Computer Science Department Technical Report
University of California
Los Angeles, CA 90024-1596

BEST-SO-FAR VS. WHERE-YOU-ARE: NEW DIRECTIONS IN SIMULATED ANNEALING FOR CAD

K. D. Boese
A. B. Kahng
A. C.-W. Tsao

December 1992
CSD-920050

Best-So-Far vs. Where-You-Are: New Directions in Simulated Annealing for CAD*

Kenneth D. Boese, Andrew B. Kahng and Chung-Wen Albert Tsao

UCLA Computer Science Department, Los Angeles, CA 90024-1596

Abstract

Simulated annealing (SA) [23] [6] has been widely used for heuristic global optimization in VLSI layout design, and is attractive both for its observed high-quality results and for its ability, in theory, to yield optimal solutions with probability one. Standard SA implementations use a *monotone decreasing*, or "cooling", temperature schedule motivated by the algorithm's proof of optimality as well as an analogy with statistical thermodynamics. In this paper, we challenge this motivation: the fact that cooling schedules are "optimal" *in theory* does not have any relation to the *practical* performance of the algorithm. Our work is based on a new "best-so-far" (BSF) criterion that we propose as the proper measure of the *practical* utility of a given annealing schedule. For small instances of several classic VLSI CAD problem formulations, including circuit placement and graph bisection, we determine annealing schedules that are *optimal* in terms of expected quality of the output solution. When the goal is solely to optimize the quality of the *last* solution seen by the algorithm (the "where-you-are" criterion used in previous theoretical analysis), we confirm the traditional wisdom of SA cooling schedules. However, if the goal is to optimize the *best* solution quality seen over the entire algorithm execution (what we call the "best-so-far" criterion), we give clear evidence that optimal schedules do not decrease monotonically toward zero, and are in fact *periodic* or *warming*. These results open up many interesting research issues regarding the BSF analysis of stochastic hill-climbing, and how to best apply stochastic hill-climbing to VLSI layout design and other difficult problem domains.

1 Preliminaries

Given a set S of feasible solutions and a real-valued cost function $f : S \rightarrow \Re$, *global optimization* may without loss of generality be formulated as the search for a global minimizer $s \in S$ such that $f(s) \leq f(s')$ $\forall s' \in S$. Typically, $|S|$ is very large compared to the number of solutions that can be examined in practice. For small instances of certain global optimizations, implicit enumeration (e.g., branch-and-bound) or polyhedral approaches can prune the solution space and afford solutions within practical time limits; other problem formulations may be tractable to problem-specific methods. However, many important global optimization formulations are not only NP-complete [8], but also have no known problem-specific solution methods. Therefore, general-purpose heuristics are of interest.

*Partial support for this work was provided by a GTE Graduate Fellowship and by NSF MIP-9110696, NSF Young Investigator Award MIP-9257982, ARO DAAK-70-92-K-0001, and ARO DAAL-03-92-G-0050. We are also grateful for the support of Cadence Design Systems under the State of California MICRO program.

1.1 Iterative Optimization Heuristics

General-purpose global optimization heuristics may almost always be viewed as iteratively applying the following two rules:

- Rule 1: Given the current solution s_i, generate a new trial solution s'.
- Rule 2: Decide whether to set $s_{i+1} = s_i$ or $s_{i+1} = s'$.

Rule 1 induces the notion of a ***neighborhood structure*** over S, where the neighborhood $N(s_i)$ of the current solution $s_i \in S$ is the set of possible trial solutions s' that can be generated from s_i. The quality of all the solutions in S defines a *cost surface* over the neighborhood structure, and global optimization is the search for a global minimum in this cost surface. Typically, the neighborhood $N(s)$ consists of a set of slight perturbations of the current solution s, e.g., a swap of two modules in a circuit placement or bisection, or a swap of two city positions in a traveling salesman tour. In practice, Rule 1 simply picks a random $s' \in N(s_i)$ from within "obvious" neighborhood structures such as those noted for the placement and bisection problems [18]. Therefore, it is Rule 2 which determines the nature of an optimization heuristic as it traverses the cost surface.

A simple instance of Rule 2 is, "Replace s_i by s' if $f(s') < f(s_i)$," which corresponds to greedy optimization. Greed has been widely employed because of its simplicity and its acceptable success in a variety of implementations, e.g., Johnson et al. [17] [18] have documented the utility of greed for several hard combinatorial problems. However, the performance of greedy methods is erratic, and achieving "stable" – i.e., predictable – performance requires multiple random initial starting solutions. Johnson et al. [17] have determined that several thousand initial random starting configurations are necessary for greed to afford stable solution quality for graph bisection instances of size $n = 500$; this number grows rapidly with n and becomes hopeless for instance sizes of, e.g., $n = 100,000$ which arise in arenas such as VLSI circuit partitioning. Moreover, central limit phenomena in the cost surface [4] imply that as problems grow large, random local minima are almost surely of "average" quality, so that simple "multi-start" heuristics [40] fail.[1] In view of these factors, global optimization heuristics must escape from local minima, i.e., perform "hill-climbing", to adequately explore the solution space of large problems.

