Simulated Annealing

Rafael E. Banchs

INTRODUCTION

This report discuss simulated annealing, one class of global search algorithms to be used in the inverse modeling of the time harmonic field electric logging problem. First, a brief description of its fundaments is presented. Then, a more precise description of the technique and the parameters involved in its implementation is discussed. Finally, an example is provided to illustrate the most important properties of simulated annealing.

FUNDAMENTS OF THE METHOD

The technique of simulated annealing has it fundaments on the parallelism existent between the problem of finding the minimum of a function of multiple variables and the statistical mechanics phenomenon of annealing. The term ‘annealing’ refers to the process in which a solid, that has been brought into liquid phase by increasing its temperature, is brought back to a solid phase by slowly reducing the temperature in such a way that all the particles are allowed to arrange themselves in a perfect crystallized state. Such a crystallized state represents the global minimum of certain energy function [1].

In order for annealing to occur properly, the two following conditions have to be met. First, the initial temperature has to be high enough to guarantee that the process will start from a state in which all the particles are randomly arranged into the liquid phase. Second, the subsequent cooling process has to be slow enough in order to guarantee that the particles will have time to rearrange themselves and reach thermal equilibrium at each temperature. Otherwise, if the initial temperature is not high enough or the cooling process is too fast, the annealing process will result

in a metastable glass instead of a perfect crystal. This represents a suboptimal situation in which a local minimum of the energy function has been reached.

The simulated annealing search technique bases it operation in considering the objective function of the minimization problem as the equivalent energy function of an illusory annealing process. In this way a control parameter T, that is referred as the ‘temperature’, is used to control the randomness of the searching process. It then constitutes a guided Monte Carlo technique.

The algorithm is defined in such a way that for high values of T, the search is performed totally at random; and then, when T is decreased, the search becomes more and more directive. This gives simulated annealing one of its two most valuable properties, the fact that it starts by evaluating gross features of the objective function (high temperatures) and evolves in such a way that ends by evaluating finer details in an optimal region (low temperatures). The second valuable property of simulated annealing is due to its random nature; the algorithm is defined in such a way that there exists always a possibility (either small or high, but greater than zero) of moving ‘upwards’ in the objective function. According to this, the algorithm does not necessarily exhibit the risk of getting trapped in a local minimum because there is always the possibility of jumping out of it.

The philosophy behind simulated annealing can be summarized by saying that when looking for the global minimum of a function the search must be performed by moving downwards most of the time but not necessarily always [2].

As it can be seen now, simulated annealing constitutes a very robust searching technique that presents important advantages over other searching techniques. However, there is a cost involved. The principal drawback of simulated annealing is its intensive computational requirements. Although it is true that it only requires evaluations of the objective function (and does not require any derivatives); much more of such evaluations than in other search techniques are required. Also, as it is discussed in the next section, success in locating the global minimum

is only guaranteed for an extremely slow cooling process. In practice, such a cooling process can increase computational requirements enormously.

STRUCTURE OF SIMULATED ANNEALING ALGORITHMS

The general simulated annealing algorithm can be described as an iterative procedure composed by two nested loops. The inner loop simulates the achievement of thermal equilibrium at a given temperature, so it is going to be referred as the thermal equilibrium loop. The outer loop performs the cooling process, in which the temperature is decreased from its initial value towards zero until certain convergence criterion is achieved and the search is stopped; this loop is going to referred as the cooling loop or annealing loop.

1.- Thermal equilibrium loop.

The operation of the inner loop can be described as follows. Starting with an initial model, each iteration of the inner loop computes a new model that may or may not be accepted according to certain probability. Three elements can be identified here:

1.a.- Perturbation scheme.

It defines the way in which the model is updated. First, a 'perturbation' is computed and then it is added to the existent model $\overline{x}_i$ in order to obtain the new one $\overline{x}_u$. Although the perturbation is always computed at random, it can be done by using different kinds of distributions. The simplest way is by using a uniform distribution over the feasible set in the model space; however, the main problem of this scheme is that it requires a very slow cooling process.

