New Simulated Annealing Algorithms

Paulo R. S. Mendonça[1]
mendonca@coe.ufrj.br

Luiz P. Calôba[1,2]
caloba@coe.ufrj.br

Abstract - **This paper introduces a new class of *D*-dimensional density probability functions to be used in Simulated Annealing algorithms and derives an appropriate cooling schedule that is proved to be inversely proportional to a previously chosen power *n* of time. This generates a new algorithm, the *n*Fast Simulated Annealing (*n*FSA), from which the Fast Simulated Annealing (FSA) is a particular case. As it will be shown, this new algorithm achieves results with an accuracy that increases with *n*, at the expenses of an initial convergence speed that decreases with *n*. This drawback is solved by the use of an adaptive algorithm, the Adaptive *n*Fast Simulated Annealing (A*n*FSA), where the parameter *n* starts at small value, producing a fast initial convergence, and is raised as the algorithm runs, finding global minima points quickly and with great accuracy.**

I. Introduction

THE drawbacks of gradient based optimization methods are well known. These drawbacks become more evident when complex and/or large dimension systems must be optimized, e.g. when training a neural network or when operating a Hopfield neural network. In these cases the algorithm is frequently trapped in local minima, and more sophisticated methods are required to escape from it. An important example of such methods is Simulated Annealing, that makes controlled use of randomness to jump out from these minima valleys.

The late results on optimization by Simulated Annealing [1], in particular the increasingly fast cooling schemes obtained, have deserved the attention of physicists and engineers. Simulated Annealing algorithms belong to the same class of methods as Neural Networks [2] and Genetic Algorithms [3], in the sense that they attempt to simulate the methods that Nature uses to solve a difficult problem. In the case of Simulated Annealing, the analogy is with the growth of a single cristal from a molten metal, that corresponds to find the global minimum of the metal internal energy, as a function of the atoms arrangement. It is known from Metallurgy that if the metal is cooled in an appropriate manner the single cristal can be built [4].

A desired property of a Simulated Annealing algorithm is a fast cooling scheme, i.e., a fast average convergence to the global minimum. The global minimum is reached if the cooling scheme is slow enough in order to guarantee that each possible state of the system is visited infinite often in time (iot) [5]. This means that the algorithm must degenerate to random search is a necessary condition to convergence, but a good algorithm must still take advantage of local information available, i. e., the cost function value and its derivatives of any order. All these features can be found in Simulated Annealing algorithms, making them a very attractive tool to solve multimodal optimization problems.

II. Convergence Speed of the Metropolis Algorithm

Although the Metropolis Algorithm [6] has been widely used as a basis for Simulated Annealing algorithms, few have been said about its convergence speed. The proposal of this section is to fill this gap.

A. The Metropolis Algorithm

Let us define a state as any possible configuration of a system. A state is said visited at instant t if the system assumes, at instant t, the configuration corresponding to this state. Let $\mathbf{x}_t$ and $E(\mathbf{x}_t)$, respectively, the state visited at instant t in a D-dimensional set of possible states and the Cost Function or Energy Function of the system associated to state $\mathbf{x}_t$. So it can be defined the Transition Acceptance Probability given by

$$P(\mathbf{x}_t \rightarrow \mathbf{x}_{t+1}) = \begin{cases} e^{\frac{-\Delta E(\mathbf{x}_t)}{kT(t)}}, & \text{if } \Delta E(\mathbf{x}_t) \geq 0; \\ 1, & \text{if } \Delta E(\mathbf{x}_t) < 0; \end{cases} \quad (1)$$

where $\Delta E(\mathbf{x}_t) = E(\mathbf{x}_{t+1}) - E(\mathbf{x}_t)$.

In [6] it was proved that if it is considered a large ensemble of systems and all states in each system can be visited at temperature T, the ensemble of states assumed by the systems converges to a Boltzmann-Gibbs distribution, i. e.,

---

[1] Paulo R. S. Mendonça and Luiz. P. Calôba are with COPPE - EE - UFRJ, Rio de Janeiro, RJ, Brazil, CP 68504, CEP 21945 - 970.

[2] Correspondence author.

0-7803-3583-X/97 $10.00 ©1997 IEEE

$$P(\mathbf{x}_\infty = \mathbf{x}) = \exp(-E(\mathbf{x})/kT)/\Sigma\exp(-E(\hat{\mathbf{x}})/kT) \quad (2)$$

where the summation is over the set of all possible states $\hat{\mathbf{x}}$.