1.2 Stochastic Hill-Climbing and the "Simulated Annealing" Analogy

*Stochastic **hill-climbing*** allows escape from local minima in the cost surface by probabilistically accepting disimprovements, or "uphill moves". The most prominent such method, ***simulated annealing*** (SA), was proposed

[1] For details on this subject, the reader is referred to discussions by Baum [4] and Kirkpatrick and Toulouse [24] on large-scale traveling salesman cost surfaces; by Kauffman and Levin [21] on optimization by genetic algorithms in "adaptive landscapes"; and by Bui et al. [5] for the graph bisection problem.

independently by Kirkpatrick et al. [23] and Cerny [6] and is motivated by analogies between the solution space of an optimization instance and microstates of a statistical thermodynamical ensemble. Figure 1 summarizes the SA algorithm, which uses the following criteria for the above-mentioned Rule 2. If $f(s') < f(s_i)$, then $s_{i+1} = s'$, i.e., the new solution is adopted. If $f(s') \geq f(s)$, the "hill-climbing" disimprovement to $s_{i+1} = s'$ still has a nonzero probability of being adopted – the so-called Boltzmann acceptance criterion – which is determined by both the magnitude of the disimprovement and the current value of a *temperature* parameter T_i. Over the M steps for which the SA algorithm is executed, a temperature *schedule* $T_0, T_1, \ldots, T_{M-1}$ guides the optimization process. Typical SA practice uses a large initial temperature and a final temperature of zero, with T_i *monotonically decreasing* according to a predefined temperature schedule or some criterion for lowering the temperature (e.g., based on the current T_i value, the number of iterations since the last improvement in the cost function, or the objective of establishing "thermodynamic equilibration" at each temperature value).

```
SA Algorithm Template

0. $s_0 \leftarrow$ random solution in $S$
1. For $i = 0$ to $M - 1$
2.    Choose $s' \leftarrow$ a random element from $N(s_i)$
3.    if $f(s') < f(s_i)$
4.        $s_{i+1} \leftarrow s'$
5.    else
6.        $s_{i+1} \leftarrow s'$ with probability $e^{-[f(s')-f(s_i)]/T_i}$
7.        otherwise $s_{i+1} \leftarrow s_i$
```

Figure 1: The simulated annealing algorithm for a given time bound of M steps.

The SA algorithm enjoys certain theoretical attractions. Using Markov chain arguments and basic aspects of Gibbs-Boltzmann statistics, one can show that for any finite S, SA will converge to a globally optimal solution given infinitely large M and a temperature schedule that converges to zero sufficiently slowly [31] [37], i.e.,

$$Pr(s_M \in R) \rightarrow 1 \quad as \quad M \rightarrow \infty \tag{1}$$

where $R \subset S$ is the set of all globally optimum solutions. In other words, SA is "optimal" in the limit of infinite time [25].[2] Several groups have refined these results by noting specific temperature schedules which guarantee convergence of SA to a global optimum. For example, Hajek [13] showed that "logarithmic cooling" using $T_i = a/\log i$ for sufficiently large a will suffice to this end. Other optimal schedules are surveyed in [2] [25]. A recent result due to Sorkin [45] is that certain classes of geometric cooling schedules are efficient on one-dimensional, deterministically fractal, error surfaces.[3]

[2]The proof views the annealing process as a sequence of homogeneous Markov processes at each temperature. The main idea behind the proof is that the Boltzmann acceptance function implies that the likelihoods of two solutions $s_a, s_b \in S$ at stationarity will be respectively proportional to $exp(-f(A)/T_i)$ and $exp(-f(B)/T_i)$, so that s_a is exponentially more likely than s_b in the infinite-time limit if $f(s_a) < f(s_b)$.

[3]This last result is particularly interesting because several recent works (e.g., [2] [20] [48]) have confirmed power-law scaling

Given its theoretical and practical successes, SA is now perhaps *the* most widely used heuristic for difficult global optimizations [25]. Historically, the milestone advances in theory and application of SA are inseparable from the field of VLSI design: the original paper of Kirkpatrick et al. [23] (treating circuit placement), along with the works of Sangiovanni-Vincentelli and coauthors [34] [37], Sechen [41], Rose [38], Lam and Delosme [26], Greene and Supowit [10], and Grover [11] [12], are just a few examples. In VLSI CAD, SA has been applied to such topics as placement, floorplanning, routing, logic minimization, PLA folding, compaction, and transistor sizing, as well as a host of other applications [2] [25] [49]; indeed, SA has become a dominant methodology across the spectrum of synthesis and layout tools.