Alternative ways, that allow the use of faster cooling processes, involve probability distributions that change with temperature. Examples of them are the Gaussian or normal distribution:

$$f(x) = \frac{1}{\sqrt{2\pi}\ aT}\ \exp\left(-\frac{x^2}{2\,(aT)^2}\right) \tag{1}$$

and the Cauchy distribution:

$$f(x) = \frac{1}{\pi}\ \frac{aT}{(aT)^2 + x^2} \tag{2}$$

where $f(x)$ is the density function, a is an scaling factor and T is the temperature parameter. In practice, the Cauchy distribution is preferred over the normal distribution because of its flatter tails, which makes it easier to escape from local minima.

1.b.- Acceptance criterion.

It determines if the new computed model is either accepted or discarded. The most popular and common acceptance criterion is the one due to Metropolis. In the Metropolis algorithm, an 'energy variation' ΔE is computed by subtracting the value of the error function at the initial model to the value of the error at the updated model:

$$\Delta E = E(\bar{x}_u) - E(\bar{x}_i) \tag{3}$$

where $E(\bar{x})$ is the error function (objective function) evaluated at model $\bar{x}$. Then, if $\Delta E < 0$ the updated model $\bar{x}_u$ is always accepted; but if $\Delta E \geq 0$ the updated model is accepted with probability:

$$P(\Delta E) = \exp\left(-\frac{\Delta E}{T}\right) \tag{4}$$

where T is the temperature. On the other hand, if the updated model is not accepted, the new iteration will proceed with the same initial model $\bar{x}_i$.

Notice that, at high values of temperature, the probability presented in (4) presents a uniform preference for any model; while, at very low temperatures, only those models for which ΔE is very small will have a substantial chance of occurrence. In fact, except for a scaling factor, (4) represents the Boltzmann distribution, which characterizes thermal equilibrium processes in the real statistical mechanics problems.

1.c.- Thermal equilibrium achievement.

As it was discussed above, at each iteration of the thermal equilibrium loop a new ‘perturbed’ model is computed according to the perturbation scheme. Such model is then accepted or rejected according to the provided acceptance criterion and a new iteration begins. This process is repeated again and again until it is considered that ‘thermal equilibrium’ has been reached. At this point the loop is ended. Some practical strategies involve the use of a maximum number of perturbations, a maximum number of acceptances or a combination of them.

2.- Cooling loop.

The cooling or annealing loop, constitutes the outer loop of the algorithm. It starts with and initial model selected at random and an initial value of temperature To. At each iteration, the temperature is decreased in a progressive manner towards zero until certain convergence criterion is achieved. Again, three elements can be identified here:

2.a.- Initial temperature.

The initial value of the temperature parameter is of critical importance to the success of the algorithm. A low initial temperature can result in a lost of the global character of the search by restricting the search to the region of the model space around the starting point. On the other hand, a too high initial temperature will keep the algorithm performing ‘random walks’ over the model space during a large number of iterations. This will result in an unnecessary waste of valuable computational time; and, what is worse, it can result in an unsuccessful search if the total number of iterations is limited.

According to this, the initial temperature value must be defined in such a way that almost any perturbation must be accepted during the first iteration of the cooling loop. In practice, there is not an easy way to determine that. Some times, available a priori information about the problem can help to determine appropriate values for the initial temperature. However, most of the time auxiliary methods have to be used. One common way to compute an good initial temperature is to compute the values of the objective function for a set of models selected at random; then the

energy variations among all of them are computed and a value of To is estimated such that, according to the acceptance criterion, the maximum energy variation is accepted with some probability close to the unit.

Another way to overcome the problem of selecting a good initial temperature is with the help of a distortion function. As it will be discussed later in more detail, distortion functions are monotone concave functions that distort the error surface. They can be used to bound the objective function between two values, let us say for example 0 and 1. Under such a situation, the maximum possible energy variation is 1 and an initial temperature of 5 will accept it with a probability of 0.82.

2.b.- Cooling schedule.

It defines the way in which the temperature is going to be decreased. It is also of crucial importance in the success of the search. A very low cooling schedule will take to many iterations to reach the global minimum and, if the total number of iterations is limited, an unsuccessful search can result. On the other hand, a too fast cooling schedule can get the algorithm trapped in a local minimum or even in any smooth region of the error surface.

