling material slowly until it reaches a stable state. Starting from an initial state, the system is perturbed at random to a new state in the neighborhood of the original one, for which a change of ΔE in the objective function value takes place. In a minimization process if the change ΔE is negative or zero then the transformation to the new state is accepted. If $\Delta E > 0$, the transformation is accepted with a certain probability of $p(\Delta E) = \exp\left(\frac{-\Delta E}{k_b T}\right)$, where T is a control parameter corresponding to the temperature in the analogy and k_b is Boltzmann's constant. The change

[*] Corresponding author.
E-mail address: jmj012001@yahoo.com.cn (J. Mingjun).

0960-0779/$ - see front matter © 2003 Elsevier Ltd. All rights reserved.
doi:10.1016/j.chaos.2003.12.032

ΔE in the objective function value corresponds to the change in the energy level (in the analogy) that occurs as the temperature T decreases. SA gives us a mechanism for accepting small increases in the objective function value, controlling though the probability of acceptance $p(\Delta E)$ through the temperatures.

A pseudo-code of the simulated annealing procedure is given as follow [10–12].

```
Input : A problem instance
Output : A (sub-optimal) solution
    1. Generate an initial solution at random and initialize the temperature $T$.
    2. while ($T > 0$) do
      (a) while (thermal equilibrium not reached) do
         (i) Generate a new solution based on the Gaussian distribution and evaluate the change in energy level $\Delta E$.
        (ii) If $\Delta E \leqslant 0$ update current state with new state.
       (iii) If $\Delta E > 0$ update current state with new state with probability $\mathrm{e}^{(\frac{-\Delta E}{k_\mathrm{b}T})}$, where $k_\mathrm{b}$ is a constant.
      (b) Decrease temperature $T$ according to annealing schedule.
    3. Output the solution having the lowest energy.
```

In this paper, by applying chaotic maps to simulated annealing, we put forward a new algorithm named the chaos simulated annealing (CSA) which is superior to classical global methods. The rest of this paper is organized as follows. In Section 2 we introduced two chaotic systems. In Section 3 we describe our algorithm. By evaluating the six benchmark functions the results of CSA comparing with the classical global methods are presented in Section 4. In Section 5 we analyze the reasons why CSA is superior to SA and discusses the advantage and disadvantage of CSA by evaluating a multi-minima problem.

2. Chaotic system

Above all, we introduce two chaotic systems.

The first chaotic system can be produced by the following well-known *one-dimensional logistic map* defined by

$$z_{k+1} = f(\mu, z_k) = \mu z_k(1 - z_k), \quad z_k \in [0, 1], \quad k = 0, 1, \ldots, \tag{2.1}$$

where z_k is the value of the variable z at the kth iteration, z_k in the interval $[0, 1]$, μ is a so-called bifurcation parameter of the system. The logistic map has special characters such as the ergodic property, stochastic property and sensitivity dependance on initial conditions of chaos.

The second chaotic system can be produced by a *new chaotic map* defined by

$$z_{k+1} = \eta z_k - 2\tanh(\gamma z_k)\exp(-3z_k^2), \quad k = 0, 1, \ldots \tag{2.2}$$

We derive this mapping model from chaotic neuron [6,7], where z is the internal state of the neuron, η is a damping factor of nerve membrane $(0 \leqslant \eta \leqslant 1)$ and the term $f(z) = 2\tanh(\gamma z)\exp(-3z^2)$ is a nonlinear self-feedback. Fig. 1 shows the chaotic graphs of the two maps. Where $z_0 = 0.01$, $\mu = 4$, $\gamma = 5$, $\eta = 0.9$, $k = 300$.