1.3 Motivations: Finite-Time Annealing

Despite the tremendous success of existing simulated annealing implementations, there are strong reasons to pursue new ideas in annealing. Our central motivation is practical: given a problem instance, the ideal global optimization algorithm should return a *good* solution in a *prescribed, finite* amount of time. Traditional SA implementations are often ill-suited to prescribed time bounds, the template of Figure 1 notwithstanding. Moreover, as noted by such researchers as Sorkin [44], the "infinite-time optimality" of SA is of questionable utility: after all, even exhaustive search and random search are also "optimal" in the limit of infinite CPU resources.

Through the study of practical, *finite-time* global optimization, we have discovered a fundamental inconsistency in the simulated annealing methodology which stems from the theoretical, *infinite-time* analysis. This inconsistency is crystallized in the dichotomy between what we term the "best-so-far" (BSF) and the "where-you-are" (WYA) criteria for annealing schedules. Section 2 defines this BSF-WYA dichotomy, and then presents an experimental methodology which tests the practical effect of choosing one criterion over the other. In Section 3, we present extensive experimental results over a number of problem formulations, including circuit placement and graph bisection. Our results show that use of the more realistic BSF criterion turns the traditional wisdom of "cooling" schedules quite literally on its head: while cooling is appropriate for optimizing the traditional WYA criterion, we find that BSF-optimal annealing schedules are both non-monotone and non-cooling, and may even look like "warming" schedules. Together, our proposal of BSF analysis and the actual solution of BSF-optimal annealing schedules provide the main contribution of this work. We conclude in Section 4 by considering the implications of this work for real-scale annealing optimizations, and by listing directions for future work.

relationships in the cost surfaces of large combinatorial optimization instances. These works also show very good statistical fits of real cost surfaces to models of high-dimensional fractional Brownian motions, which are a class of statistical fractals.

2 A New Look At Simulated Annealing

2.1 Best-So-Far Versus Where-You-Are

Throughout the literature on stochastic hill-climbing methods, the SA algorithm is universally implemented with *monotone decreasing* temperature schedules. As surveyed in [2] [25] [49], hundreds of papers have been written on various "cooling" and "equilibration" approaches, and many new fields (e.g., finite-time thermodynamics [35] [39], rapid mixing of Markov chains [15], etc.) have opened up as a result of these investigations. The thermodynamic analogy suggests that monotone decreasing temperature schedules allow SA to explore "large features" of the cost surface at high T, then perform finer optimization at lower T. In fact, this is the basic idea behind the SA proof of optimality [31] [37]. However, our results below suggest that the analogy with physical annealing, along with the theoretical analysis of hill-climbing, have together led researchers to *incorrectly* concentrate on the "cooling" paradigm. Specifically, we note that the "optimality" of SA (Equation 1) has always been analyzed with respect to what we call a "where-you-are" (WYA) implementation of the algorithm. According to the theoretical analysis, at the final time step M the SA algorithm simply returns the last solution seen (i.e. s_M, which is indeed "where you are"), and it is this *single solution* that in the limit of $M \to \infty$ has probability 1 of being optimal (see Figure 2 (left)). This theoretical model is completely at odds with common sense: *in practice*, no implementation will simply ignore all of the solutions $s_0, s_1, \ldots, s_{M-1}$. At the very least, we can remember the best solution seen so far, and return it if the final solution s_M is not as good (this is indicated in the "best-so-far" (BSF) template of Figure 2(right)). While any practical SA implementation will return the best-so-far solution, this is never mentioned in any standard description of SA. Moreover, the distinction between BSF and WYA is moot with respect to traditional convergence proofs, since "optimality" of a WYA implementation trivially implies "optimality" of its BSF counterpart. Thus, the BSF analysis of simulated annealing is for all purposes completely absent from the literature.

SA WYA Implementation

```
0. $s_0 \leftarrow$ random solution in $S$
1. For $i = 0$ to $M - 1$
2.    Choose $s' \leftarrow$ a random element from $N(s_i)$
3.    if $f(s') \leq f(s_i)$
4.          $s_{i+1} \leftarrow s'$
5.    else
6.          $s_{i+1} \leftarrow s'$ with probability $e^{-[f(s')-f(s_i)]/T_i}$
7.          otherwise $s_{i+1} \leftarrow s_i$
8. Return $s_M$
```

SA BSF Implementation

```
0. $s_0 \leftarrow$ random solution in $S$
1. For $i = 0$ to $M - 1$
2.    Choose $s' \leftarrow$ a random element from $N(s_i)$
3.    if $f(s') \leq f(s_i)$
4.          $s_{i+1} \leftarrow s'$
5.    else
6.          $s_{i+1} \leftarrow s'$ with probability $e^{-[f(s')-f(s_i)]/T_i}$
7.          otherwise $s_{i+1} \leftarrow s_i$
8. Return $s_i$, $0 \leq i \leq M$, such that $f(s_i)$ is minimum.
```

Figure 2: (Bounded-time) simulated annealing templates, contrasting WYA and BSF implementations in Line 8.