One common cooling process is the logarithmic schedule:

$$T_k = \frac{\alpha\ To}{\ln(1+k)} \tag{5}$$

where T_k is the value of the temperature at iteration k, To is the initial temperature and α is the cooling speed parameter. This schedule, has been proved to guarantee convergence to the global minimum when α=1 [3]. However, it constitutes such a slow cooling schedule that it is rarely used in practice. Although, the use of values of α smaller than 1 can speed up the process, logarithmic cooling schedules are considered in general slow.

Another common cooling schedule, and more used in practice, is the geometric schedule:

$$T_k = \alpha^k\ To \tag{6}$$

In this type of schedule, α must be smaller but close to 1. The most typical values of α are between 0.8 and 0.99; smaller values can result in a excessively fast cooling.

Finally, another popular cooling schedule is the exponential one:

$$T_k = To \ \exp\left(-\alpha\, k^{1/N}\right) \qquad (7)$$

where N is the dimensionality of the model space. This kind of schedules are very fast during the first iterations, but the speed of the exponential decay can be reduced by using values of α smaller than 1. Exponential cooling schedules are ideal to be used with temperature-dependent perturbation schemes. Figure 1 illustrates the relative speeds of the three given cooling schedules.

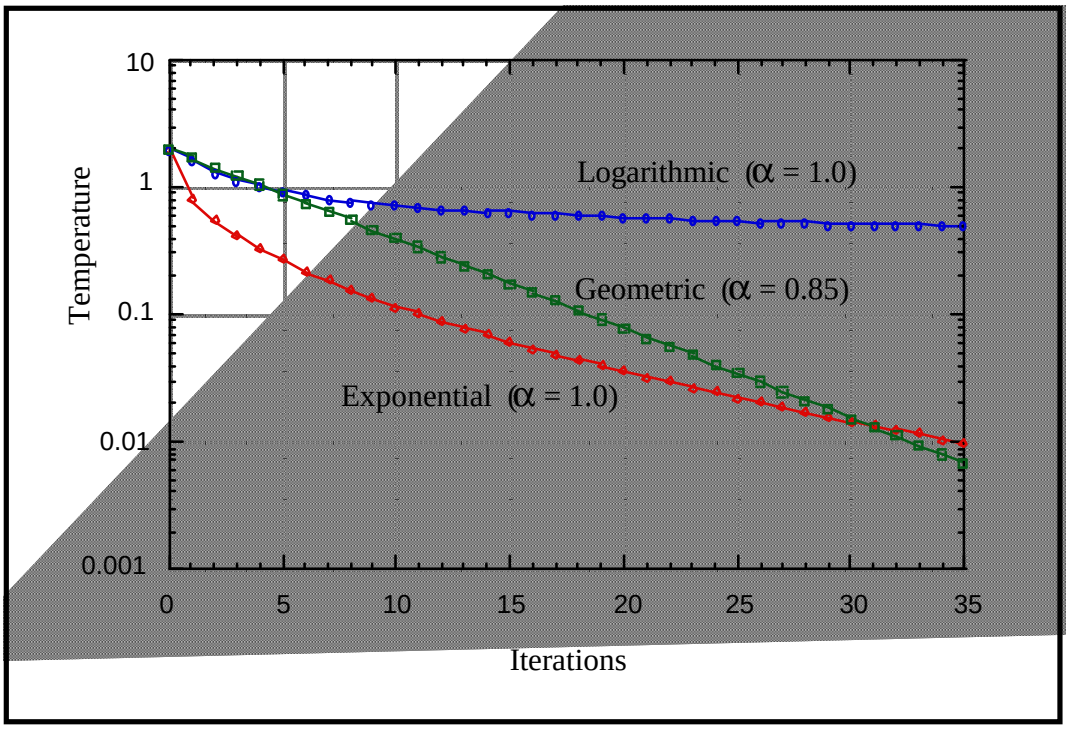

Figure 1: Cooling schedules.

2.c.- Stopping criterion.

The most appropriate stopping criterion to be used in the simulated annealing algorithm will depend on the type of search been conducted. In the case of an hybrid type of search, in which the

objective of the simulated annealing algorithm is just to provide the local search algorithm with a good starting model, a much more relaxed stopping criterion can be used. On the other hand, when an stand alone global search is intended a more careful criterion must be considered in order to obtain the desired accuracy.