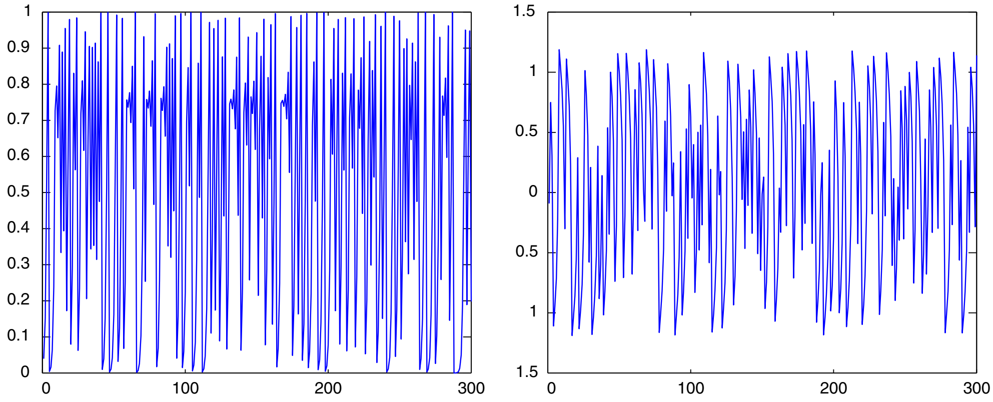

Fig. 1. The first is the logistic mapping, the second is the new chaotic mapping.

3. Chaos simulated annealing (CSA)

Consider the following continuous global optimization problem

$$(P)\quad \begin{cases} \min & f(x) \\ \text{s.t.} & x_i \in [a_i, b_i], \quad i = 1, 2, \ldots, n, \end{cases} \tag{3.1}$$

where $[a_i, b_i] \subset R$, $i = 1, 2, \ldots, n$. and f is a real-valued continuous function. Based on the chaos system and simulated annealing, the chaos simulated annealing (CSA) is proposed for solving the problem (P), which combines the features of chaotic system and simulated annealing. CSA for solving problem (P) is described as follows.

```
Step 1: Chaotic initialize. Given an initial value $z_0$. Generate the different chaotic variables $z_{k_i}$, $i = 1, 2, \ldots, n$ by formula (2.1) or (2.2), where $k_i$ is a random integer in set $\{1, \ldots, 400\}$. The initial solution $x_0 = (x_{0,1}, x_{0,2}, \ldots, x_{0,n})^{\mathrm{T}}$ is produced by the formula $x_{0,i} = a_i + (b_i - a_i) * z_{k_i}$, $i = 1, 2, \ldots, n$.
Step 2: Initialize the highest temperature $Tmax$, the lowest $Tmin$ and the iterative number $Lmax$. Set $T = Tmax$ and $m = 0$. Set the best solution $x^* = x_0$ and the best value $f^* = f(x^*)$.
Step 3: while ($T > Tmin$) do
  (a) while $m \leqslant Lmax$ do
      (1) Generate a new solution $y_m = (y_{m,1}, y_{m,2}, \ldots, y_{m,n})^{\mathrm{T}}$ based on the formula $y_{m,i} = x_{m,i} + \alpha \times (b_i - a_i) \times z_{k_m}$, $i$ is randomly chosen from the set $\{1, 2, \ldots, n\}$, where $z_{k_m}$ is a chaotic variable produced by formula (2.1) or (2.2). $k_m$ is a random integer in set $\{1, \ldots, 400\}$. $\alpha$ is a variable which decreases by the formula $\alpha = \alpha \times e^{-\beta}$ in each iteration. Other components of $y_m$ are same as that of $x_m$.
      (2) Evaluate the change in energy level $\Delta E^* = f(y_m) - f^*$ and $\Delta E = f(y_m) - f(x_m)$.
      (3) If $\Delta E^* \leqslant 0$ update the best solution $x^* = y_m$ and the best value $f^* = f(x^*)$.
      (4) If $\Delta E \leqslant 0$ update current state with new state, $x_{m+1} = y_m$.
      (5) If $\Delta E > 0$ update current state with new state with probability $\exp^{(\frac{-\Delta E}{T})}$.
      (6) $m = m + 1$.
  (b) $Lmax = Lmax + d$, $m = 0$.
  (c) Decrease temperature $T$ according to annealing schedule by formula $T := \delta \times T$.
Step 4: Output the best solution $x^*$ and the best value $f^*$.
```

The main distinctions between CSA and SA are chaotic initialization and chaotic sequences replacing the Gaussian distribution in Step 1 and Step 3. Due to the special characters of the chaotic system such as ergodic property, stochastic property and sensitivity dependance on initial conditions CSA is more likely to converge to the global optimal solution.