2.2 Related Work

As noted in Section 1, our main contributions lie in opening BSF annealing as a field of study, and in determining *optimal* BSF schedules for a variety of problem classes. To the best of our knowledge (and after extensive search through the literature), all of our contributions – the experimental protocol, our results, and the implications of these results – are completely new. However, we have found two isolated hints of BSF analysis in the existing literature on simulated annealing. The more direct hint is contained in the 1988 work of Hajek [13], which establishes necessary and sufficient conditions under which a monotone decreasing cooling schedule will be guaranteed to yield a globally optimum solution in infinite time. The result is established for SA under the WYA criterion; however, the possibility of BSF analysis is briefly suggested by Hajek, in the following sentence:[4] "It would be interesting to know the behavior of $min_{n \leq k} V(X_n)$ rather than the behavior of $V(X_k)$."

A weaker allusion to BSF annealing analysis is contained in the 1989 paper of Hajek and Sasaki [14], which discussed the possibility of non-conventional annealing schedules. The main result of [14] was showing the existence of a special class of optimization problems for which monotone cooling schedules are suboptimal. An ancillary result in the paper was that under a neighborhood structure in which the difference between the cost of any two adjacent solutions is zero or a constant, there exists an optimal annealing schedule where all T_i are either 0 or $+\infty$ (cf. our results showing optimal "periodic" schedules in Section 3 below). While the BSF criterion is not mentioned in [14], the authors of the paper suggest two measures of schedule quality; the first measure is simply the WYA criterion, while the second is given by the expected number of steps required to first encounter a solution with cost less than or equal to some constant c^+. Clearly, this latter quality measure holds some similarities to our BSF criterion.[5]

Finally, we note that our motivating studies of optimal schedules with prescribed lengths follows in the direction established by Strenski and Kirkpatrick [46]. Strenski and Kirkpatrick studied a highly structured graph bisection instance with eight nodes, and used numerical methods to estimate optimal schedules according

[4](Here, we have quoted Hajek's notation, which is clear from context.) As it turns out, the infinite-time optimality of BSF schedules is amenable to some analysis. Recall from our earlier discussion that any optimal WYA schedule is trivially optimal in the BSF sense. Moreover, under the weak assumption that the Markov chain for annealing at infinite temperature is irreducible, it is easy to show that any temperature schedule bounded away from zero and given infinite time will reach the global optimum at least once with probability 1. This yields a set of conditions which are sufficient, but not necessary, for a schedule to be optimal under the BSF criterion. The main result in [13] is a necessary and sufficient condition for a schedule to be optimal under the WYA criterion. To be specfic, if $\sum_{i=1}^{\infty} \exp(-d^*/T_i) = +\infty$, where d^* is a constant depending on the configuration of the solution space S (more precisely, d^* is the maximum "basin depth" around any local minimum that is not a global minimum; see [13] for a formal definition of basin depth), then the schedule is optimal. Such a result is intuitively very reasonable (it simply means that SA will expect to escape from any local minimum without requiring infinite time to do so), but it holds only for the class of schedules that are monotone decreasing and have limit $T_\infty = 0$. We conjecture that the necessary conditions for infinite-time optimality under BSF are essentially the same as Hajek's conditions for WYA optimality, except that the schedule need not be either monotone decreasing or have a limiting value of zero.

[5]Two additional references to BSF implementation of the SA algorithm are given in [20] and [42]. However, each of these works was concerned with "scaling" phenomena of large-scale optimization cost surfaces (recall the discussion of Footnote 3), and did not treat any quality measure for BSF annealing schedules.

to the criterion of expected WYA solution quality. The optimum WYA schedules found in [46] are essentially monotone decreasing to zero, except for a small initial run $T_i = 0$ before the monotone schedule kicks in.[6] In what follows, we develop a technique which also estimates optimal annealing schedules, and which can apply to either BSF or WYA analysis. In Section 3, we achieve a direct contrast with the optimal WYA annealing schedules of [46] by estimating optimal BSF schedules for the same graph bisection instance.