A common stopping criterion is to wait until a certain defined number of acceptances is not achieved for some number of successive temperature values. This kind of criterion represents a particular case of the parameter variation condition already implemented as stopping criterion for the local search algorithms [4]. In this way, the same criterion implemented for the local search algorithms can be used for the simulated annealing. However, it is important to notice that during its final iterations, simulated annealing moves really slow or can even stop for a while. For this reason, when an stand alone global search is intended, the counter parameter K (defined in [4]) must be chosen large enough to avoid stopping the search prematurely.

ENERGY FUNCTION AND DISTORTION FUNCTIONS

As it is already known, another important factor in the success of the search is the proper definition of the energy function or objective function. For the same reasons presented in [4], the objective function selected for the implementation of the simulated annealing algorithm is the mean square error between the given data and the model response. Such a function is defined by (1) in [4]. Alternatively, its normalized version (given by (5) in [4]) can be also used. However, the algorithm implementation allows the use of distortion functions to produce alterations into the mean square error surface.

The use of monotone concave functions in order to produce a distortion of the energy function is a common practice in simulated annealing. This is done in order to improve the performance of the algorithm; and such an improvement can be obtained in various manners. The use of distortion functions generally accelerate the speed of convergence of the algorithm [5]. This is

because they can deform the energy function in such a way that differences between the local minima and the global minimum are accentuated. Also, as it was mentioned before, they can be used in order to bound the objective function and facilitate the selection of the initial temperature.

Some commonly used distortion functions are, the logarithmic distortion:

$$E_d(\bar{x}) = \ln\left(\beta E(\bar{x}) + 1\right) \tag{8}$$

the exponential distortion:

$$E_d(\bar{x}) = -\exp\left(-\beta E(\bar{x})\right) + 1 \tag{9}$$

and rational distortion:

$$E_d(\bar{x}) = E(\bar{x})^{1/\beta} \tag{10}$$

where $E_d(\bar{x})$ represents the new distorted error surface, $E(\bar{x})$ is the original objective function and β is a shrinking factor that controls the 'strength' of the deformation. β must be greater than zero.

Figure 2 presents an example of how an exponential distortion function improves a given original objective function.

