Journal of mathematics and computer Science 14 (2015) 16 - 23

Contents list available at JMCS

Journal of Mathematics and Computer Science

Journal Homepage: www.tjmcs.com

A New Dynamic Simulated Annealing Algorithm for Global Optimization

Hasan Yarmohamadi [*,1], Jahanshah Kabudian[*,2], Seyed Hanif Mirhosseini [*,3]

[*] Department of Computer Engineering and Information Technology, Razi University, Kermanshah, Iran.

[1]*hasanyarmohamadi69@gmail.com*

[2]*kabudian@razi.ac.ir - (corresponding author)*

[3]*hanif.smh@gmail.com*

Article history:
Received July 2014
Accepted October 2014
Available online November 2014

Abstract

Many problems in system analysis in real world lead to continuous-domain optimization. Existence of sophisticated and many-variable problems in this field emerge need of efficient optimization methods. One of the optimization algorithms for multi-dimensional functions is simulated annealing (SA). In this paper, a modified simulated annealing named Dynamic Simulated Annealing (DSA) is proposed which dynamically switch between two types of generating function on traversed path of continuous Markov chain. Our experiments indicate that this approach can improve convergence and stability and avoid delusive areas in benchmark functions better than SA without any extra mentionable computational cost.

Keywords: Continuous Global Optimization; Dynamic Simulated Annealing.

1. Introduction

Global optimization or global search have been used for solving problems and obtain near-global solution. Global optimization has functionality in many types of problems such as economic, engineering and social systems for cost minimization or profit maximization. Existence of complicated problems which usually lead to multi-dimensional functions with different behavior, emerge need of many optimization methods. Optimization is a process of improving the fitness of at-hand solution in regard to a target fitness function for a predefined purpose. Simulated annealing is one of these approaches which guarantee achieving of solution in continuous-domain search space. Simulated annealing (SA) has ability to escape local traps due to its stochastic acceptance function. SA can be introduced as a continuous Markov chain from beginning to the optimum solution [1].

For making the global optimum certain, the chance of choosing any state of Markov chain must be none-zero; and in SA this condition is satisfied as long as cooling procedure is slow enough. One criticism that has been imported to SA is that it does not support the parallel functionality, ergo parallel versions of SA have been proposed to overcome this objection [2]. The other criticism to

SA is lack of memory, hence memory added to SA in [3]. Although, even serial and memoryless variants of SA has better performance than Genetic algorithm in practice [4, 5]. In this work, an extension to the standard SA have been proposed, which dynamically change generation function based on progress of solution.

2. Simulated Annealing

Boltzmann annealing was essentially introduced as a Monte Carlo importance-sampling technique for doing large-dimensional path integrals arising in statistical physics problems [6]. Simulated annealing is a global search and optimization method that is based on the principles of thermodynamics and from statistical standpoint it ensures achieving global optimum if temperature decrease rate (temperature schedule) is chosen properly [8]. This method has extensive functionality in both continuous and discrete regions, simultaneously [4, 5]. SA choose a point with a density function g(x) around current state x^*. Then, the cost of new state is calculated by cost function, and at last an acceptance function h(x) decides to accept this state or not. It is clear that if the new state has lower cost, it has a greater chance to be accepted. At the beginning of the search, range of possible new points is wide and chance of acceptance and denial is almost equal due to high temperature. With decrease of temperature, range of new state will be decreased which leads to an increase in resolution. In the next part, we will explain how SA works and example some of its variants.

2.1. Boltzmann (Gaussian) Annealing.

Boltzmann or standard SA performs as follows [4, 5]:

1. Choose initial temperature (T_1) and initial point (x_1).
2. Let $k=1$, $x^*=x_1$
3. Generate a new point x randomly in neighborhood of x^* with probability density function $g(x;x^*)$,

$$g(x) = g(x;\mu) = g(x;x^*)$$

$$g(x;x^*) = (2\pi.T_k)^{-\frac{D}{2}}.\exp\left(-\frac{\| x - x^* \|^2}{2T_k}\right) \qquad (1)$$

4. Calculate cost of x using cost function *E(x).*
5. Accept new point *x* with acceptance probability *h(x)*

$$\Delta E = E(x) - E(x^*) \qquad (2)$$

$$h(x) = \frac{1}{1 + \exp\left(\frac{\Delta E}{T_k}\right)} \qquad (3)$$

6. Let $k=k+1$.
7. Reduce temperature by the following temperature schedule:

$$T_k = \frac{T_0}{\ln k} \qquad (4)$$

8. If ending criteria is met, then STOP, else go to step 3.

In this algorithm, *D* is dimension of *x* or dimension of the search space. $\|x\text{-}x^*\|$ is the Euclidian distance of two *D*-dimensional *x* and x^* vectors. Acceptance function *h(x)* is usually in the mentioned form which is called Barker function, but there is another function which is called Metropolis function and is legacy of Monte Carlo algorithm [1, 4].