2.3 Experimental Methodology: Computing Optimal Finite-Length Schedules

For any given finite schedule length M, an optimal schedule is one which yields best expected solution quality after M steps. We compute optimal schedules based on one-step *transition matrices* $A(T_i)$ which are induced over the solution space by each possible T_i value. For a given T_i, $[A(T_i)]_{jk}$ is the probability of moving from solution s_j to solution s_k when the temperature parameter is equal to T_i. (Here, we abuse notation with respect to the indices j and k, which are used for arbitrary solutions $s_j, s_k \in S$. We retain the original interpretation for all other subscripts of s, e.g., s_i still denotes the i^{th} solution encountered during the SA run.) We let C denote the $|S| \times 1$ column vector of costs for each solution in S, and we let P denote the starting distribution, i.e., the $|S| \times 1$ vector of probabilities that each solution in S is chosen as the initial solution s_0.

- To optimize the WYA criterion, the expected WYA quality of a given schedule can be calculated by simply taking the product of the transition matrices for all the T_i in the schedule. In other words, the expected WYA solution quality is given by

$$E[f(s_M)] = C \cdot A(T_M) \cdot A(T_{M-1}) \cdot \cdots \cdot A(T_1) \cdot P.$$

To optimize the BSF criterion, we have developed a variation of this methodology where solution states are converted into "sinks" in order to record whether they have ever been visited. To be specific, the solution $s_j \in S$ is converted into a sink by setting $[A(T_i)]_{jj} = 1$ and $[A(T_i)]_{jk} = 0$ $(k \neq j)$ for all T_i. This leads to two distinct BSF quality measures, each of which is amenable to optimization:

- To optimize the probability that the BSF solution is optimal, we first denote the global optimum solution by $s_j \in S$, and then convert s_j into a sink in order to yield perturbed transition matrices T_i'. We then have

$$Pr(min_{0 \leq i \leq M} f(s_i) = f(s_j)) = [A(T_M') \cdot A(T_{M-1}') \cdot \cdots \cdot A(T_1') \cdot P]_j. \quad (2)$$

In other words, the j^{th} component of the $|S| \times 1$ matrix product shown will give the probability that the M^{th} transition is *to* the state s_j. Since s_j is set up as a sink, this j^{th} component actually gives the

[6] Intuitively, the initial $T_i = 0$ temperatures allow the SA algorithm to quickly reach the "interesting" region of the cost surface, which are near the local minima. See the discussion of the "Energy Landscape Conjecture" that we propose below.

probability that we reached s_j at any point during the M steps of the annealing run.

- More generally, the probability of ever reaching a solution of cost $\leq c$ is obtained by converting all solutions with cost $\leq c$ into sinks, generating transition matrices T_i' just as above, and then summing the entries of the $|S| \times 1$ matrix $A(T_M') \cdot A(T_{M-1}') \cdot \cdots \cdot A(T_1') \cdot P$ which correspond to solutions having cost $\leq c$. By lowering c from $c = f(s_j)$ to $c = f(s_j) - \epsilon$ and then observing the resulting change in this sum, we can obtain the probability that the BSF solution cost will be exactly $f(s_j)$. To compute the expected BSF solution cost, i.e., $E[min_{0 \leq i \leq M} f(s_i)]$, we compute the probability that each solution cost $f(s_j)$ is the BSF cost. This yields a $1 \times |S|$ row vector, and we take the appropriate weighted average of the solution costs by computing the inner product of this row vector with the cost vector C.

Note that the linear form of the WYA equation makes it relatively easy to calculate the partial derivative of the WYA quality, i.e., $E[f(s_M)]$, with respect to each T_i. Such a calculation was used by Strenski and Kirkpatrick [46] in estimating optimal WYA schedules via a gradient method. Because the corresponding equations for expected BSF solution quality do not lend themselves to an extension of this methodology, we use a different and more general methodology in our work, as follows.

In our experiment, we select optimal schedules based on a discrete set of 100 evenly-spaced temperature values T_i such that the lowest possible T_i is 0 and the highest possible value is effectively $+\infty$.[7] When M is very small (e.g., $M \leq 8$, or $M \leq 12$ if fewer distinct T_i values are allowed), we can exhaustively enumerate all possibilities and determine globally optimal schedules. For larger values of M, exhaustive enumeration of all possible temperature schedules is impossible, and we therefore use a perturbative method to generate locally optimal schedules. Here, the current schedule is perturbed deterministically, and we adopt the single-step change in a single temperature T_i which yields the greatest improvement in overall schedule quality. The iterative process is terminated when a locally optimal schedule is found (note that this recalls the gradient method used by Strenski and Kirkpatrick [46] to determine locally optimal WYA schedules). For each estimation, we begin with several different initial schedules, some with all T_i = some constant for $1 \leq i \leq M$, and others with all T_i randomly chosen; we then adopt the best-quality schedule found. With respect to this point, we stress that we observed very few distinct locally minimum schedules, with *all* of these locally minimum schedules (including the best one) being qualitatively very similar (in other words, simply finding the locally optimal schedule obtainable from, say, an initial schedule of all $T_i = 100$ would yield results that are essentially identical to those we report). Furthermore, for all values of M up to the limits of our available hardware (Sun Sparc 1+), we have observed that exhaustively determined globally optimal schedules are essentially identical to locally optimal schedules.