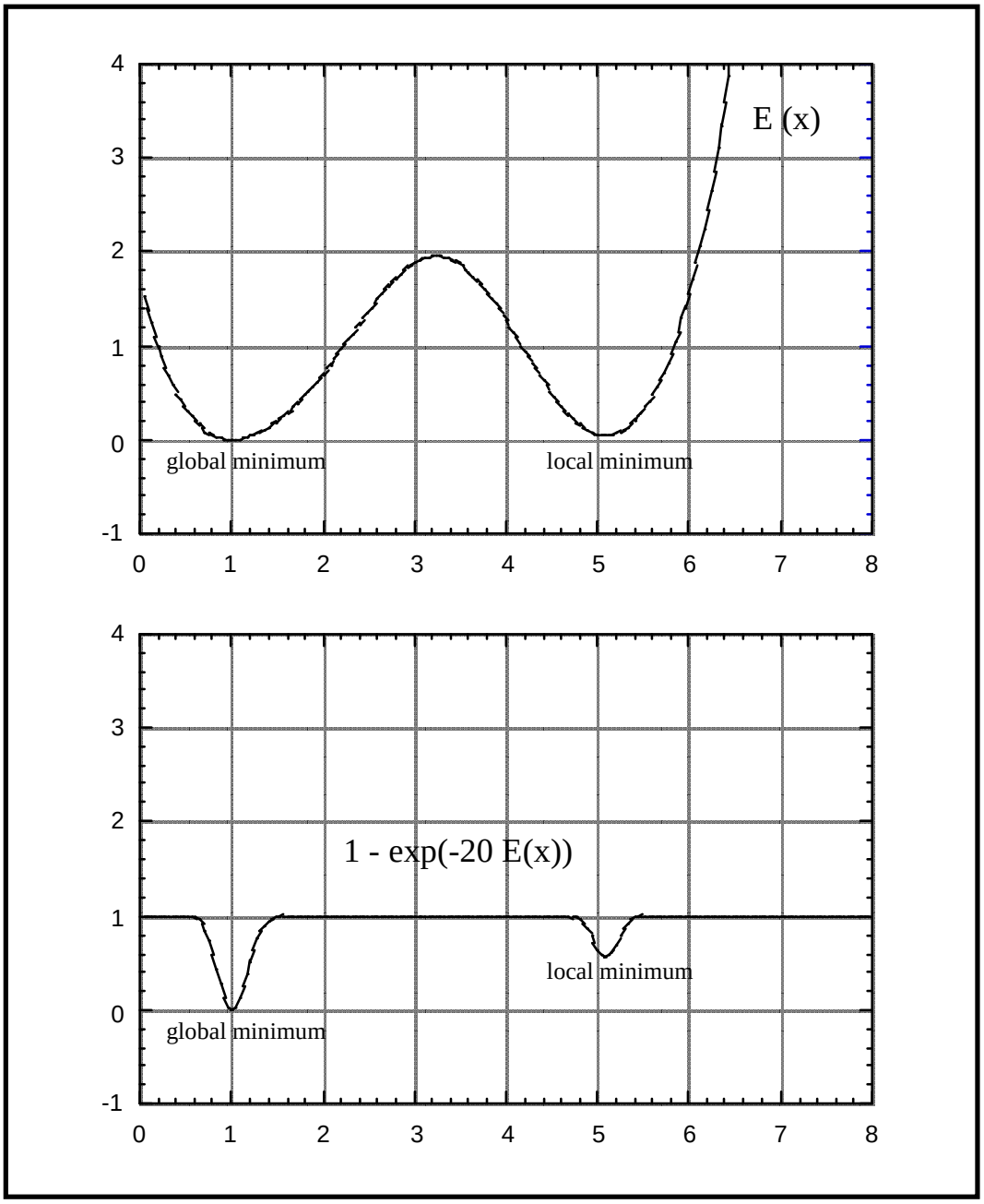

Figure 2: Example of an objective function and an exponential distortion of it.

MARKOV CHAINS AND SIMULATED ANNEALING

A Markov chain is a sequence of random variables that satisfies the following condition:

$$P\left(x_n \middle| x_0, x_1, \ldots x_{n-1}\right) = P\left(x_n \middle| x_{n-1}\right) \tag{11}$$

where $P\left(x_n \middle| x_0, x_1, \ldots x_{n-1}\right)$ is the conditional probability for x_n to occur as the nth element in the sequence, given the sequence's evolution $x_0, x_1, \ldots x_{n-1}$; and $P\left(x_n \middle| x_{n-1}\right)$ represents the conditional probability of x_n, given the previous occurrence of x_{n-1} in the sequence.

Condition (11) is called the Markov condition and it states that the probability of the next outcome or state in the sequence depends only on the present state, and it is totally independent of the previous evolution of the chain.

The transition matrix **P** of a Markov chain is defined by an stochastic square matrix whose entries are given by:

$$p_{ij} = P\left(x_n = j \middle| x_{n-1} = i\right) \tag{12}$$

where p_{ij} represents the probability of going from state i to state j. When p_{ij} is independent of n for all possible states i and j, the Markov chain is said to be homogeneous. On the other hand, if the transition probabilities change with n, the chain is said to be non-homogeneous [6].

As it was explained before, in the simulated annealing algorithm, new models are computed according to a given perturbation scheme and are accepted or discarded according to an acceptance criterion. In this process, the occurrence of a proposed new model, as the next model in the sequence, depends on the current one but is totally independent of the previous ones. As it can be seen now, the evolution of models generated by the simulated annealing algorithm exhibits the Markov property (11) and constitutes, indeed, a Markov chain. However, it is a non-homogeneous chain. This is because the acceptance probabilities depends on the temperature parameter, which is constantly varied during the evolution of the algorithm.

An important fact is that the theory of Markov chains provides excellent means for studding the properties of simulated annealing algorithms. In [3], it is proved that for a initial temperature

large enough and α=1, the logarithmic cooling schedule presented in (5) provides a sufficient condition for asymptotic convergence to the global minimum of the energy function.

