On Step Width Adaptation in Simulated Annealing for Continuous Parameter Optimisation

Lars Nolle[1], Alec Goodyear[1], Adrian A Hopgood[2] , Phil D Picton[3], and Nicholas StJ Braithwaite[1]

[1]Oxford Research Unit, The Open University, Berkley Road, Oxford, OX1 5HR, UK
{L.Nolle, A.Goodyear, N.S.Braithwaite}@open.ac.uk
[2]Department of Computing, The Nottingham Trent University, Burton Street, Nottingham, NG1 4BU, UK
Adrian.Hopgood@ntu.ac.uk
[3]School of Technology and Design, University College Northampton, Northampton, NN2 6JD, UK
Phil.Picton@northampton.ac.uk

Abstract. Simulated annealing is a general optimisation algorithm, based on hill-climbing. As in hill-climbing, new candidate solutions are selected from the 'neighbourhood' of the current solution. For continuous parameter optimisation, it is practically impossible to choose direct neighbours, because of the vast number of points in the search space. In this case, it is necessary to choose new candidate solutions from a wider neighbourhood, i.e. from some distance of the current solution, for performance reasons. The right choice of this distance is often crucial for the success of the algorithm, especially in real-world application where the number of fitness evaluations is limited. This paper explains how in such a case the use of a variable radius of this neighbourhood, refereed to as maximum step width, can increase the over-all performance of simulated annealing. A real-world example demonstrates the increased performance of the modified algorithm.

1 Introduction

Simulated annealing (SA) is a robust general optimisation method that was first introduced by Kirkpatrick et. al. [1], based on the work of Metropolis et. al. [2]. It simulates the annealing of a metal, in which the metal is heated-up to a temperature near its melting point and then slowly cooled to allow the particles to move towards an optimum energy state. This results in a more uniform crystalline structure and so the process allows some control over the microstructure. SA has been demonstrated to be robust and capable of dealing with noisy and incomplete real-world data [3,4].

Simulated annealing is a variation of the hill-climbing algorithm. Both start off from a randomly selected point within the search space. Unlike in hill climbing, if the fitness of a new candidate solution is less than the fitness of the current solution, the new candidate solution is not automatically rejected. Instead it becomes the current solution with a certain transition probability $p(T)$. This transition probability depends on the difference in fitness ΔE and the temperature T. Here, 'temperature' is an abstract control parameter for the algorithm rather than a real physical measure.

B. Reusch (Ed.): Fuzzy Days 2001, LNCS 2206, pp. 589–598, 2001.

Equation 1 gives a common transition function p*(T)* for a given temperature and a given difference in fitness:

$$p(T) = \frac{1}{1 + e^{\frac{\Delta E}{T}}} \tag{1}$$

Where:

T	temperature – control parameter for cooling schedule
p*(T)*	probability of transition for temperature T
ΔE	difference between previous candidate solution and current one

The algorithm starts with a high temperature, which is subsequently reduced slowly, usually in steps. Equation 2 shows the standard cooling function introduced by Kirkpatrick [1]. Many others can be found in the literature [5,6].

$$T_{n+1} = \alpha T_n \tag{2}$$

Where:

T_n	temperature at step n
α	cooling coefficient ($\alpha < 1$)

On each step, the temperature must be held constant for an appropriate period of time (i.e. number of iterations) in order to allow the algorithm to settle in a 'thermal equilibrium', i.e. in a balanced state. If this time is too short, the algorithm is likely to converge to a local minimum. The combination of temperature steps and cooling times is known as the annealing schedule, which is usually selected empirically. Fig. 1 shows the flowchart of the basic simulated annealing algorithm.

Fig. 2 shows a typical run of a SA algorithm, optimising a real-world system [4]. The dotted line represents the best solution found so far, the thick line represents the fitness on each iteration. It can be observed that the algorithm initially explores the search space (large differences in fitness), while later on (after approximately 400 iterations in this case) it exploits the most promising region.