[7] We experimented with using $T_i \in \{1, 2, \ldots, 100\}$ as well as T_i chosen such that the transition probabilities for an "average" uphill move end up being $0.01, 0.02, 0.03, \ldots, 0.99, \approx 1.00$. Results were qualitatively the same with either of these methodologies.

With respect to the BSF criterion, we have estimated optimal length-M schedules for each of the two distinct objectives, $E[BSF\ cost]$ and $Pr(BSF\ cost = opt)$. We have found that the resulting schedules are essentially identical. Below, we report schedules that are (locally) optimal with respect to $E[BSF\ cost]$, because we believe that in practice the average quality of the annealing solution is more useful information than the probability that it is globally optimum. Finally, we note that because each $A(T_i)$ is of size $|S| \times |S|$, this computation is only feasible for very small problem instances, since the solution space S often grows exponentially with the problem size n (e.g., n = number of modules in a circuit placement instance corresponds to $S = n!$).

3 Experimental Results: Optimal BSF Annealing Schedules

In this section, we study small instances of three combinatorial optimizations (graph bisection, graph placement, and the traveling salesman problem) which are prominent in the VLSI CAD literature. We also study a fourth optimization that is embedded into a synthetic, "levels-oriented" cost structure which captures recent scaling models for large-scale optimization cost surfaces. For each of these four examples, we use the methodology of Section 2.3 above to solve for locally optimal BSF (expected $min_{0 \le i \le M} f(s_i)$) and WYA (expected $f(s_M)$) schedules; we then evaluate each schedule according to both the BSF and the WYA criteria.

3.1 Graph Bisection

The graph bisection problem is stated as follows: Given a graph $G = (V, E)$ with $|V|$ even, partition V into disjoint U and W, with $|U| = |W|$, such that the number of edges $(u, w) \in E$ with $u \in U$, $w \in W$ is minimized. Graph bisection is basic to recursive netlist partitioning in top-down layout design, as well as floorplanning, area estimation, etc. Use of annealing to solve the graph bisection problem is less common than use of such iterative greedy methods as the Kernighan-Lin algorithm [22] or its enhancement by Fiduccia and Mattheyses [7]. Nevertheless, SA has been well-studied in the context of graph bisection, notably by Johnson et al. [17]. The annealing algorithm has also been carefully compared against iterative methods by Bui et al. [5] and [47].

We first study the same highly-structured instance that was treated by Strenski and Kirkpatrick in [46]. This instance consists of a complete graph of eight nodes, with edge weights calculated as shown in Figure 3(a). Each of the eight nodes is represented by a leaf in the height-3 binary tree shown in the figure; the edge between any two nodes has weight α^k, where k is the height of the least common ancestor between the two nodes in the binary tree. Both our experiments and those of [46] use $\alpha = 3$. The globally optimum partition is $\{1, 2, 3, 4\}\{5, 6, 7, 8\}$, which corresponds to solution A in Figure 3(b) with $cost = 16$. Because of the symmetries in the edge weight construction, there are only five classes of equivalent-cost solutions.

However, the multiplicities of these classes, and the relative probabilities of the transitions among them, are both highly non-uniform, as seen in Figure 3(b). Following the experimental protocol described above, we computed locally optimal SA temperature sequences using the discrete range of possible temperatures $\{0, 1, 2, \ldots, 100\}$. Assuming that all initial temperature sequences are equally likely, a locally optimal WYA schedule for $M = 30$ is

$$T_{WYA} = \{0,0,0,0,6,6,6,5,5,5,5,5,5,5,5,5,5,4,4,4,4,3,0,0,0,0,0,0,0,0\}$$

which almost exactly matches the results of [46], while at the same time confirming the traditional cooling intuition. On the other hand, the locally optimal BSF schedule is completely different:

$$T_{BSF} = \{0,0,0,0,0,100,0,0,0,0,0,100,0,0,100,0,0,0,100,0,0,0,0,0,100,0,0,0,0,0.\}$$

In terms of the expected WYA solution cost $E[f(s_{30})]$, observe that T_{WYA} has cost 23.2534 while T_{BSF} for the optimal BSF schedule is much worse, with cost 31.4629. However, the expected BSF solution cost of T_{BSF} is 20.8978, while for T_{WYA} the expected BSF cost is 22.2717. Clearly, the optimal schedule in terms of the traditional objective turns out to be suboptimal when measured by its *practical*, BSF utility. It should be noted that the optimum BSF schedule seems *periodic*, and is evocative of the "iterated descent" methodologies discussed by Baum [4] and Johnson [19].