B. Convergence Speed of the Metropolis Algorithm

Let ν_r be the number of systems of the ensemble at state r and P_{rs} the probability of the state r be selected to a transition from state s, henceforth called *state visiting probability*. Let E_r be the Energy associated to state r. Consider $P_{rs} = P_{sr} = P$. Taking into account all the systems of the ensemble, the number of state transitions from r to s, if, without loss of generality, $E_r > E_s$, will be given for $\nu_r P\exp[-(E_r - E_s)/(kT)]$. The number of systems that will make a transition from s to r is $\nu_s P$. Let $\nu_r(m)$ be the number of systems at state r after m transitions. In a matricial notation, we have:

$$\begin{bmatrix}\nu_r(m)\\ \nu_s(m)\end{bmatrix} = \begin{bmatrix}1-PB & P\\ PB & 1-P\end{bmatrix}\begin{bmatrix}\nu_r(m-1)\\ \nu_s(m-1)\end{bmatrix}. \quad (3)$$

where $B = \exp(-(E_r - E_s)/(kT))$. After some manipulations, we obtain

$$\begin{bmatrix}\nu_r(m)\\ \nu_s(m)\end{bmatrix} = \frac{1}{1+B}\times \begin{bmatrix}1+B(1-P(1+B))^m & 1-(1-P(1+B))^m\\ B-B(1-P(1+B))^m & B+(1-P(1+B))^m\end{bmatrix}\begin{bmatrix}\nu_r(0)\\ \nu_s(0)\end{bmatrix} \quad (4)$$

Eq. 4 shows the exponential convergence of Metropolis Algorithm, and also can be used in a direct proof of eq. 2.

III. Simulated Annealing

Simulated Annealing algorithms are a straightforward generalization of the Metropolis Algorithm [4], where the temperature parameter is time-varying, i. e., $T = T(t)$, and $T(t)$ converges to zero as t increases, following a cooling schedule. In this situation the Boltzmann-Gibbs distribution given by eq. 2 converges to a Dirac distribution centered at the *global* minimum of the Cost Function as $T \to \infty$. Another difference from Simulated Annealing to the Metropolis Algorithm is that in the former the states are not chosen uniformly among all the space of states, but a new state $\mathbf{x}_{t+1}$ is generated from $\mathbf{x}_t$ through the relation $\mathbf{x}_{t+1} = \mathbf{x}_t + \Delta\mathbf{x}_t$ where $\Delta\mathbf{x}_t$ is an isotropic random vector where the variance of each coordinate is proportional to $T(t)$ and whose probability density is given by $g(\Delta\mathbf{x}_t)$.

As shown in [5] the basic condition for the convergence of a Simulated Annealing algorithm is the *iot* condition, i. e., no state can never be visited. A crucial result is that this condition is equivalent to the relation

$$\sum_{t_0}^{\infty} g(\Delta\mathbf{x}_t) = \infty. \quad (5)$$

Now we have a limitation in the choice of $T(t)$, henceforth called cooling schedule, since it has, together with $g(\Delta\mathbf{x}_t)$, to satisfy eq. 5.

A. CSA and FSA

Two of the most common Simulated Annealing algorithms are the Classical Simulated Annealing [7], based on a Gaussian probability density and determined by the pair

$$g(\Delta\mathbf{x}_t) = \exp(-(|\Delta\mathbf{x}_t|^2)/2T(t))/(2\pi T(t))^{D+1};\quad T(t) = T(0)/(1+\ln(1+t)); \quad (6)$$

and the Fast Simulated Annealing [5], based on a Cauchy probability density to $\Delta\mathbf{x}_t$, determined by

$$g(\Delta\mathbf{x}_t) = \frac{(\Gamma((D+1)/2)T(t))}{(\pi(T^2(t)+|\Delta\mathbf{x}_t|^2))^{(D+1)/2}};\quad T(t) = T(0)/(1+t); \quad (7)$$

where $\Gamma(\cdot)$ is the Gamma Function and D and $|\Delta\mathbf{x}_t|$ are respectively the dimension of the space of states and the Euclidean norm of $\Delta\mathbf{x}_t$.

B. The n*Fast Simulated Annealing*

Let us consider the representation of the Cauchy density $g(\Delta\mathbf{x}_t)$ shown in eq. 7 in spherical coordinates, with $T(t) = 1$ for all t. Due to the assumption of isotropy, we obtain a one-dimensional distribution function of the magnitude parameter ρ_t. Applying the bijective mapping $\rho_t \to T(t)[(1+r_t)^n - 1]$ over ρ_t, we obtain, after some manipulations, the D-dimensional n-Cauchy probability density, denoted by $g_{nC}(r_t)$, as follows:

$$g_{nC}(r_t) = \frac{D\Gamma\left((D+1)/2\right)}{n\sqrt{\pi}\Gamma\left(D/2+1\right)} \times \times \frac{\left[\left(1+r_t/T(t)\right)^{1/n}-1\right]^{D-1}\left(1+r_t/T(t)\right)^{(1-n)/n}}{T(t)\left\{1+\left[\left(1+r_t/T(t)\right)^{1/n}-1\right]^2\right\}^{(D+1)/2}}. \quad (8)$$