$$h(x) = \min\left(1, \exp\left(-\frac{\Delta E}{T_k}\right)\right) \qquad (5)$$

In standard SA or Boltzmann Annealing, Gaussian or normal probability density function (PDF) is used as generating function and if temperature schedule is not faster than the formula in step 7, it is guaranteed that the global optimum will be found. It is obvious that the decreasing rate of the temperature is too slow and therefore reaching a good solution requires many iterations, that is why faster variants of SA has been proposed.

2.2. Fast (Cauchy Annealing)

The simulated annealing methodology can be readily extended to use any reasonable PDF or generating function *g*(*x*), without relying on the principles underlying the ergodic nature of statistical physics. Specifically, using Cauchy distribution as generating function has some definite advantages over Gaussian one (Boltzmann form) [1]. This approach called fast annealing or Cauchy annealing due to its PDF. An Isotropic Cauchy distribution PDF for a *D*-dimensional domain is as follows [9].

TABLE I. BENCHMARK FUNCTIONS

Title	Function	Range
Rosenbrock **F1**	$f(x) = \sum_{i=1}^{D-1}\left[100(x_{i+1} - x_i^{\,2})^2 + (1 - x_i)^2\right]$	$-2.048 \le x_i \le 2.048$
Rastrigin **F2**	$f(x) = 10D + \sum_{i=1}^{D}\left[x_i^{\,2} - 10\cos(2\pi x_i)\right]$	$-5.12 \le x_i \le 5.12$
Ackley **F3**	$f(x) = -20.\exp\left(-0.2.\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^{\,2}}\right) - \exp(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)) + 20 + e$	$-32.768 \le x_i \le 32.768$
Griewangk **F4**	$f(x) = \frac{1}{4000}\sum_{i=1}^{D} x_i^{\,2} - \prod_{i=1}^{D}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$-600 \le x_i \le 600$
Sphere **F5**	$f(x) = \sum_{i=1}^{D} x_i^{\,2}$	$-5.12 \le x_i \le 5.12$
Sum of different powers **F6**	$f(x) = \sum_{i=1}^{D} \lvert x_i \rvert^{i+1}$	$-1 \le x_i \le 1$

$$g(x;x^*) = \frac{\Gamma\left(\frac{D+1}{2}\right)}{\pi^{\left(\frac{D+1}{2}\right)}}.\frac{T_k}{\|\, x - x^* \,\|^2 + {T_k^2}^{\left(\frac{D+1}{2}\right)}} \qquad (6)$$

To ensure reaching global optimum, the temperature schedule shall not be faster than below:

$$T(k) = \frac{T_0}{k} \qquad (7)$$

3. Dynamic Simulated Annealing

In this approach, instead of exploiting Gaussian or Cauchy distributions, a dynamic combination of them have been used. In fact a switching criterion has been proposed to determine which distribution should be used at each step. This parameter is called *SW* and shows amount of relative dispersion of produced vectors in search space in latest steps; in other words as relative dispersion rise, *SW* will rise concordantly. *SW* is defined as follows:

$$\begin{aligned}
MED &= [\ \text{median}\{x^1_{k-M},...,x^1_{k-1},x^1_k\}\ \text{median}\{x^2_{k-M},...,x^2_{k-1},x^2_k\}\ ...\ \text{median}\{x^D_{k-M},...,x^D_{k-1},x^D_k\}\ ] \\
MED &= [\ \text{median}\{x^1_{k-M:k}\}\ \text{median}\{x^2_{k-M:k}\}\ ...\ \text{median}\{x^D_{k-M:k}\}\ ] \\
MAD &= [\ \text{median}\{|x^1_{k-M:k} - MED^1\ |\}\ \text{median}\{|x^2_{k-M:k} - MED^2\ |\}\ ...\ \text{median}\{|x^D_{k-M:k} - MED^D\ |\}\ ] \\
DEV &= \left[\ \frac{MAD^1}{\varepsilon + |\ MED^1\ |}\ \ \frac{MAD^2}{\varepsilon + |\ MED^2\ |}\ \ ...\ \ \frac{MAD^D}{\varepsilon + |\ MED^D\ |}\ \right] \\
SW &= \left\| DEV \right\|
\end{aligned} \qquad (8)$$

