Proceedings of the 1995 Winter Simulation Conference
ed. C. Alexopoulos, K. Kang, W. R. Lilegdon, and D. Goldsman

SIMULATED ANNEALING: PAST, PRESENT, AND FUTURE

Mark Fleischer

Department of Engineering Management
Old Dominion University
Norfolk, VA 23529-0248, U.S.A.

ABSTRACT

This paper provides an overview of the simulated annealing algorithm and describes its historical foundation in thermodynamics as well the genesis and evolution for solving difficult optimization problems. An example illustrating its application to graph problems is provided as well as a look at ongoing, state-of-the-art research using SA.

1 INTRODUCTION

Few developments in the field of optimization have generated as much enthusiasm and criticism or spawned as many practioners and detractors as the invention of the simulated annealing (SA) algorithm. This paper describes the genesis and evolution of this remarkable optimization technique and how this computer simulation of a thermodynamic system can be used to solve many types of optimization problems.

Along with a few other types of generalized optimization schemes, SA is considered a *meta-heuristic*. Its generality and applicability stems from its foundation in thermodynamics and statistical mechanics. Thus, it can be used to solve many combinatorial optimization problems (COPs) and some continuous variable problems (Bonomi and Lutton 1984). The SA algorithm searches the set of solutions, referred to as a *configuration space* (Tovey 1988), in much the same manner as a thermodynamic system changes from one energy state to another. The development of SA therefore established a mathematical analogy between optimization problems and thermodynamic systems thus creating a new foundation from which to analyze and solve such problems. It has over the years been extensively studied with varying degrees of enthusiasm, criticism, and a great deal of experimentation (see *e.g.*, Bonomi and Lutton 1984, Eglese 1990).

2 WHAT IS BEING SIMULATED?

The genesis of SA comes from attempts to simulate the effects of a heat bath on various chemical substances. This was accomplished by Metropolis, *et al.* (1953) who established criteria to simulate how thermodynamic systems change from one energy level to another. These criteria, based on the laws of thermodynamics, require that a system of particles exhibit energy levels in a manner that maximizes the thermodynamic entropy at a given temperature value. At the same time, the *average* energy level must be proportional to a constant, the temperature. This average is based on some ensemble of energy states any one of which the system may be in at any particular moment.

This ensemble of energy states implies that the system "visits" such energy states spontaneously, but subject to the constraints mentioned above. Bonomi and Lutton (1984) state these requirements as a simple nonlinear math program in which the entropy of the state vector is maximized subject to the constraints that the expected kinetic energy is proportional to a fixed contant, the temperature and that the total probability of system states equals 1. This simple math program leads to requirements on how the system changes from one energy level to another. In simulation parlance, this is referred to as the *Metropolis Acceptance Criterion* which defines the probability that some candidate energy level is *accepted* as the current energy level in the next time increment and is given by the following expression:

$$\Pr\{\text{Accept } \Delta E\} = \begin{cases} e^{-\Delta E/t} & \Delta E > 0 \\ 1 & \Delta E \leq 0 \end{cases} \qquad (1)$$

The steady-state probabilities $\pi_i(t)$ for some state i is given by

$$\pi_i(t) = \frac{e^{-E_i/t}}{B(t)} \qquad (2)$$

where $B(t)$ is the well-known Boltzmann partition

function of statistical mechanics and E_i is the energy level associated with some energy partition (a well-defined range of energy values) denoted by state i (Metropolis, *et al.* 1953). A simulation using this acceptance probability therefore obeys the laws of statistical mechanics in a discrete manner and therefore permits simulation studies of materials in a heat bath.