If z_{k_m} in CSA is generated by the one-dimensional logistic map $z_{k+1} = \mu z_k(1 - z_k)$, $z_k \in [0, 1]$, $k = 0, 1, \ldots$, CAS is named *CSA1*. If the z_{k_m} is produced by the new chaotic mapping $z_{k+1} = \eta z_k - 2\tanh(\gamma z_k)\exp(-3z_k^2)$, $k = 0, 1, \ldots$, CSA is named *CSA2*.

4. Computational results

In order to examine the performance of CSA, we apply it to function optimization problem. The six benchmark functions in our numeric experiments are chosen as below.

Example 1 GP (*Goldstein–Price function*: $n = 2$)

$$\begin{aligned} f_G(X) = {} & [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 + 6x_1x_2 + 3x_2^2)] \\ & \times [30 + (2x_1 - 3x_2)^2 \times (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)], \\ & -2 < x_i < 2, \ i = 1, 2. \end{aligned}$$

The global minimum is equal to 3 and the minimum solution is $(0, -1)$. There are four local minima in the minimization region.

Example 2 BR (*Branin*: $n = 2$)

$$f(x_1, x_2) = a(x_2 - bx_1^2 + cx_1 - d)^2 + e(1 - f)\cos(x_1) + e,$$

where $a = 1$, $b = 5.1/(4\pi^2)$, $c = 5/\pi$, $d = 6$, $e = 10$, $f = 1/(8\pi)$, $-5 \leqslant x_1 \leqslant 10$, $0 \leqslant x_2 \leqslant 15$, $x_{\min} = (-\pi, 12.275)$; $(\pi, 2.275)$; $(3\pi, 2.475)$. $f(x_{\min}) = 5/(4\pi)$. There are no more minima.

Example 3, 4 HN (*Hnrtman functions*: $n = 3, 6$)

$$f_H = \sum_{i=1}^{4} c_i \exp\left[-\sum_{j=1}^{n} \alpha_{ij}(x_j - p_{ij})^2\right] \quad 0 \leqslant x_i \leqslant 1,\ i = 1, 2, \ldots, n.$$

For $n = 3$, the global minimum is equal to -3.86 and it is reached at the point (0.114, 0.556, 0.882). For $n = 6$ the minimum is -3.32 at the point (0.201, 0.150, 0.477, 0.275, 0.311, 0.657).

Example 5 RA (*Rastrigin function*: $n = 2$)

$$g(x_1, x_2) = x_1^2 + x_2^2 - \cos(18x_1) - \cos(18x_2) \quad -1 \leqslant x_1, x_2 \leqslant 1$$

which has 50 minima in the region. The global minimum equal to -2 and the minimum point is $(0, 0)$.

i	α_{i1}	α_{i2}	α_{i3}	c_i	p_{i1}	p_{i2}	p_{i3}
$n = 3$							
1	3	10	30	1	0.3689	0.1170	0.2673
2	0.1	10	35	1.2	0.4699	0.4387	0.7470
3	3	10	30	3	0.1091	0.8742	0.5547
4	0.1	10	35	3.2	0.03815	0.5743	0.8828
	α_{i1}	α_{i2}	α_{i3}	α_{i4}	α_{i5}	α_{i6}	c_i
$n = 6$							
1	10	3	17	3.5	1.7	8	1
2	0.05	10	17	0.1	8	14	1.2
3	3	3.5	1.7	10	17	8	3
4	17	8	0.05	10	0.1	14	3.2
	p_{i1}	p_{i2}	p_{i3}	p_{i4}	p_{i5}	p_{i6}	
1	0.1312	0.1696	0.5569	0.0124	0.8283	0.5886	
2	0.2329	0.4135	0.8307	0.3736	0.1004	0.9991	
3	0.2348	0.1451	0.3522	0.2883	0.3047	0.6650	
4	0.4047	0.8828	0.8732	0.5743	0.1091	0.0381	

Example 6 SH (*Shubert function*)

$$f(x_1, x_2) = \left\{\sum_{i=1}^{5} i\cos((i+1)x_1 + i)\right\}\left\{\sum_{i=1}^{5} i\cos((i+1)x_2 + i)\right\} \quad -10 \leqslant x_1,\ x_2 \leqslant 10.$$

In the region the function has 760 local minima, 18 of which are global with $f = -186.7309$.