In above equations MED, MAD and DEV are vectors, ε is a very small positive quantity. x^d_k, MED^d and MAD^d are *d*-th element of x_k, MED and MAD vectors respectively. Furthermore, x_k is *k*-th point in *D*-dimensional space and generated in *k*-th iteration of SA algorithm. ||*SW*|| indicates Euclidean norm of *SW* vector in $\mathbb{R}^D$ and MED is a D-dimensional vector whose elements are median of corresponding elements of latest *M* point. Elements of MAD vector is Median Absolute Deviation of corresponding dimension of given vectors. To determine when the generating function is switched, a threshold parameter θ is used. The following heuristic rule have been proposed to determine the generating function (distribution selection). These properties plus the good properties of previous SA including guaranteed reach to the global optimum, makes the dynamic simulated annealing a proper approach for global optimization. Parameter θ or switch threshold is experimentally determined and in this paper it is equal to 1.

$$\begin{cases} \text{if} \ \left(\dfrac{\theta}{10} < SW \le \theta\right) \ \text{then} & g(x) = Gaussian \\ \qquad \text{otherwise} & g(x) = Cauchy \end{cases} \qquad (9)$$

TABLE II. CONVERGENCE RESAULTS OF F1 FOR 100 INDEPENDENT RUNS

Optimization method	f(x)=F1(x)			
	Minimum (best)	Maximum (worst)	Mean	Standard deviation
BA (Gaussian)	1.0022E-02	1.3231E+01	1.9644E+00	2.6656E+00
FA (Cauchy)	**9.7110E-05**	1.1155E+01	2.8121E-01	1.5707E+00
DSA	1.0934E-04	**3.3229E-01**	**9.9068E-02**	**8.6642E-02**

TABLE III. CONVERGENCE RESAULTS OF F2 FOR 100 INDEPENDENT RUNS

Optimization method	f(x)=F2(x)			
	Minimum (best)	Maximum (worst)	Mean	Standard deviation
BA (Gaussian)	4.6755E-02	7.4573E+00	3.2503E+00	1.9507E+00
FA (Cauchy)	**2.0856E-03**	9.9763E+00	**1.1956E+00**	**1.6176E+00**
DSA	4.7882E-03	**9.0036E+00**	2.1472E+00	2.2066E+00

TABLE IV. CONVERGENCE RESAULTS OF F3 FOR 100 INDEPENDENT RUNS

Optimization method	f(x)=F3(x)			
	Minimum (best)	Maximum (worst)	Mean	Standard deviation
BA (Gaussian)	7.9053E-02	2.2289E+01	1.0211E+01	7.8159E+01
FA (Cauchy)	2.6156E-03	2.0053E+01	1.2086E+01	9.3443E+00
DSA	**1.5715E-04**	**2.0046E+01**	**8.9881E+00**	**9.2223E+00**

TABLE V. CONVERGENCE RESAULTS OF F4 FOR 100 INDEPENDENT RUNS

Optimization method	f(x)=F4(x)			
	Minimum (best)	Maximum (worst)	Mean	Standard deviation
BA (Gaussian)	1.3448E+00	3.9654E+01	1.4267E+01	9.8273E+00
FA (Cauchy)	6.7997E-02	6.2810E+00	**1.1595E+00**	1.2110E+00
DSA	**6.5608E-02**	**4.3928E+00**	1.2115E+00	**9.3918E-01**

TABLE VI. CONVERGENCE RESAULTS OF F5 FOR 100 INDEPENDENT RUNS

Optimization method	f(x)=F5(x)			
	Minimum (best)	Maximum (worst)	Mean	Standard deviation
BA (Gaussian)	1.8544E-02	6.817E+00	1.8571E+00	1.8571E+00
FA (Cauchy)	3.5222E-04	**6.1917E-02**	1.6375E-02	1.5086E-02
DSA	**1.4508E-04**	6.4603E-02	**1.1898E-02**	**1.3164E-02**

TABLE VII. CONVERGENCE RESAULTS OF F6 FOR 100 INDEPENDENT RUNS

Optimization method	f(x)=F6(x)			
	Minimum (best)	Maximum (worst)	Mean	Standard deviation
BA (Gaussian)	1.2030E-03	8.2380E+00	1.4368E+00	1.4974E+00
FA (Cauchy)	5.5286E-04	1.5926E-01	2.5823E-02	3.0358E-02
DSA	**2.6052E-04**	**5.8822E-02**	**1.2160E-02**	**1.3367E-02**