3 THE ROAD TO OPTIMIZATION

It was only a matter of time till analogies were made between thermodynamic systems and combinatorial optimization problems. Both deal with large ensembles—in the former case, there are large numbers of particles; in the latter case, there are large numbers of objective function values. But the similarity in complexity and size was eclipsed in the early 80's. At that time Kirkpatrick, *et al.* (1983) and, independently, Cĕrny (1985) reasoned that the Metropolis Acceptance Criterion could be used in a simulation of the annealing process. It is well known in the field of metallurgy, for example, that annealing a substance—slowly cooling a material—can relieve stresses and aid in the formation of a perfect crystal lattice. By allowing particles to move about, but with gradually decreasing kinetic energy, conditions tend to prevail that lead to such perfect crystal lattices *and low energy system states.* They realized that an analogy between energy state values and objective function values in optimization problems was possible.

Once the analogy was made, it was easy to see some advantages in using SA for solving such COPs. Using the Metropolis Acceptance Criterion, transitions from one solution state to another can be defined using the following expression.

$$p_{ij}(t_k) = \begin{cases} G_{ji}e^{-\Delta f_{ji}^{+}/t_k} & \Delta f_{ji}^{+} > 0 \\ 1 & \Delta f_{ji}^{+} \leq 0 \end{cases} \tag{3}$$

where G_{ji} is the probability of generating candidate j given the current state i (this means that $j \in N(i)$, or j is in the *neighborhood* of i) and $\Delta f_{ji}^{+} \equiv \max\{0, f_j - f_i\}$.

This transition probability implies that *uphill* moves are possible with a probability inversely related to the height of the hill and decreasing with decreasing temperature. This aspect of SA is one of its most attractive features—it can avoid becoming trapped in local optima and, hence, has a greater chance of finding the global optima. Indeed, this is what annealing is all about—cooling the system slowly enough so that the minimum energy state or minimum objective function value can be realized.

Because the changes from one state to another can be expressed as a transition probability, it was possible to model the annealing process as a Markov chain. This naturally lead researchers to investigate its mathematical properties. It was clear early on that the stationary distribution converged to the optimum state vector (Aarts and Korst 1989). If $\pi(t)$ is the stationary distribution at temperature t and $\mathbf{q}^*$ is the optimum state vector where the total probability of optimal states is equal to 1, then the i^{th} element of the vector is very similar to (2) where the objective function value f_i is merely substituted for the energy level and $N(t)$ is a normalization coefficient analogous to the Boltzmann partition function (Aarts and Korst 1989).

$$\pi_i(t) = \frac{e^{-f_i/t}}{N(t)} \tag{4}$$

It is easy to see that

$$\lim_{t \to 0} \pi(t) = \mathbf{q}^* \tag{5}$$

(Aarts and Korst 1989). From a mathematical point of view, this requires running SA at a fixed temperature for an infinite number of iterations—not very practical to say the least.

The question therefore arose as to whether it was possible to lower the temperature *at each iteration* and still converge in probability to the optimum state vector. Since the talisman of annealing is lowering the temperature, *how fast* the temperature is lowered became a rich area of investigation. If the temperature is lowered too quickly, we end up *quenching* the system. This corresponds to finding a local optimum, not a global optimum. But if the temperature is lowered at each iteration, the state distribution must be modeled as an *inhomogeneous* Markov chain.

Mitra, *et al.* (1986) and Hajek (1988) determined various formulations (using different approaches) for such *cooling schedules* which yield an inhomogeneous Markov chain that is *strongly ergodic* and therefore converges in *distribution* as well as probability using the form

$$t_k = \frac{\gamma}{\log(c + k)} \tag{6}$$

where t_k is the temperature at time index k. Thus, if $\mathbf{v}^{[k]}$ is a state probability vector at time index k, then by using (6),

$$\lim_{k \to \infty} \mathbf{v}^{[k]} = \mathbf{q}^* \tag{7}$$

These mathematical aspects, its hill-climbing capability, convergence in probability and distribution, and the ease of implementation (described below)

gave rise to a great deal of enthusiasm for the algorithm (Eglese 1990). Yet, there was also the disquieting fact that SA, in its generic form, is a "dumb" optimization scheme; it is blind to the global optimum and can find it only to leave it in the next iteration searching the entire configuration space in total ignorance of the quality of the solution it left. This lead to many criticisms of the method and a great deal of research designed to highlight its limitations (see *e.g.,* Johnson, *et al.,* 1989).