If $n=1$ we obtain the Cauchy probability density in spherical coordinates. Using eq. 5 one may prove that a suitable cooling schedule for the *n*FSA is given by

$$T_{nC}(t) = T(0)/(1+t)^n . \quad (9)$$

Up to now, nothing was said about the value of $T(0)$. Since T controls the size of the tail of the probability density, a small value of $T(0)$ may excessively constrain the initial space of search, while a too large value may results in an unnecessary broad initial space of search. However, the ranges of the variables involved in an optimization problem are often known.

Nevertheless, if we want to define the space of search in a probabilistic way, eq. 8 provides us with a suitable tool. Let α be the probability a jump with size grater than L be executed. Then

$$T(0) = L/\{\{\tan[p_D^{-1}(1-\alpha)]+1\}^n - 1\}, \quad (10)$$

where

$$p_D(\psi) = \int_0^{\psi} \frac{D\Gamma\left[(D-1)/2\right]}{\sqrt{\pi}\Gamma\left(D/2-1\right)} \sin^{D-1}\theta d\theta . \quad (11)$$

From eq. 10 we see that for fixed α and L the initial temperature decays with *n*.

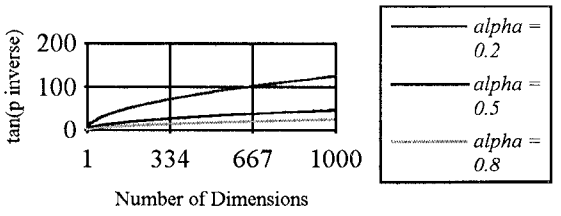

Fig. 1. Plot of $\tan(p_D^{-1})$ for D from 1 to 1000, linear scales.

In Fig. 1 we show the behaviour of $\tan(p_D^{-1}(1-\alpha))$ for three values of α and for D on the interval [1, 1000]. To derive eq. 10, the hypothesis that we are dealing with the initial temperature has no influence, so eq. 10 can also be used to develop a temperature based stopping criterion, where the final temperature value is the one that allows jumps greater than L_f only with probability α_f.

C. Paralelization of the Algorithm

The generation of the D-dimensional *n*-Cauchy random variable demands a great computational effort when $D \geq 3$. In this case, closed analytical expressions are not available. A possible solution to this problem is the use of a random vectors table. Although presenting some evident advantages, like velocity and simplicity, this approach introduces quantization errors unless a large amount of memory is available.

Another option is the paralelization of the algorithm [8], where a one-dimensional *n*-Cauchy random variable is generated to each dimension of the space of states. Now, instead of a scalar ρ_t we will have a vector $\boldsymbol{\rho}_t$ and a new probability density function $g_{dnC}(\boldsymbol{\rho}_t)$ as shown bellow:

$$g_{dnC}(\boldsymbol{\rho}_t) = \left(\frac{2}{n\pi}\right)^D \prod_{i=1}^{D} \frac{\Phi_i^{(1-n)/n}}{T_{dnC_i}(t)[1+(\Phi_i^{1/n}-1)^2]} \quad (12)$$

where $\Phi_i = 1+\rho_{i,t}/T_{dnC_i}(t)$ and

$$T_{dnC_i}(t) = [T_i(0)/(1+t)^n]^{1/D}, \; i = 1,\ldots,D, \quad (13)$$

is the set of cooling schedules that together with $g_{dnC}(\boldsymbol{\rho}_t)$ satisfy eq. 5.

IV. The Adaptive *n*Fast Simulated Annealing

A. Preliminary Results

To evaluate the performance of the *n*FSA, let us apply the parallel algorithm to minimize the Rastrigin's function $f_D(\mathbf{x})$

$$f_D(\mathbf{x}) = 10D + \sum_{i=1}^{D}\left[x_i^2 - 10\cos(2\pi x_i)\right] \quad (14)$$

where $\mathbf{x} = [x_1, x_2, \ldots, x_D]$ and $D = 100$.

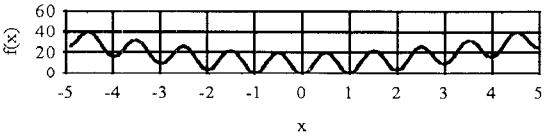

Fig.2. One-dimensional Rastrigin's function.

Fig 3. shows the evolution of the Cost Function for several values of n.