These functions have the characteristics as unimodal/multimodal, low-dimensional/high-dimensional. Because of these characteristics, it is difficult to seek for the global optimal solution. Using the above test functions, we have compared the performance of CSA with the performance of classical global optimization methods (see Table 2). The values of the algorithm parameters are listed in Table 1. The results of CSA and classical global optimization methods are listed in Table 3. The results of CSA1 and CSA2 are the average outcomes of 100 independent runs. The reliability is

Table 1
Parameters of numeric experiments

Function	*Tmax*	*Tmin*	δ	*Lmax*	*d*	α	β	z_0	μ (CSA1)	η (CSA2)	γ (CSA2)
GP	10	0.01	0.94	2	1	1	1.01	0.01	4	0.9	5
BR	10	0.01	0.80	2	1	1	1.01	0.01	4	0.9	5
HN3	10	0.01	0.88	2	1	1	1.01	0.01	4	0.9	5
HN6	10	0.01	0.95	2	1	1	1.01	0.01	4	0.9	5
RA	10	0.01	0.84	2	1	1	1.01	0.01	4	0.9	5
SH	10	0.01	0.98	2	1	1	1.01	0.01	4	0.9	5

Table 2
Global optimization methods used for performance analysis

Method	Name (Reference)
PRS	Pure random search [13]
MS	Multistart [14]
SA1	Simulated annealing based on stochastic differential equations [14]
SA2	Simulated annealing [14]
TS	Taboo search [15]
CSA1	This work
CSA2	This work

Table 3
Average number of objective function evaluations used by seven methods to optimize six functions

Method	GP	BR	Hn3	Hn6	RA	SH
PRS	5125	4850	5280	18 090	5964	6700
MS	4400	1600	2500	6000	N/A	N/A
SA1	5439	2700	3416	3975	N/A	241215
SA2	563	505	1459	4648	N/A	780
TS	486	492	508	2845	540	727
CSA1	377	271	360	1820	441	278
CSA2	300	281	379	1865	441	289

excellent: in each case at least 95% of runs have been successful (with the average result less than 3.5% of the global minimum) except for function GP with 50%. With this degree of precision, the global minimum in all our test functions was isolated from local minima, so that the solution can always be refined to any desired accuracy by any local optimizer.

Results in Table 3 for a standard set of test functions indicate that CSA1 and CSA2 are reliable and efficient: more so than SA1, SA2 and TS methods. CSA1 and CSA2 significantly reduce the amount of blind search characteristic of earlier techniques such as PRS and MS. So our algorithm is efficient.

Table 4
Parameters of numeric experiments

Function	*Tmax*	*Tmin*	δ	*Lmax*	*d*	α	β	z_0	μ	η	γ
JAF $n = 30$	10	0.001	0.93	2	10	1	1.01	0.01	4	0.9	5
JAF $n = 100$	10	0.001	0.93	2	10	1	1.01	0.01	4	0.9	5
JAF $n = 1000$	10	0.00005	0.90	2	20	1	1.01	0.01	4	0.9	5

Table 5
Simulated results of function JAF

JAF	Numbers of evaluations	SA		CSA1		CSA2	
		Mean best	CPU (s)	Mean best	CPU (s)	Mean best	CPU (s)
$n = 30$	96 266	0.5068	82.29	0.0207	115.1	0.0205	138.8
$n = 100$	96 266	18.377	121.6	0.0935	143.4	0.0901	205.5
$n = 1000$	133 632	19.625	399.3	9.9506	345.1	7.3727	397.6