Enhancements to the generic implementation are, however, often obvious and easy enough to implement. The most popular method is to keep track of the best solution obtained up to the current iteration. On the other hand, in *iterated descent* one can perform repeated local searches with random starting solutions often with similar results (Dowsland 1993). It therefore seemed that the convergence properties of SA were attractive only from a scientific and mathematical point of view, and that its practical use as an optimization scheme is often overstated. Indeed, this may explain why SA has so many variations, promoters, and detractors (Eglese 1990). There is, however, ongoing research into how to make SA more effective as described below.

Thus, with the mathematical justifications for SA complete, many researchers explored the application of SA on many different types of problems. This revealed several important implementation issues that are addressed in the following section.

4 IMPLEMENTATION ISSUES

In addition to the cooling schedule, several implementation issues must be addressed before any reasonable SA algorithm can be developed for a specific problem. The following hightlight the more salient implementation issues all of which must have some clear articulation.

- An appropriate cooling schedule.
- A suitable objective function.
- A well-defined neighborhood structure.

Insofar as the cooling schedule is concerned, many implementations use finite-time schedules based on an initial temperature and a final temperature (Fleischer 1993). Other schedules that approximate (6), but are easier and faster to calculate have been used (see *e.g.,* Johnson *et al.* 1989). Still others attempt to *adapt* the cooling schedule to the type of problem at hand by modifying the gross structure of the cooling schedule (Ingber 1993).

Central to SA is the definition of a suitable objective function. This will depend on the type of problem at hand. For COPs, it can often be calculated in conjunction with the selection of candidate solutions. For continous variable problems, however, this can be the slowest aspect of SA (Bohachevsky 1986).

Many COPs, for example naturally lend themselves to a reasonable neighborhood structure once an objective function is clearly defined (see the example below). These structures must make it possible for SA to explore *all* solutions (even if most are never visited) and at the same time permit efficient calculation of Δf_{ji}^{+}.

Indeed, the particular neighborhood structure used and the problem instance itself have implications for how well SA performs. Many researchers have explored various ways of modifying the neighborhood size used in SA in order to improve its performance. This has lead to differing and, at times, *seemingly* contradictory results. Waterman and Goldstein (1988) applied SA to several instances of the Traveling Salesman Problem (TSP) using different neighborhood sizes. They report that the finite-time performance of SA degrades if the neighborhood size is too large or too small, the implication being that some optimal neighborhood size exists. Cheh, *et al.* (1991) show that for some problems, including the TSP, performance improved for smaller neighborhood sizes. Yao (1991), on the other hand, reports that larger neighborhood sizes improve the performance of SA.

Fleischer (1993) and Fleischer and Jacobson (1994) attempted to resolve these issues by showing that it is possible to model SA as a *Markov information source.* Relying on information theoretic concepts, it became apparent that the neighborhood structure can affect the information rate or level of total uncertainty associated with SA. They showed that the higher the entropy of the associated Markov chain, the better SA tends to perform.

For many practical applications, these implementation issues do not present much difficulty. The following section demonstrates an implementation scheme for the minimum vertex cover problem.

5 AN EXAMPLE OF SA

Perhaps the best way to illustrate how SA can be implemented is to use a simple example. Consider the vertex cover (VC) problem. A vertex cover is a set of nodes such that all arcs in some graph $G(V, E)$ have at least one end impinging on a node in the set. This problem, along with other related problems, concern the determination of how many nodes (and

which ones) constitute a *minimum* vertex cover. Such COPs can be very difficult when the number of nodes becomes large and are, in fact, *NP*-hard (Garey and Johnson 1979) and can be solved using SA. Figure 1 shows such a vertex cover where the shaded nodes correspond to the vertex covering set.