A SIMPLE BUT ILLUSTRATIVE EXAMPLE

The present section illustrates with a simple example the most important properties of the simulated annealing searching technique. Two functions are considered in the example:

$$f_1(x) = \frac{1}{4}(x-3)^2 \tag{13}$$

$$f_2(x) = \frac{1}{2}\ \mathrm{Cos}\left(\frac{8\pi}{19}(x-1)\right) \tag{14}$$

where x = ln(σ) represents an unidimensional model space, and the region of study is going to be limited to -10.0 $\leq$ x $\leq$ 10.0 (4.5x10^-5 $\leq \sigma \leq$ 2.2x10^4).

The inversion data set is defined as:

$$\overline{m}^T = [m_1\ m_2] = [f_1(1)\ \ f_2(1)] = [1\ \ 1/2] \tag{15}$$

And the energy function or objective function is given by:

$$E(x) = (f_1(x) - f_1(1))^2 + (f_2(x) - f_2(1))^2 \tag{16}$$

which is the same objective function presented in Figure 2. As it can be seen from Figure 2, the global minimum of the energy function is located al x = 1 (σ =2.72), and it also exhibits a local minimum at x = 5.1 (σ =164).

Figure 3 illustrates the evolution of a properly tuned simulation annealing search. For this simulation, a Cauchy distribution with a = 0.1 was used as perturbation scheme; the Metropolis' criterion described before was used as acceptance criterion; and a geometric decay with α = 0.9 and initial temperature of 1000 was used as cooling schedule. No distortion function and no normalization of the error terms were used. The maximum number of iterations allowed was restricted to 300.

As it can be seen from Figure 3, during the first 50 iterations, the algorithm performs an almost pure random evaluation of the model space. Between iterations 60 and 70, a preference for two specific regions, corresponding to the global and local minima, starts to appear more clearly. Finally, the algorithm stops at iteration 140 giving a final value of x = 1.01 (σ =2.73).

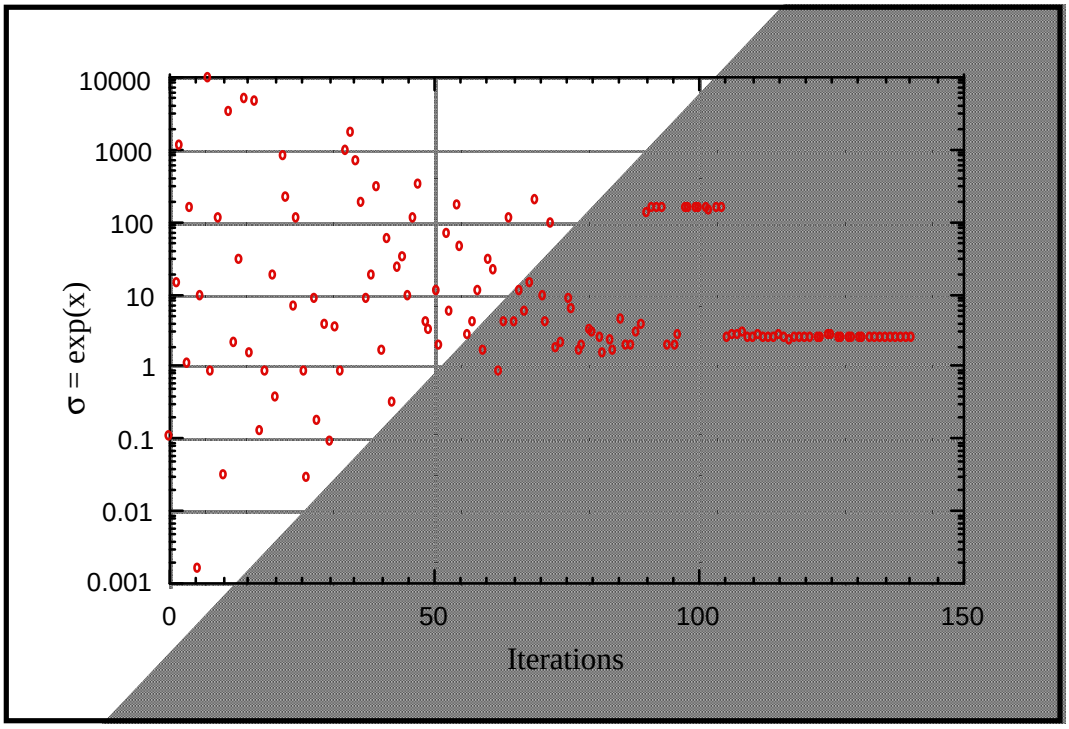

Figure 3: Evolution of a properly tuned simulated annealing algorithm.