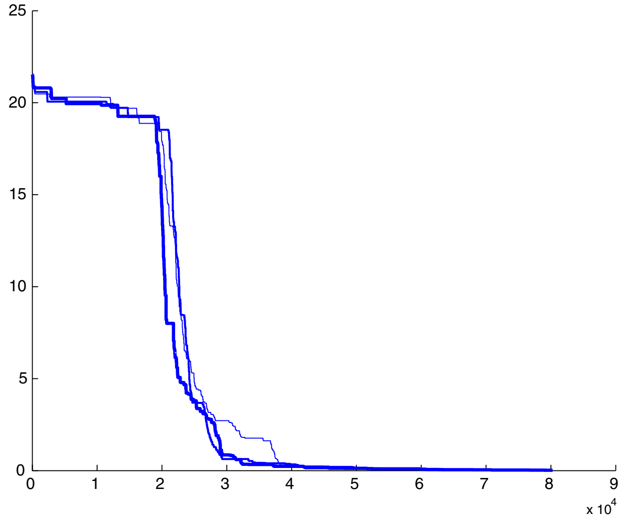

Fig. 2. Comparison of the three algorithms for JAF as $n = 30$. The thinnest line indicates the results of SA, the thinner lines indicates the results of CSA1 and the thickest lines indicates the results of CSA2.

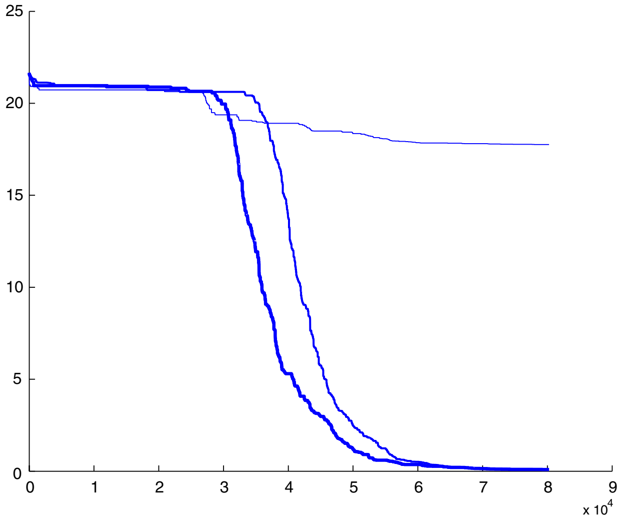

Fig. 3. Comparison of the three algorithms for JAF as $n = 100$. The thinnest line indicates the results of SA, the thinner lines indicates the results of CSA1 and the thickest lines indicates the results of CSA2.

5. Analysis of chaos simulated annealings effect

By evaluating a high-dimensional and multi-minimum function, this section shows that why CSA is of highly efficiency comparing with SA. The function is chosen as follows:

Example 7 JAF (*Ackley's Function*)

$$f(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2}\right) - \exp\left(\sqrt{\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)}\right) + 20 + e,$$

$-32 \leqslant x_i \leqslant 32$, $n = 30, 100, 1000$. The global minimum is 0 at $x = 0$.

We compare the mean value and CPU time between SA and CSA (CSA1 and CSA2). All the parameters are listed in Table 4. the numerical results of 20 independent runs of Example 5 are listed in Table 5. We present the CPU time for the example by using computation system Matlab on the computer of which CPU frequency is 1600 MHz.

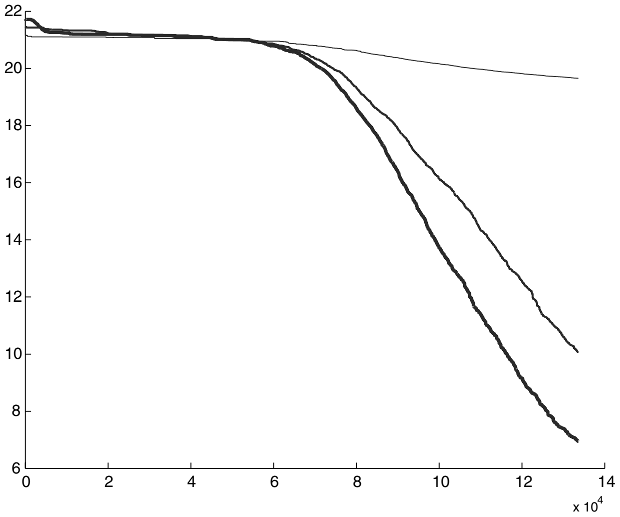

Fig. 4. Comparison of the three algorithms for JAF as $n = 1000$. The thinnest line indicates the results of SA, the thinner lines indicates the results of CSA1 and the thickest lines indicates the results of CSA2.