Hence, the behaviour of the algorithm at the beginning of the search is similar to a random walk, while towards the end it performs like ordinary hill climbing.

2 Need for step width variation

For continuous parameter optimisation, it is practically impossible to choose direct neighbours of the current solution as new candidate solutions, simply because of the vast number of points in the search space. In this case, it is necessary to choose new candidate solutions from some distance in a random direction of the current solution in order to travel in an acceptable time through the search space. This distance could either be a fixed step width s or it could have an upper limit s_{max}. In the first case, the neighbourhood would be defined as the surface of a hypersphere around the current solution, in the second case the neighbourhood would be defined as the volume of the hypersphere.

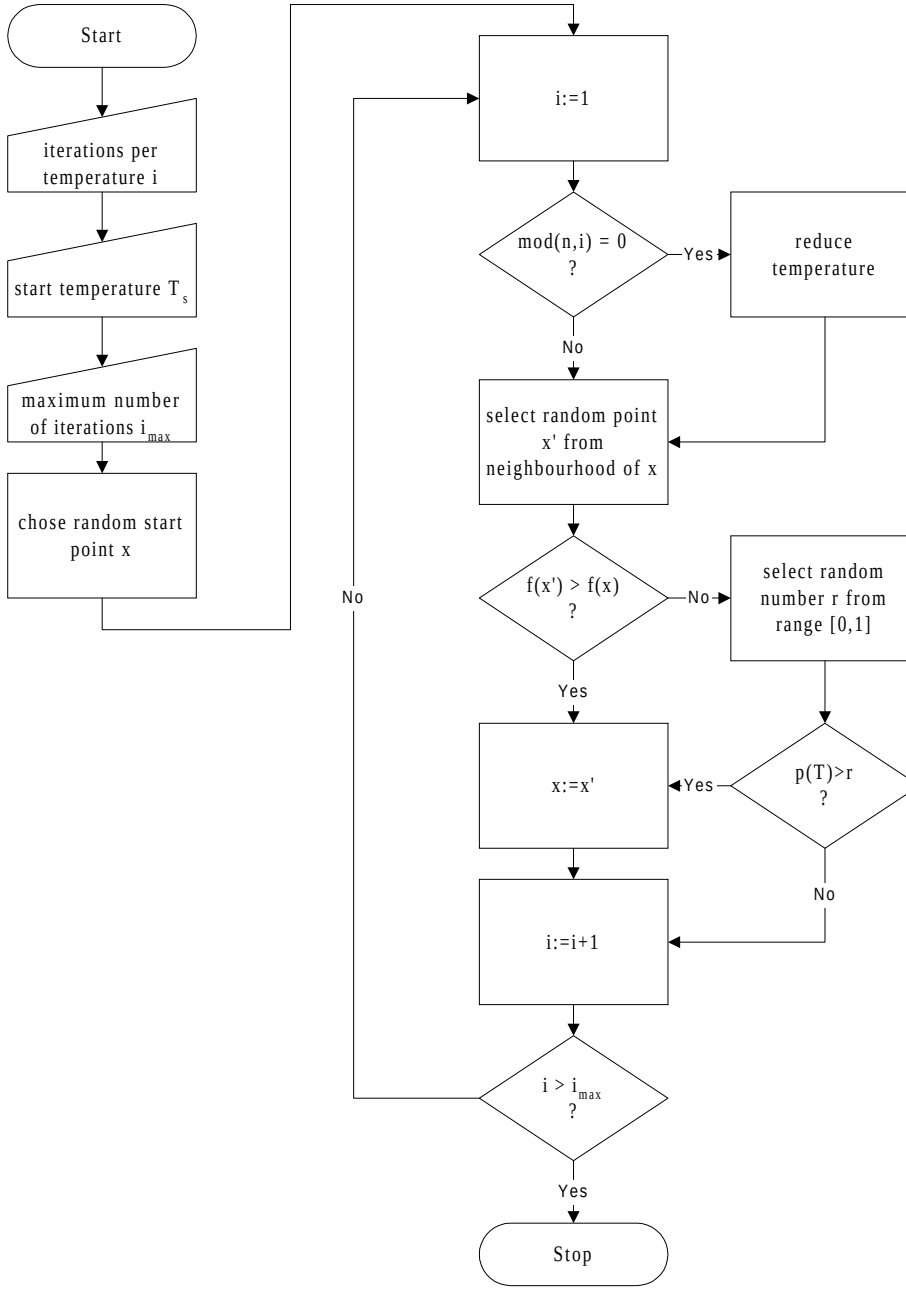

Fig. 1. Flowchart of the basic SA algorithm

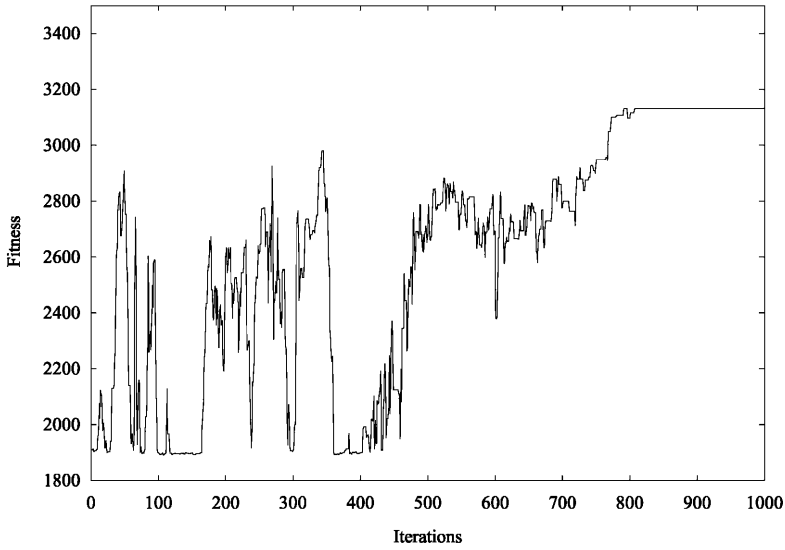

Fig. 2. Typical run of a SA algorithm; at the beginning of the search the algorithm explores the search space, while it is exploiting the most promising region of the search space after the algorithm has been 'frozen'

In the latter case, new candidate solutions might be generated by adding small, equally distributed random numbers from the interval $[-s_{max}, s_{max}]$ to each component of the current solution vector. The maximum step width s_{max} is crucial to the success of SA. If s_{max} is chosen to be too small and the start point for a search run is too far away from the global optimum, the algorithm might not be