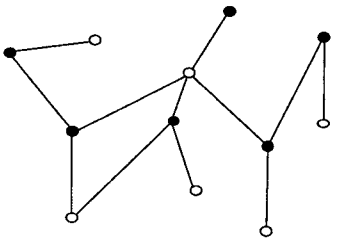

Figure 1: A Vertex Cover

Since solutions to this problem correspond to specific combinations of nodes, a reasonable way of defining the configuration space would be to have *all* combinations of nodes correspond to solutions. The neighborhood structure could therefore be defined by allowing candidate solutions to be obtained by adding or subtracting any one node from the current set. Changing the membership status of any one node creates a new combination of nodes and, hence, a new solution. Thus, if n is the number of nodes in a graph, then the *neighborhood* can be defined as all the new combinations obtainable by merely changing the membership status—flipping—one of the n node's status to be either be in the set or out of the set. This means that each combination has n neighboring solutions.

Thus, in the same instance we have defined the objective function to be some function based on the number of nodes in the current set of nodes and we have defined a neighborhood in which a candidate solution is obtained by flipping any one of n nodes. Athough this makes defining candidate solutions easy, it also permits *all* combinations of nodes to be part of the solution space. This results in creating a solution space S the size of which is given by the following:

$$|S| = \sum_{i=0}^{n} \binom{n}{i} = 2^n \tag{8}$$

There is therefore an apparent tradeoff in simplifying candidate selection at the expense of enlarging the solution space.

This simplified generation technique, however, often ends up generating solutions that do not constitute a vertex cover—combinations of nodes that do not cover every arc. More generally, some solutions may violate the contraints of a given problem. One way around this problem is to use some penalty function for solutions that violate the constraints of the problem. For the VC problem, one objective function formulation suggested by Aarts and Korst (1989), is the following: Let $V' \subseteq V$ be the subset of nodes corresponding to the current solution. Let $E(V')$ be the number of *uncovered* arcs given the nodes in V'. A suitable objective function can be defined as

$$f_{\text{VC}}(V') = |V'| + \lambda E(V') \tag{9}$$

where $\lambda > 1$ serves as a *penalty parameter* (Aarts and Korst 1989). Anecdotal evidence suggests that allowing all combinations of nodes to be part of the configuration space "smoothes out" the landscape of SA making it easier for SA to find the global optima (see *e.g.,* Johnson *et al.* 1989, Fleischer 1993).

This method also leads to a very efficient way of calculating Δf_{ji}^{+}. Aarts and Korst (1989) define the indicator variable

$$I_{V'}(u) = \begin{cases} 1 & \text{if } u \in V' \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

Using this, it is possible to efficiently calculate the change in objective function by

$$\Delta f = \left[I_{V'}(u) - I_{V \setminus V'}(u)\right] \left(\lambda \sum_{\substack{\{u,v\} \in E \\ v \notin V'}} 1 - 1 \right) \tag{11}$$

(Fleischer 1993). The following pseudo-code displays the simplicity and ease of application that is the hallmark of the SA algorithm.

initialize:
    Generate initial set.
    Calc cooling schedule parameters.
anneal: do while $k < k^*$
    1: $k = k + 1$, calc t_k
    2: select random node u.
    and temporarily change its
    membership status.
    3: calc $\Delta f(u)$.
    4: if $\Delta f(u) \leq 0$
    then goto accept
    else generate a
        U(0,1) random variable R
    5: if $R \leq e^{-\Delta f(u)/t_k}$
    then goto accept
    else change the status of u
    back and goto anneal
accept
    6: adopt the change in status
    of node u
    7: goto anneal

As the SA algorithm proceeds and the temperature parameter decreases, the probability of accepting a candidate solution violating the constraints is reduced. Typically, at termination of the algorithm, a solution which does not violate the constraints is produced. Like twinkling stars, the nodes blink in and out of the current set randomly as they are flipped and *probabilistically* settle into an optimal configuration through the annealing process.