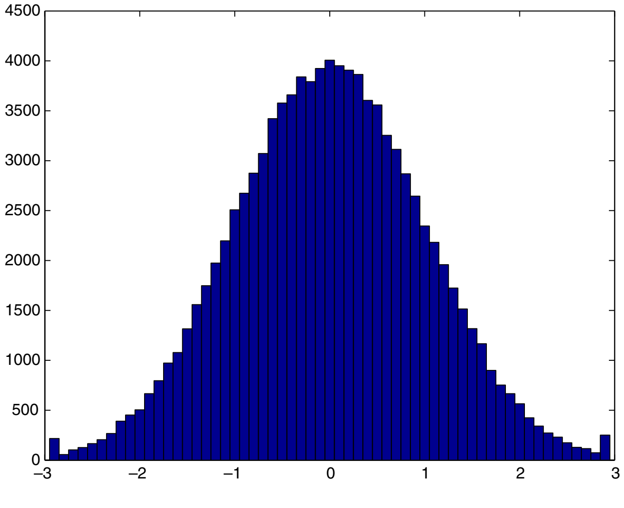

Fig. 5. Orbital points' distribution of Gaussian.

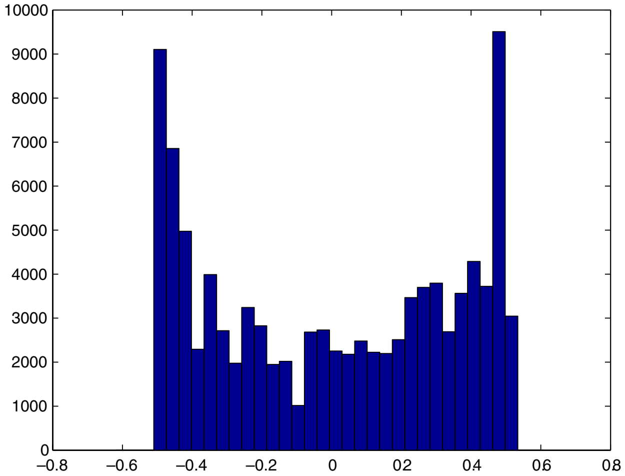

Fig. 6. Orbital points' distribution of Logistic map.

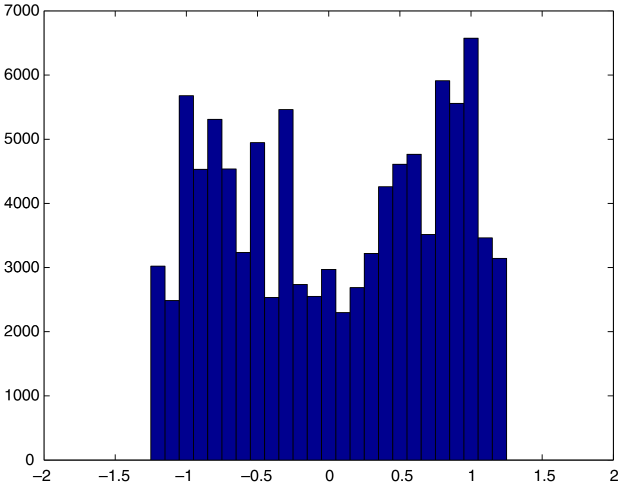

Fig. 7. Orbital points' distribution of new chaotic map.