able to get near that optimum before the algorithm 'freezes', i.e. the temperature becomes so small that $p(T)$ is virtually zero and the algorithm starts to perform only hill climbing. In that case, it will get stuck in the nearest local optimum. If, on the other hand, the step width has been chosen to be too large, and the peak of the optimum is very narrow, the algorithm might well get near the global optimum before the algorithm 'freezes', but never reaches the top, because most of the steps are simply too large so that new candidate solutions 'fall off' the peak. Hence, there is always a trade-off between accuracy and robustness in selecting an appropriate maximum step width. If s_{max} is too small, SA has the potential to reach the peak of the 'frozen-in' optimum, but it cannot be guaranteed that this optimum is the global one. On the other hand, if s_{max} is too large, SA has the potential to get near the global optimum, but it might never reach the top of it.

A solution could be to use small steps and to adjust the cooling schedule to increase the length of the Markov chains. This is not always be possible in real-world optimisation problems with time constrains, i.e. a limited number of possible fitness evaluations. The solution proposed here is to adapt the maximum step width s_{max} to the search process itself. In general, there are two possible approaches. Firstly, s_{max} may be adapted to an SA parameter, like current iteration, temperature, etc. The search starts with a large value for s_{max} and is subsequently decreased towards the end. Secondly, s_{max} can be adapted to the fitness landscape, e.g. ΔE or $p(T)$.