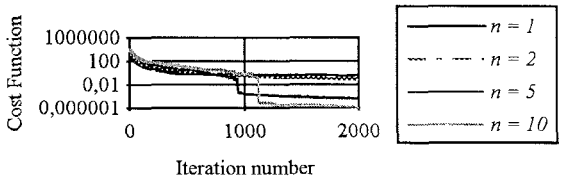

Fig. 3. Results for $n=1$, $n=2$, $n=5$ and $n=10$. The theoretical initial temperature value was used, with $\alpha = 0.8$ and $L=1$.

B. The AnFSA Algorithm

The preliminary results presented in the last subsection show an unexpected result. When the parameter n increases, the algorithm takes a longer time sampling the region, but finds the global minimum with crescent precision. For values of n greater than 5, this effect is more clear, and becomes comparable to the phenomena that occur during the phase transitions in the solidification of liquids. When the liquid (e. g. a molten metal) reaches its solidification temperature the decay on its internal energy is not followed by any further decrease in the temperature until the solidification process is complete. In other words, a small variation of the temperature around the solidification temperature corresponds to a large variation of the internal energy of the system, exactly as shown in Fig. 3.

This phenomenon can be used to develop an algorithm that works always in the critical temperature. Let us define the convergence rate at order k at iteration j, $CR_k(j)$, as

$$CR_k(j) = \sqrt{\frac{\left|\sum_{i=k}^{2k-1} E^2(j-i) - \sum_{i=0}^{k-1} E^2(j-i)\right|}{\sum_{i=k}^{2k-1} E^2(j-i)}}, \qquad (15)$$

where $E(m)$ is the value of the Cost Function at iteration m. The adaptive procedure is to increase the value of the parameter n always the convergence rate is smaller than a given fixed value r. In the minimization of the Rastrigin's function good values of k and r were empirically found as 20 and 0.01, respectively.

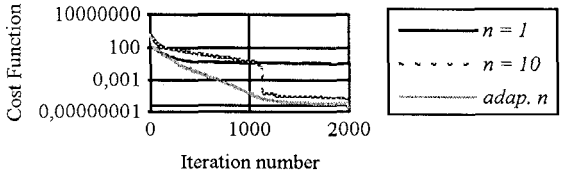

Fig. 4. Comparison between the AnFSA and FSA with $n=1$ and $n=10$.

From Fig. 4 we can see that the A*n*FSA combines a fast convergence in the initial iterations with a great accuracy, overtaking both FSA and *n*FSA with $n=10$. An important observation is that the parameter n must be adjusted also in eq. 10, resulting in an effective cooling schedule much faster than the CSA, FSA or *n*FSA ones. In this example the value of the parameter n was adaptively increased, reaching values around 50.

V. Conclusions

New Simulated Annealing algorithms are presented with cooling schedules faster than the previous CSA and FSA. An analysis of the initial and final temperatures values is done, not only providing an analytical basis to the tuning of the algorithms but also being directly used on its design.

These algorithms may represent a powerful tool in complex, multimodal or large dimension optimization problems. Possible applications are in backpropagation training and in the operation of Hopfield neural networks [9].

References

[1] D. A. Stariolo and C Tsallis "Optimization by Simulated Annealing", in *Annual Reviews of Computational Physics II*, ed. Dietrich Stauffer. Singapore: World Scientific, pp. 343-356, 1994.

[2] S. Haykin, *Neural Networks, a Comprehensive Foundation* New York: Macmillan Publishing, p. 191, 1994.

[3] L. Davis, *Handbook of Genetic Algorithms*, New York: Van Nostrand Reinhold, 1991.

[4] S. Kirkpatrick, C. D. Gelatt and M. P. Vecchi, "Optimization by Simulated Annealing", *Science*, vol. 220, pp. 671-680, 1983

[5] H. Szu and R. Hartley, "Fast Simulated Annealing", *Physics Letters A*, vol. 122, (3), (4), pp. 157-162, 1987

[6] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth and A. H. Teller, "Equation of State Calculations by Fast Computing Machines", *The Journal of Chemical Physics*, vol. 21, (6), pp. 1087-1092, 1953.

[7] S. Geman and D Geman, "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images", *IEEE PAMI*, vol. 6, pp.721-741, 1984.

[8] Lester I, "Very Fast Simulated Re-Annealing", *J. Math Comput. Modeling*, vol. 12, pp.967-973, 1989.

[9] J. A. Apolinário Jr., P. R. S. Mendonça, R. O Chaves and L. P. Calôba, "Criptanalysis of Speech Signals Ciphered by TSP Using Annealed Hopfield Neural Network and Genetic Algorithms", *Proc. of 39th MWSCAS*, Iowa, USA, 1996.