6 FUTURE DEVELOPMENTS

6.1 Simulating Realistic Annealing

It would be nice if it were possible to simulate how a human being handles the annealing process in real life and take advantage of the artisan's experience and expertise. Such a simulation could lead to improvements in SA's performance. From glass blowers to metallugists, the *art* of annealing requires some sense and knowledge of how well something is being annealed. Usually, these experts have some notion of when parts of a system need to be reheated and cooled again. This notion has lead some researchers to consider non-monotonic cooling schedules. Glover (1995) describes a method of non-monotonic trajectories in many different optimization settings. Osman (1993) applied such non-monotonic cooling schedules to SA for the vehicle routing problem.

Some approaches seek to adapt the type of cooling schedule to a particular problem instance (Ingber 1995). Since the landscape of a solution space may be hilly or smooth, how SA is cooled can have significant impact on the quality of the solution. This requires some *a priori* knowledge of the landscape which may not always be possible. But coupling the annealing process with some *actual* knowledge of how well the system is "settling down" could be even more beneficial. Like those hand-held amusements where one tries to get a small bead into a hole, the problem of annealing is to settle into the optimum state. The closer the bead is to the hole, the more delicately (lower temperature) one manipulates the puzzle. If the bead is far from the hole, the more vigorously the puzzle is manipulated. The use of some feedback information (discussed below) could therefore lead to improvements in SA's performance.

6.2 Parallel Simulated Annealing

Another avenue of active research involves the use of parallel computing. From massively parallel systems, known as Boltzmann Machines (Aarts and Korst 1989), to smaller scale parallelization (Azencott 1992), research on effective parallel algorithms is moving along at a swift pace and SA is at the heart of many of these techniques. Many problems using parallel computing schemes are based on *SPMD* (single program, multiple data) designs where it is not necessary to break up problems into independent subproblems. This allows parallel SA to be applied to virtually any COP.

The rationale for parallel SA is that if two heads are better than one, then two computers should be better than one. Intuitively, if two processors executing SA are applied to a given problem, they ought to find the global optimum sooner than only one processor. Roughly speaking, each processor would have to cover or search only half the solution space in a given amount of time. Adding more processors could speed things up even more.

But some method of communication must exist between the parallel processors. Otherwise all we can expect is two processors working independently and solving the problem as efficiently as a single processor. Indeed, two independent processors are just that—two examples of a single processor. The only difference compared to a single processor is that our electric bill would double.

In Boltzmann Machines, this communication is at the heart of its architecture. Each computer would be analogous to, for example, a node in the graph problem described earlier. Each node would then execute SA in order to minimize some function based on the weight of selected arcs which are connected to other nodes and, hence, processors (see Aarts and Korst 1989 for a complete description). For those types of problems that do not have a structure amenable to Boltzmann machines, the *SPMD* architecture is available.

6.3 Cybernetic Optimization by SA

Certainly, if some problem is to be solved by SA, then knowing the best direction to move in (which candidate solutions to accept) would natually be advantageous. More computational power coupled with more *intelligent* search patterns would obviously improve the efficiency of SA (see Glover 1989). But because SA is a *random* search algorithm, only *probabilistic information* is available for feedback purposes. Thus, there is really no way to tell, with certainty, if the current solution is optimal or not. But it is useful information nonetheless. Fleischer and Jacobson (1995) and Fleischer (1995) have recently developed a method, cybernetic optimization by simulated annealing (COSA), that takes advantage of such probabilistic information. This method provides useful feedback information *and* encompasses a method of

parallelization.

This information comes from other parallel computers solving the same or related problems using SA and takes advantage of SA's convergence properties. Because of this property, parallel processors will tend to converge to the same or similar solutions. Consequently, the proximity of two or more solutions make it possible to develop a **P**robabilistic **M**easure of **Q**uality (*PMQ*) that can be used to generate an error signal. This *PMQ* can then be used to modify the temperature control parameter that governs the search mechanism inherent in SA. Thus, rather than altering some direction, rate or acceleration by feedback, as in many feedback control systems, we can alter the *probabilities and associated uncertainty levels* by feedback. This concept forms the foundation of what is referred to as a **P**robabilistic **F**eedback **C**ontrol network. Put another way, if one is searching for gold (an inherently probabilistic endeavor), one is more likely to succeed if one spends more time looking where all the other prospectors are looking. SA can thus provide both aspects of feedback control: the generation of an error signal and the capability to utilize that information to modify a control parameter.