3 Real-world example

In this example, SA has been employed to deduce fourteen Fourier terms in a radio-frequency (RF) waveform for a Langmuir probe compensation system [4].

Langmuir probes [7] are important electrostatic diagnostics for RF-driven gas discharge plasmas [8]. These plasmas are inherently non-linear, and many harmonics of the fundamental are generated in the plasma.

RF components across the probe sheath distort the measurements made by the probes. To improve the accuracy of the measurements, these RF components must be removed. This has been achieved by active compensation, i.e. by applying an RF signal to the probe tip. Not only amplitude and phase of the applied signal have to match that of the exciting RF, also its waveform has to match that of the harmonics generated in the plasma. The active compensation system uses seven harmonics to generate the required waveform. Therefore, fourteen heavily interacting parameters (seven amplitudes and seven phases) need to be tuned before measurements can be taken. Fig. 3 shows the closed optimisation loop.

The goal is to maximise the parameter 'floating potential' of the Langmuir probe system, which acts as a fitness measure. Previous work has shown that this is a hard optimisation problem, because the inputs heavily influence each other. Also, the fitness at the global optimum is not known a priori.

Fig. 4 shows the set-up of the experiment and Fig. 5 shows the Langmuir probe inserted into an Argon plasma.

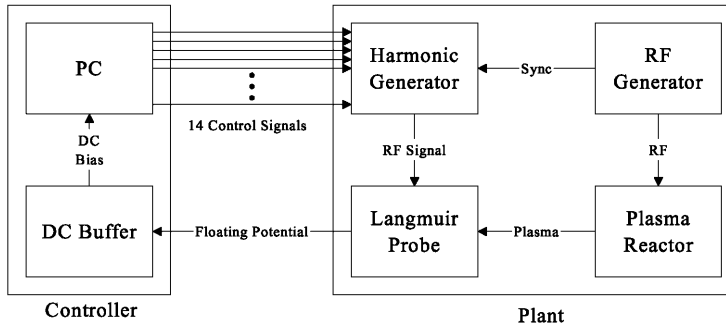

Fig. 3. Closed optimisation loop

Fig. 4. Experimental set-up

Fig. 5. Langmuir probe in Argon plasma

Each of the digital inputs and the digital output of the system to be optimised had a resolution of 12 bits, hence the search space consisted of approximately 3.7×10^{50} search points.

During the experiments, it became evident that the selection of s_{max} was crucial to the solutions found by SA. Fig. 6 shows a typical run with a small maximum step width (s_{max}=50). It can be observed that the algorithm has difficulties in getting out of the 'flat' region of the search space from where it started. Only after approx. 800 iterations did it reach the next optimum, where it performed hill climbing.

In Fig. 7, a typical run with a large step width can be seen (s_{max}=4000). While the algorithm seems to find the region of the global optimum easily, it failed to exploit this region.

Even with a carefully selected value for s_{max}, it was not possible to ensure stable and reproducible solutions. While most of the search runs resulted in fitness values of around 2800, in a few runs values of around 2900 have been found. In order to overcome this behaviour, a scaling function has been empirically designed to adapt the maximum step width to the current iteration (equation 3).