The example presented in Figure 3 is of special importance because it also illustrates the ability of simulated annealing for escaping from local minima. Notice how, around the 100th iteration, the algorithm seems to get trapped in the local minimum at x = 5.1 (σ =164). However, it manages to escape and finds the correct solution. Such a situation also reveals the importance of an adequate stopping criterion. The use of a poor stopping criterion may result in termination of the search before the algorithm succeeds in escaping from a local minimum.

Figures 4 and 5 instead, show how a bad tuning of the simulated annealing algorithm can lead to an unsuccessful search. In fact, one of the most practical difficulties presented by this type of searching method is that the proper selection of the algorithm parameters is not always obvious. In most of the cases the proficiency of the parameter values is related to the specific properties of the energy function and other characteristics of the particular problem under consideration.

In particular, Figure 4 illustrates what happens when the cooling is performed too slowly. This experiment is the same of Figure 3 but a value of $\alpha = 0.99$ was used instead of 0.9. Notice how the maximum number of iterations is reached and the algorithm is still moving randomly through the model space.

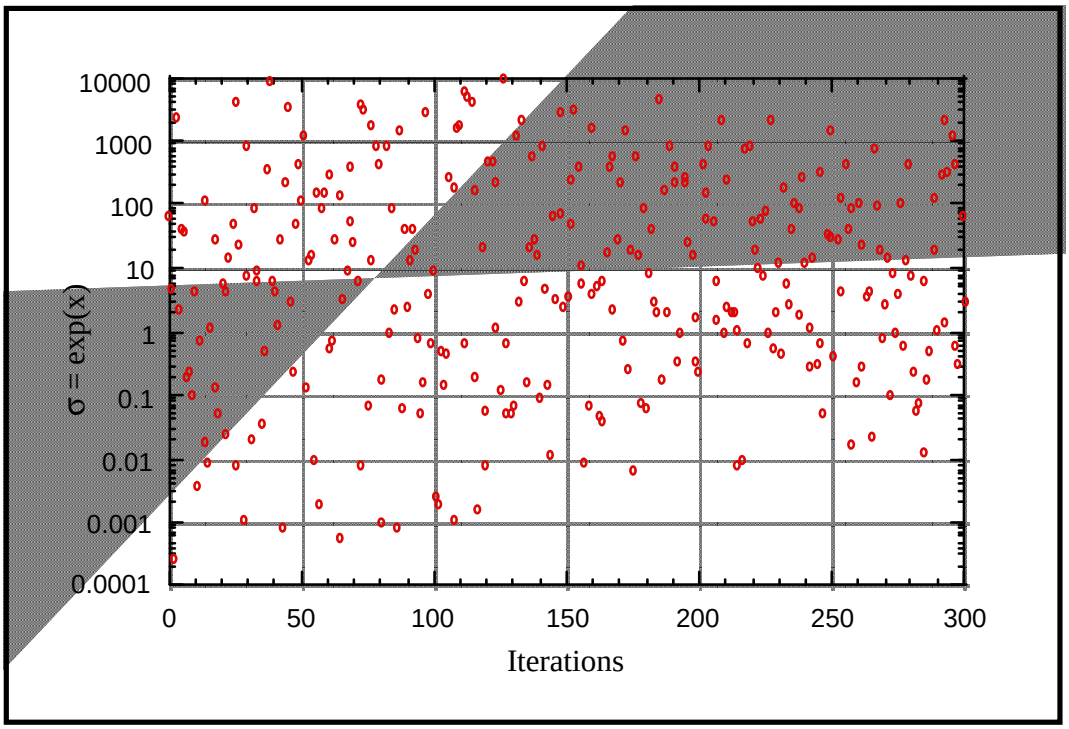

Figure 4: Simulated annealing with $\alpha = 0.99$.