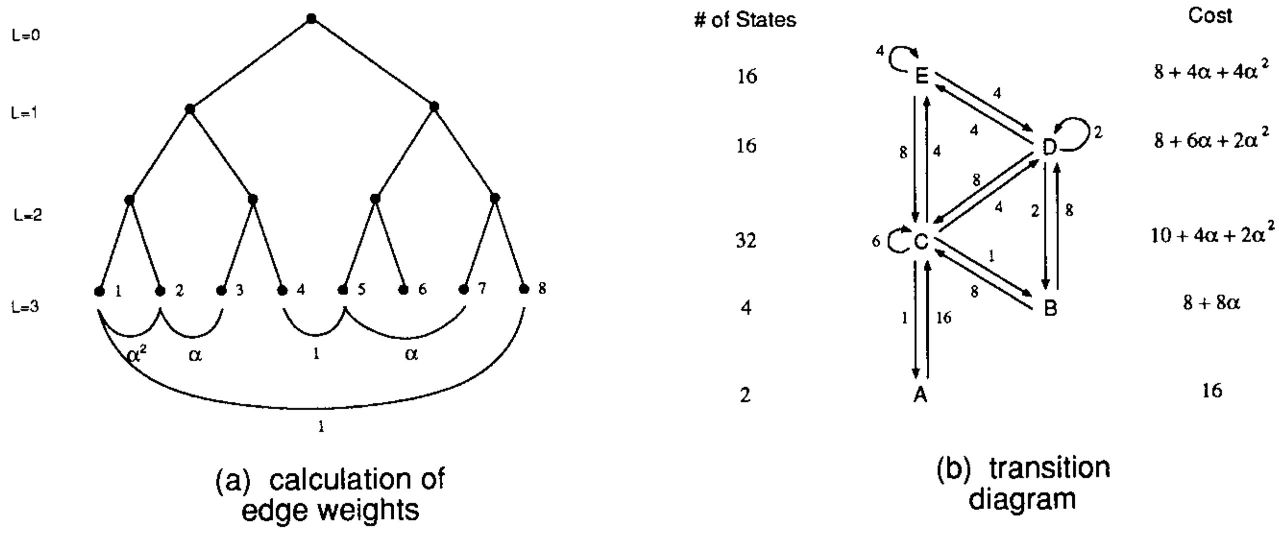


Figure 3: Edge weight calculation and state transition diagram for the complete graph used as a bisection instance by Strenski and Kirkpatrick.

Figure 4 shows the practical win of BSF annealing schedules. The figure plots the expected BSF solution quality (i.e., the "real" quality) of both locally optimal BSF schedules and the locally optimal WYA schedules. From the scaling properties of these two curves, it is apparent that a "horizontal displacement" occurs: by optimizing WYA, one will be forced to use proportionally more and more time steps in order to match the solution quality of the BSF solution. For example, in order to match the expected solution quality that the

	Number of Steps	Expected BSF Quality	Expected WYA Quality	Schedule									
Optimize BSF	5	2.038	2.038	0	0	0	0	0					
	10	1.780	1.981	0	0	0	0	100	0	0	0	0	0
	15	1.603	1.978	0	0	0	0	0	100	0	0	0	100
				0	0	0	0	0					
	20	1.475	1.920	0	0	0	0	0	100	0	0	0	100
				0	0	0	100	0	0	0	0	0	0
	30	1.306	1.967	0	0	0	0	0	100	0	0	0	0
				0	100	0	0	100	0	0	0	100	0
				0	0	0	0	100	0	0	0	0	0
	40	1.202	1.969	0	0	0	0	0	100	0	0	0	0
				0	100	0	0	0	0	100	0	0	100
				0	0	0	0	100	0	0	0	100	0
				0	10	0	0	100	0	0	0	0	0
	50	1.135	1.970	0	0	0	0	0	100	0	0	0	100
				0	0	100	0	0	0	0	100	0	0
				100	0	0	0	100	0	0	0	0	100
				0	0	0	100	0	0	11	0	0	100
				0	0	0	0	100	0	0	0	0	0
	70	1.063	2.048	0	0	0	0	0	100	0	0	0	0
				100	0	0	12	0	0	100	0	0	12
				0	0	100	0	0	12	0	0	100	0
				0	12	0	0	100	0	0	12	0	0
				100	0	0	12	0	0	100	0	0	13
				0	0	100	0	0	0	0	100	0	0
				100	0	0	0	0	100	0	0	0	0
Optimize WYA	5	2.038	2.038	0	0	0	0	0					
	10	1.797	1.797	0	0	0	0	0	0	0	0	0	0
	15	1.661	1.675	0	0	0	0	5	5	0	0	0	0
				0	0	0	0	0					
	20	1.548	1.586	0	0	0	0	6	6	5	5	5	5
				4	0	0	0	0	0	0	0	0	0
	30	1.396	1.453	0	0	0	0	6	6	6	5	5	5
				5	5	5	5	5	5	5	4	4	4
				4	3	0	0	0	0	0	0	0	0
	40	1.298	1.359	0	0	0	0	6	6	6	5	5	5
				5	5	5	5	5	5	5	5	5	5
				5	4	4	4	4	4	4	4	4	4
				3	3	0	0	0	0	0	0	0	0
	50	1.231	1.289	0	0	0	0	6	6	6	5	5	5
				5	5	5	5	5	5	5	5	5	5
				5	5	4	5	4	4	4	4	4	4
				4	4	4	4	4	4	4	4	4	3
				3	3	0	0	0	0	0	0	0	0
	70	1.143	1.120	0	0	0	0	6	6	6	5	5	5
				5	5	5	5	5	5	5	5	5	5
				4	5	4	5	4	5	4	4	4	4
				4	4	4	4	4	4	4	4	4	4
				4	4	4	4	4	4	4	4	4	4
				4	4	4	4	4	4	3	3	3	3
				3	3	0	0	0	0	0	0	0	0