The idea behind the *PMQ* is basically this: if two or more processors generate solutions that are close to the optimum an increasing proportion of time (*i.e.*, convergence in probability), then the two solutions should be *close to each other* an increasing proportion of time. Conversely, *if two or more processors generate solutions that are "close" to each other in some sense, then it is more likely that at least one is close to the optimum.* Measuring how close two or more solutions from parallel processors are to each other can therefore provide information as to the quality of the solutions, the *PMQ*. This probabilistic information provides the basis for the generation of an error signal.

The idea behind the control scheme is this: when it is more likely that at least one processor is producing a relatively good solution, then the algorithm should lower the information rate and uncertainty associated with the next state by lowering the temperature. This makes it more likely that the algorithm will explore solutions that are nearby for a greater number of iterations. Conversely, if the solutions are likely to be poor, then increasing the information rate and uncertaintly levels by raising the temperature is appropriate. This would increase the randomness of the search and allow SA to explore more remote regions of the solution space.

The PMQ metric can be based on the degree of coincidence of the sets of solutions from several processors such as the intersection set or the symmetric difference of the sets depending on the nature of the problem (Fleischer and Jacobson 1995). The size of these sets can be used to modulate the temperature in the manner described above (see Fleischer and Jacobson 1995 for details).

7 CONCLUSION

From early attempts to simulate thermodynamical systems to optimization strategies to parallel computing, SA seems to be involved in many threads of research. Recent advances and applications, in addition to its use with other optimization strategies, suggest that SA will be intimately involved in many successful research endeavors well into the future. It is indeed, a *cool* algorithm notwithstanding some of the heat it has generated in the past.

ACKNOWLEDGEMENTS

This research was partially supported by the National Aeronautics and Space Administration, Contract NAS1-19858-13. The author would like to thank Sheldon H. Jacobson and David Goldsman for their support in the writing of this article.

REFERENCES

Aarts, E. and J. Korst. 1989. *Simulated annealing and Boltzmann machines: a stochastic approach to combinatorial optimization and neural computing*, New York, N.Y. John Wiley & Sons.

Azencott, R. ed. 1992. *Simulated annealing: parallelization techniques*, New York, N.Y., John Wiley & Sons, Inc.

Bohachevsky, I., M. Johnson, and M. Stein. 1986. Generalized simulated annealing for function optimization, *Technometrics*, Vol. 28, No. 3, pp. 209-217.

Bondy, J.A. and U.S.R. Murty. 1976. *Graph theory with applications*, New York, N.Y. North-Holland, Elsevier Science Publishing Co., Inc.

Bonomi, E. and J. Lutton. 1984. The n-city travelling salesman problem: statistical mechanics and the Metropolis algorithm, *SIAM Review*, Vol. 26, No.4. pp. 551- 568.

Cheh, K., J. Goldberg and R. Askin. 1991. A note on the effect of neighborhood structure in simulated annealing. *Computers in Operations Research*, Vol. 18, No. 6, pp. 537-547.

Cĕrny, V. 1985. A thermodynamical approach to the travelling salesman problem: an efficient simula-

tion algorithm. *Journal of Optimization Theory and Application*, Vol. 45, pp. 41-55.

Dowsland, K. 1993. Simulated annealing, in *Modern Heuristic Techniques for Combinatorial Problems*, C. Reeves, editor, New York, John Wiley & Sons.

Eglese, R.W. 1990. Simulated annealing: a tool for operational research, *European Journal of Operational Research*, Vol. 46, No. 3. June 15, 1990. pp. 271-281.

Fleischer, M. 1993. Assessing the performance of the simulated annealing algorithm using information theory. Doctoral dissertation, Department of Operations Research, Case Western Reserve University, Cleveland, Ohio.