$$s_{\max}(n)=\frac{2s_0}{1+e^{\frac{\beta n}{n_{\max}}}} \tag{3}$$

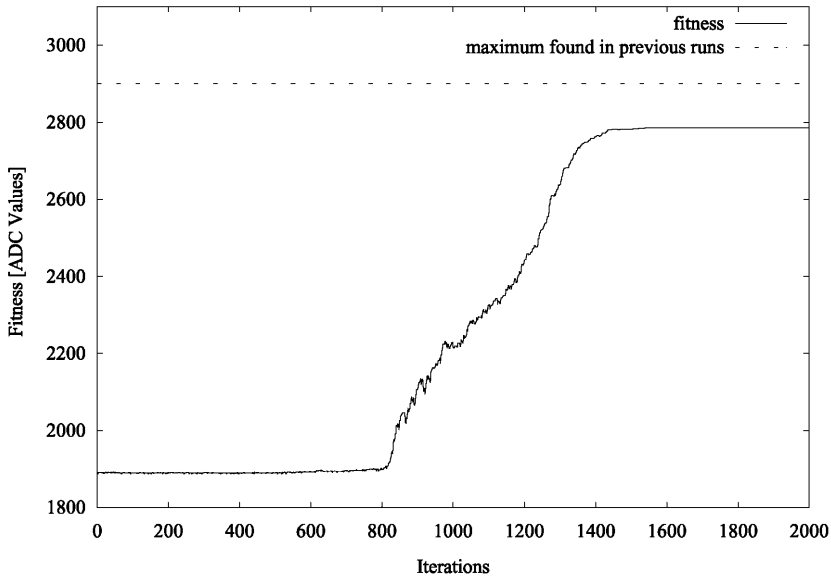

Fig. 6. The step width has been chosen too small; the algorithm failed to reach the region containing the maximum peak during the exploration phase of the search

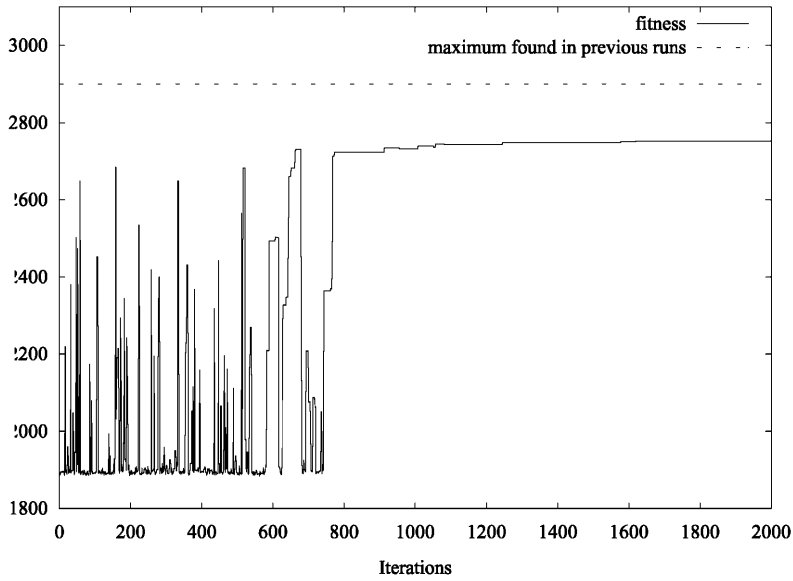

Fig. 7. The maximum step width has been chosen too large; the algorithm failed to find the maximum peak during the exploitation phase of the search run

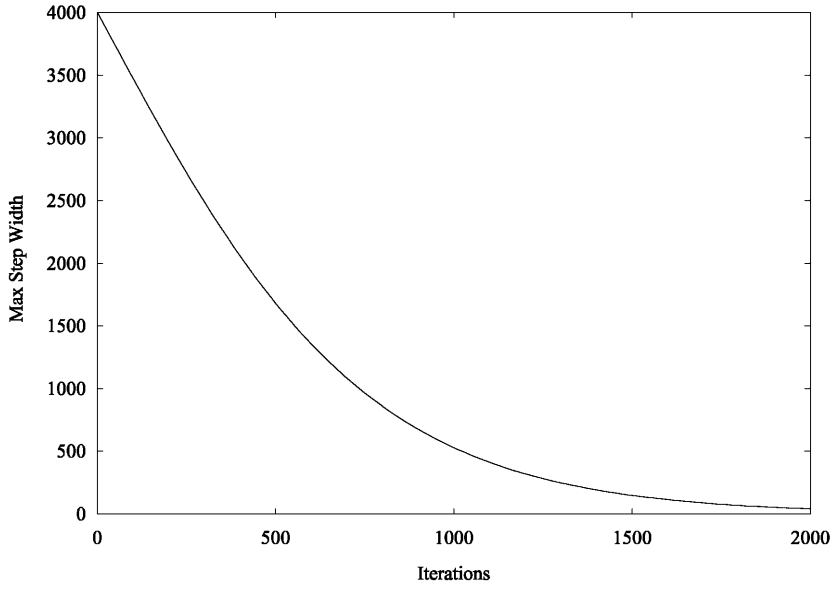

Fig. 8. Step width scaling function used in this research

Where:

$s_{max}(n)$	maximum step width at iteration n
s_0	initial maximum step width
n	current iteration
n_{max}	maximum number of iterations
β	adaptation constant; a value of 5.3 was chosen

Fig. 8 shows the graphical representation of the scaling function while Fig. 9 shows a typical run with the improved SA.

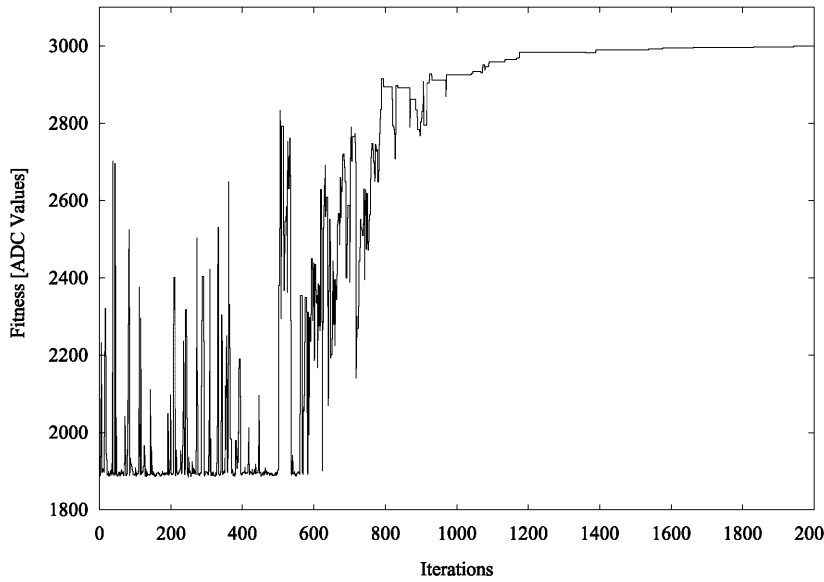

Fig. 9. Search run with step width adaptation; the algorithm explores the search space at the beginning of the search and transforms smoothly into hill-climbing

For evaluation, three different methods have been compared: method 1 used a static maximum step width of 50, method 2 used a static maximum step width of 4000 and finally, method 3 used the adaptive step width with an initial value of 4000. Table 1 shows the results of the experiments.

Table 1. Average Fitness and deviation of fitness for methods used during this research

Method No	Average Fitness [Volts]	Deviation of Fitness [Volts]
1	12.20	1.82
2	11.93	1.05
3	17.84	0.44

Fig. 10 gives a graphical representation, where the dots represent the average fitness gained and the error bars represent the standard deviation for the results.