Note: The best results are in boldface

4. Simulation and Results

For evaluation of methods, we used six benchmark functions mentioned in Table I. The used benchmark functions includes Rosenbrock, Rastrigin, Ackley, Griewangk, Sphere and ‘Sum of Different Powers’ functions. The Rosenbrock function (a.k.a Rosenbrock valley or banana function) is a classic benchmark which has a hard-to-reach optimum placed in a long, narrow, parabolic-shaped flat valley. Sphere function or first De Jong’s function is simplest test function. It is unimodal, convex and continuous. Rastrigin function is based on sphere function with an additional cosine term to produce frequent local minima [10]. Griewangk function is similar to Rastrigin and possess many local minima spread wide and regularly. Coefficient of sigma controls the amplitude of ripples in its surface. Ackley’s function is also a multimodal test function which can be parameterized vastly based on test requirements. We use recommended default coefficients mentioned in [10]. Sum of different powers function is a good sample of unimodal function which has a harder path to reach the global optimum in comparison to famous sphere function. All the test functions used here have global optimum equal to *f(x)* =0 which is located in origin of coordinates. Standard range for each function is mentioned in Table I. It should be noted that the problems here are two dimensional.

For detracting effect of transient random good solutions, we evaluate each condition with 50 independent runs with chaotic initial seed for generating random numbers while start point is identical for all methods per run. The results are shown in Tables II to VII based on test function on 1000 iteration. Minimum and maximum solutions are best and worst solutions among 50 runs. Also mean value and standard deviation of the solutions are reported. As it is obvious in most cases, proposed approach possesses better convergence.

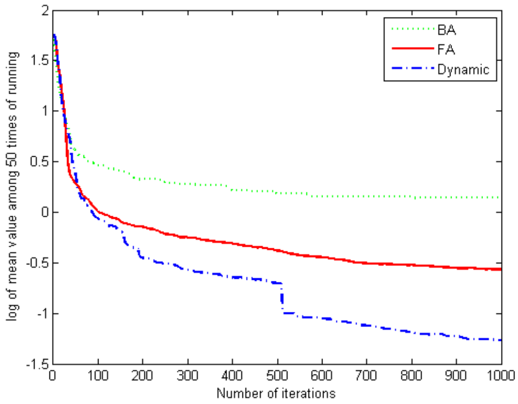


Fig. 1. Results of 2D Rosenbrock function

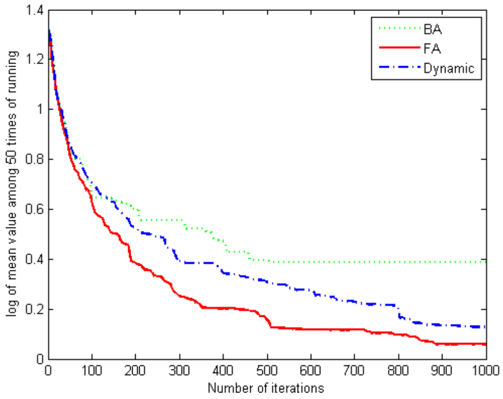
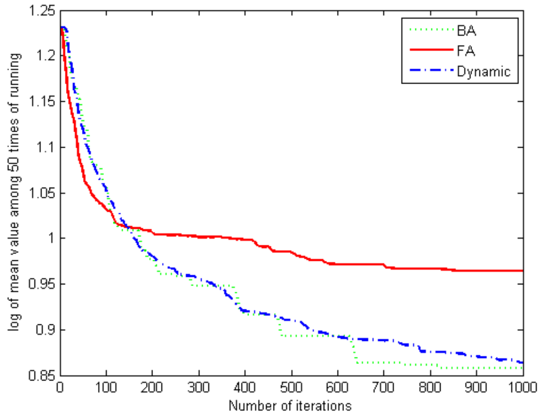