Fleischer, M. and S. H. Jacobson. 1994. Information theory and the simulated annealing algorithm: experimental results. Technical Report, Department of Engineering Management, Old Dominion University, Norfolk, VA.

Fleischer, M. 1995. Cybernetic optimization by simulated annealing: accelerating convergence by parallel processing and probabilistic feedback control, Technical Report. Department of Engineering Management, Old Dominion University, Norfolk, VA.

Fleischer, M. and S. H. Jacobson. 1995. Cybernetic optimization by simulated annealing: an implementation of parallel processing using probabilistic feedback control. In *Metaheuristics: The State of the Art 1995* (Proceeding of the Metaheuristics International Conference 1995), ed. J. Kelly, I. Osman, Kluwer Academic Publishers, New York, N.Y. To appear.

Garey, M. and D. S. Johnson. 1979. *Computers and intractability: a guide to the theory of NP-completeness*, New York, N.Y. Freeman and Co.

Glover, F. 1989. Tabu search-part I. *ORSA Journal on Computing*, Vol. 1, No. 3. pp. 190-206.

Glover, F. 1995. Tabu thresholding: improved search by nonmonotonic trajectories. To appear in *INFORMS Journal on Computing.*

Goldstein, L. and M. Waterman. 1988. Neighborhood size in the simulated annealing algorithm. *American Journal of Mathematical and Management Sciences*, Vol. 8, Nos. 3 & 4. pp. 409-423.

Ingber, L. 1995. Adaptive simulated annealing: lessons learned. *Journal of Control and Cybernetics.* To appear.

Johnson, D. S., C. R. Aragon, L. A. McGeoch and C. Schevon. 1989. Optimization by simulated annealing: an experimental evaluation, part I (graph partitioning). *Operations Research*, Vol. 37, No. 6. November-December, 1989. pp. 865-892.

Kirkpatrick, S., C. D. Gelatt, Jr., and M. P. Vecchi. 1983. Optimization by simulated annealing. *Science*, Vol. 220, No. 4598, May 13, 1983. pp. 671-680.

Manber, U. 1989. *Introduction to algorithms: a creative approach*, New York, N.Y. Addison-Wesley Publishing Company.

Metropolis, N., A. Rosenbluth, M. Rosenbluth, A. Teller and E. Teller. 1953. Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, Vol. 21, pp. 1087 - 1092.

Mitra, D., F. Romeo and A. Sangiovanni-Vincentelli. 1986. Convergence and finite- time behavior of simulated annealing. *Advances in Applied Probability*, Vol. 18, pp. 747-771.

Osman, I. 1993. Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. *Annals of Operations Research*, Vol. 41, pp. 421.

Tovey, C. 1988. Simulated simulated annealing. *American Journal of Mathematical and Management Sciences*, Vol. 8. Nos. 3&4, pp. 389-407.

Wiener, N. 1948, 1961. *Cybernetics: or control and communication in the animal and the machine*, 2nd ed. Cambridge, Massachussetts. The M.I.T. Press.

Yao, X. 1991. Simulated annealing with extended neighborhood. *International Journal of Computer Mathematics*, Vol. 40, pp. 169-89.

AUTHOR BIOGRAPHY

MARK FLEISCHER is an Assistant Professor in the Department of Engineering Management at Old Dominion University. He received his Ph.D. degree in Operations Research from Case Western Reserve University (1994), a J.D. degree from Cleveland State University (1984), and a B.Sc. degree from the Massachusetts Institute of Technology (1978). Before his stint as an attorney, Dr. Fleischer worked in the Office of Management and Budget in New York City where he was an early advocate of applying operations research methods to the city's management problems. He is currently conducting research in optimization using a cybernetic optimization scheme for the National Aeronautics and Space Administration at the Langley Research Center. His research interests include applied probability, optimization, parallel computing, information theory and control theory and how these areas interface. He is an active member of INFORMS and its special interest section on computer science.