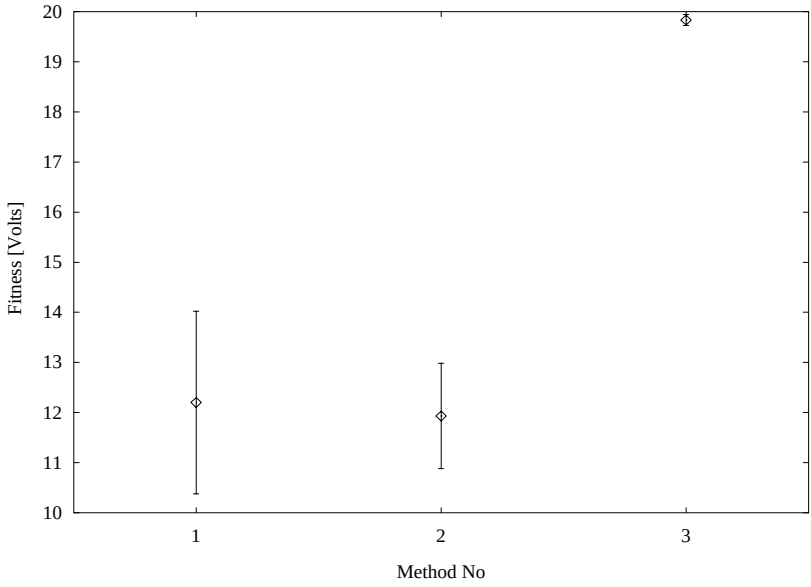

Fig. 10. Comparison of methods; method 1 is a standard SA algorithm with small maximum step width, method 2 is a standard SA algorithm with large maximum step width, and method 3 is a SA algorithm with step width adaptation

It can be observed that not only can the average fitness be improved significantly, but also the robustness, i.e. the reproducibility, has been improved dramatically (smaller error bars). That means the algorithm is more likely to reach the peak of the global optimum. This makes it suitable for industrial applications, which usually require a high degree of robustness and reproducibility.

4 Conclusions

For continuous parameter optimisation, selecting an appropriate maximum step width s_{max} for simulated annealing is always a compromise between accuracy and robustness. If it is too small, SA has good exploitation capabilities but reduced chances of reaching the global optimum to exploit it. If s_{max} is too large, the algorithm has good exploration capabilities, and hence is likely to find the region of the search space containing the global optimum, but it then performs badly in the exploitation of this region.

A solution to this problem is the on-line adaptation of the maximum step width to either a SA parameter, like temperature or iteration, or to the fitness landscape. In this research, the performance of SA in the optimisation of a Langmuir probe system has been dramatically improved by adapting the maximum step width to the current iteration. This has not only improved the average fitness for the system, it has also improved the reproducibility of the results, which is of great importance for industrial applications.

5 Acknowledgements

This work was funded by the Engineering and Physical Sciences Research Council (EPSRC) under grant reference GR/M71039/01.

6 References

1. Kirkpatrick, S., Gelatt Jr, C. D., Vecchi, M. P.: “Optimization by Simulated Annealing”, Science, 13 May 1983, Vol. 220, No. 4598, pp 671-680
2. Metropolis, A., Rosenbluth, W., Rosenbluth, M. N., Teller, H., Teller, E.: “Equation of State Calculations by Fast Computing Machines”, The Journal of Chemical Physics, Vol. 21, No. 6, June 1953, pp 1087-1092
3. Nolle, L., Walters, M., Armstrong, D.A., Ware, J.A.: “Optimum Finisher Mill Set-up in the Hot Rolling of Wide Steel Strip Using Simulated Annealing”, Proceedings Of the 5th International Mendel Conference, Brno, 1999, pp 100-105
4. Nolle, L., Goodyear, A., Hopgood, A.A., Picton, P.D., Braithwaite N.StJ.: “Automated Control of an Actively Compensated Langmuir Probe System Using Simulated Annealing”, submitted for publication at The 5th World Multi-Conference on Systemics, Cybernetics and Informatics, SCI 2001, Orlando, Florida USA, July 22-25, 2001
5. Geman, S., Geman, D.: “Stochastic Relaxation, Gibbs distribution and the Bayesian restoration of images”, IEEE Trans. PAMI, 5 (1984), pp 721-741
6. Huang, M.D., Romeo, F., Sangiovanni-Vincentalli, A.: “An efficient general cooling schedule for simulated annealing”, Proceedings of the IEEE International Conference on Computer Aided Design, Santa Clara, 1986, pp 381-384
7. Swift, J. D. S., Schwar, M. J. R.: “Electrical Probes for Plasma Diagnostics”, Ilitte, London, 1970
8. Lieberman, M. A., Lichtenberg, A. J.: “Principles of Plasma Discharges for Materials Processing”, Wiley Interscience, New York, 1994