JAF has several hundreds or thousands of minima in the region. The global minimum is zero at $x = 0$. So it is very difficult to seek for the global optimal solution. Table 5 shows CSA2 is the best of the three. For JAF, when $n = 30$, Fig. 2 shows that the effective of three Algorithms have no obvious difference. But SA can not converge to the global optimal solution by Figs. 3 and 4 as $n = 100, 1000$. Fig. 5 shows that the orbital points of Gaussian distribution mainly distribute near zero. So it is difficult for SA to escape the local optimum. The results listed in Table 5 support the conclusion. Fig. 6 shows that a lot of orbital points of Logistic map distribute near the edges, so CSA1 has advantage to escape the local optimum but is difficult to seek for the global optimal solution(see Fig. 4). The orbital points distribution of *New chaotic map* is similar to uniformly distribution and has two peaks near −0.8 and 0.8 by Fig. 7. So CSA2 has advantage to escape to local optimum and can simultaneously converge to global optimum. For the three algorithms, CSA2 gains the best precision. As we can see in Table 5, SA spends less CPU time for the low-dimensional problems and spends more CPU time for the high-dimensional problems. CSA1 spends relatively less CPU time, but could get better precision. CSA2 takes more time to converge to the global minimum. Considering CPU time and the precision, CSA1 is applicable and efficient.

6. Conclusion

Simulated annealing and chaotic system have been study for many years, but the combination of simulated annealing and chaos is in a exploring state. Only a few papers are presented [5,16]. Simulated annealing is derived from the annealing of solid, which must search the whole solution space in theory. The chaotic system has the characteristics of the ergodic property, stochastic property and sensitivity dependance on initial conditions. This inspires us with the idea of introducing chaos in the simulated annealing naturally. By applying it to classical function optimization problems, our algorithm is examined and good results are obtained. But the theory of algorithm and how to combined with chaotic system is still perfected.

Acknowledgements

We would like to thank Professor M.S. El Naschie and the referees for their advises. This research is supported by the National Nature Science Foundation of China (Grant No. 90103033).

References

[1] Lorenz EN. Deterministic non-periodic flows. J Atmos Sci 1963;20:130–41.

[2] Chen G, Dong X. From chaos to order: Methodologies, perspectives and applications. Singapore: World Scientific; 1998.

[3] Antoniou I, Akishina EP, Ivanov VV, Kostenko BF, Stalios AD. Cellular automata study of high burn-up structures. Chaos, Solitons & Fractals 2003;28:1111–28.

[4] Harb AM, Abdel-Jabbar N. Controlling hopf bifurcation and chaos in a small power system. Chaos, Solitons & Fractals 2003;18:1055–63.

[5] Yuan XH, Yuan YB, Zhang YC. A hybird chaotic genetic algorithm for short-term hydro system scheduling. Math Comput Simulat 2002;59:319–27.

[6] Luo CZ, Shao HH. Evolutionary algorithms with chaotic mutations. Control Decision 2000;15:557–60.

[7] Yang L, Chen T. Application of chaos in genetic algorithms. Commun Theor Phys 2002;38:168–72.

[8] Kirkpatrick S, Gelat CD, Vecchi MP. Optimization by simulated annealing. Science 1983;220:671–80.

[9] Metropolis N, Rosenbluth A, Rosenbluth M, et al. Equation of state calculations by fast computing machines. J Chem Phys 1953;21:1087–92.

[10] Francis RL, McGinnis LF, White JA. Facility layout and location: An analytical approach. Englewood Cliffs, NJ: Prentice-Hall; 1992.

[11] Pardalos PM, Pitsoulis LS, Mavridou TD, Resende MGC. In: Parallel search for combinatorial optimization: Genetic algorithm, simulated annealing, Tabu search and GRASP. Lecture notes in computer science, vol. 980. Springer Verlag; 1995.

[12] Mavridou TD, Pardalos PM. Simulated annealing and genetic algorithms for the facility layout problem: a survey. Comput Optim Appl 1997;7:111–26.

[13] Anderssen RS, Jennings LS, Ryan DM. Optimization. St. Luncia, Australia: University of Queensland Press; 1972.

[14] Dekkers A, Aarts E. Global optimizations and simulated annealing. Math Program 1991;50:367–93.

[15] Cvijović D, Klinowski J. Taboo search: an approach to the multiple minima problem. Science 1995;267:664–6.

[16] Chen LN, Aihara K. Chaotic simulated annealing by a neural network model with transient chaos. Neural Networks 1995;8: 915–30.