Fig. 2. Results of 2D Rastrigin function



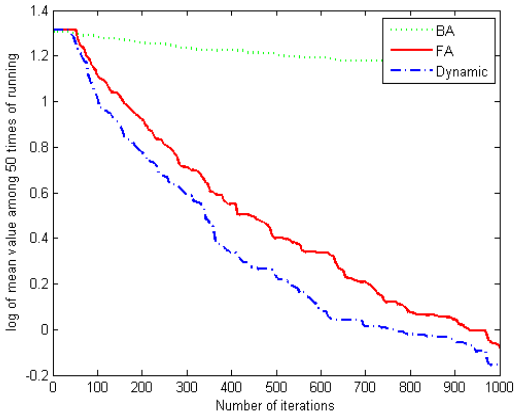
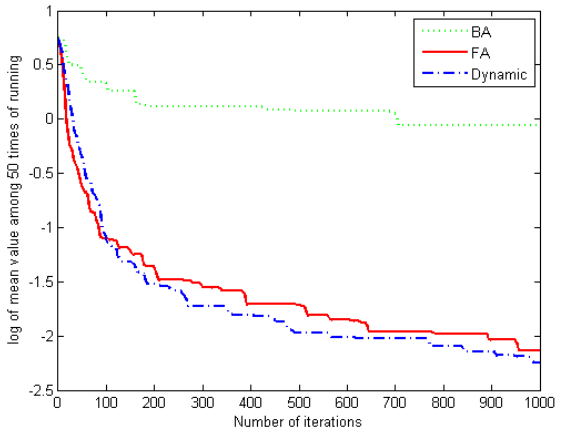


Fig. 3. Results of 2D Ackley function

Fig. 4. Results of 2D Griewangk function



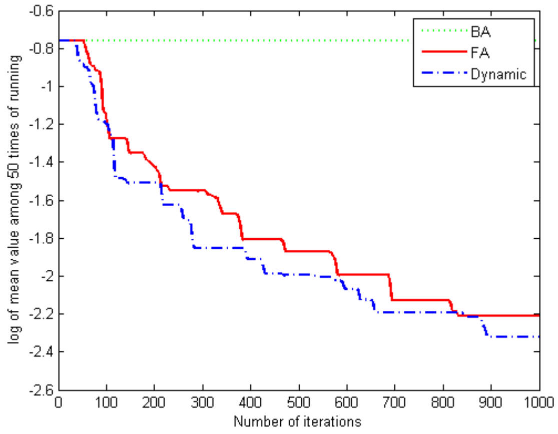


Fig. 5. Results of 2D Sphere function

Fig. 6. Results of 2D Sum of Products function

5. Conclusion

Simulated annealing is a powerful global optimization methods which from statistical standpoint, ensures achieving global optimum. All simulated annealing variants usually uses a fixed generating function for generating new points. In this paper a novel approach have been proposed which exploits a dynamic generating function which fits itself with progress of solving procedure. According to flexibility of this ‘dynamic’ method, it can be adjoined to any other variants of SA. Results of this approach in practice shows that it possesses better convergence and evasion of traps regarding of the fact that it switches between generating functions. Furthermore this approach imposes no mentionable extra time and computation complexity.

References

[1] Ingber, Lester, “*Very Fast Simulated Reannealing*” Mathematical and Computer Modeling, vol. 12 (1989), no. 8.

[2] Kimura, K., Taki, K., “*Time-Homogeneous Parallel Annealing Algorithm*”, Technical Report, Institute for New Generation Computer Technology, Tokyo, Japan, (1992).

[3] Lo C.-C., Hsu, C.-C., “*An Annealing Framework with Learning Memory*”, IEEE Trans. on SMC, Vol. 28 (1998), No. 5, pp. 648-661.

[4] Ingber, L., Rosen, B., “*Genetic Algorithms and Very Fast Simulated Reannealing: A Comparison*”, Mathematical and Computer Modelling, Vol. (1992) 16, No. 11.

[5] Ingber, L., “*Simulated Annealing: Practice versus Theory*”, Mathematical and Computer Modelling, Vol. 18 (1993), No. 11, pp. 29-57.

[6] Rajasekaran, Sanguthevar, John H. Reif, “*Nested Annealing: A Provable Improvement to Simulated Annealing*”, Theoretical Computer Science, No. 1 (1992), pp. 157-176.

[7] Smith, Kevin I., Richard Everson, DraganSavic, “*A Study of Simulated Annealing Techniques for Multi-Objective Optimization*”, (2006).

[8] Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P., “*Optimization by Simulated Annealing*”, Science, Vol. 220 (1983), No. 4598, pp. 671-680.

[9] Lindsey, J.K., Lindsey, P.J., “*Multivariate Distributions with Correlation Matrices for Nonlinear Repeated Measurements*”, Computational Statistics & Data Analysis,Vol. 50 (2006), No. 3, pp. 720-732.

[10] Molga, Marcin, CzesławSmutnicki. “*Test Functions for Optimization Needs*”, (2005).