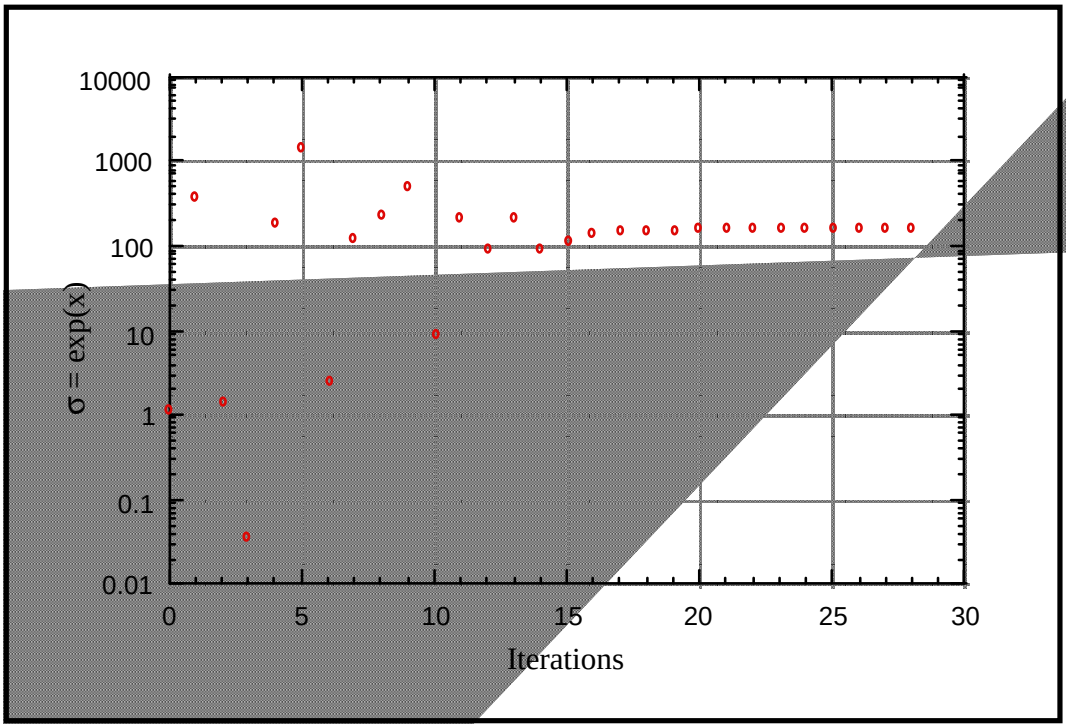

Figure 5: Simulated annealing with α = 0.50.

On the other hand, Figure 5 illustrates what happens when the cooling is performed too fast. Again, the experiment is the same of Figure 3 but a value of α = 0.5 was used instead of 0.9. Notice how the algorithm does not perform a proper global search and, in this particular running, stops prematurely after getting stuck in the local minimum of the energy function.

CONCLUSIONS

Simulated annealing constitutes a global searching technique that presents important advantages over other conventional searching methods. Among its most important properties is its capacity for escaping from local minima. However, its intensive computational requirements and the practical difficulties involved in the proper choice of its parameters are factors that reduce its potentiality in many cases.

In the particular case of the time harmonic field electric logging problem, in which function evaluations are computationally expensive, the feasibility of simulated annealing depends basically on the available amount of time. For this reason, hybrid searching schemes are proven to be the most viable procedure [7]. In this kind of schemes, simulated annealing may be used to roughly approximate a good solution that will be later adjusted by a local search algorithm.

REFERENCES

[1] van Laarhoven, P.; Aarts, E. (1988), Simulated annealing: Theory and Applications., Kluwer Academic Publishers.

[2] Kirkpatrick, S.; Gelatt, C.; Vecchi, M. (1983), Optimization by Simulated Annealing., Science 220, 671-680.

[3] Geman, S.; Geman, D. (1984), Stochastic Relaxation, Gibbs Distribution, Bayesian Restauration of Images., IEEE Transactions Pattern Anal. Mech. Intell. 6, 721-741.

[4] Update Report #13: Gradient Methods.

[5] Azencott, R. (1992), Simulated Annealing: Parallelization Techniques, John Wiley & Sons, Inc.

[6] Grimmett, G.; Stirzaker, D. (1994), Probability and Random Processes, Oxford Science Publications.

[7] Chunduru, R. (1996), Global and Hybrid Optimization in Geophysical Inversion. University of Texas at Austin.