Table 1: Locally optimal annealing schedules for the highly structured 8-node graph bisection instance of Strenski and Kirkpatrick [XXX]. Note that all costs are written as multiples of the optimum solution cost, which is $f(s^*) = 16.0$.

optimal 50-step BSF schedule can achieve, the WYA methodology would have to use a schedule over 70 steps in length.

The construction used by Strenski and Kirkpatrick can be generalized to the complete graph on 2^k nodes. For example, we may define a 16-node instance using edge weights from the set $\{1, \alpha, \alpha^2, \alpha^3\}$, and again use the value $\alpha = 3$. For this 16-node instance, symmetries in the solution space allow us to reduce the number of solution classes to 28. To determine the effects of scaling on the results we obtain, we computed locally optimal BSF and WYA schedules for this larger bisection instance; the results are shown in Table 2. Here, the key observations are that the optimal BSF schedules are no longer periodic, and that for runs with 50

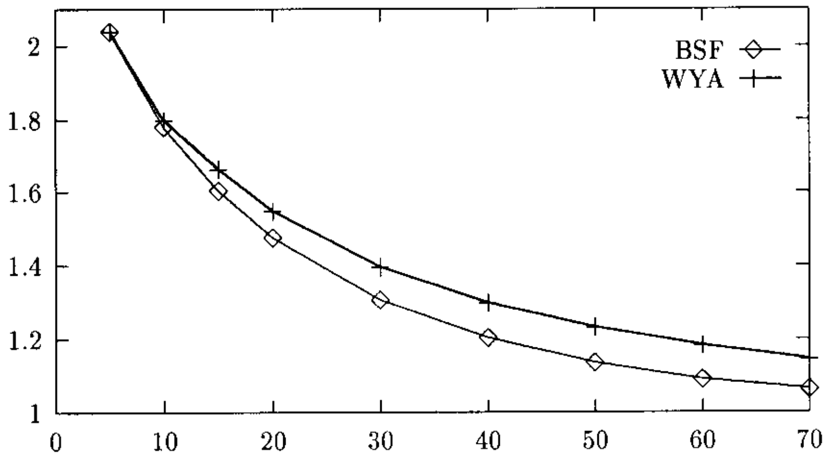


Figure 4: BSF solution qualities for BSF-optimal and WYA-optimal schedules on the eight-node graph bisection instance of Strenski and Kirkpatrick.

steps the WYA and BSF optimal schedules are very similar. This is probably due to the fact that the global minimum solution has very low cost (64) compared to its neighboring solutions (which have cost 160). It is thus very difficult to leave the global optimum solution once it has been visited.

	Number of Steps	Expected BSF Quality	Expected WYA Quality	Schedule									
Optimize BSF	25	2.643	2.655	0	0	0	0	0	0	0	0	0	4
				5	6	6	6	6	6	6	6	6	6
				6	6	6	6	0					
	50	2.274	2.313	0	0	0	0	0	0	0	0	4	6
				7	8	8	9	9	9	9	10	10	10
				10	10	10	10	10	10	10	10	10	10
				10	10	10	10	10	9	9	9	9	9
				9	9	9	8	8	8	8	8	8	0
	75	2.035	2.081										
	100	1.856	1.896										
	150	1.606	1.644										
Optimize BSF	25	2.646	2.646	0	0	0	0	0	0	0	0	0	3
				3	3	3	3	3	2	2	0	0	0
				0	0	0	0	0					
	50	2.280	2.296	0	0	0	0	0	0	0	0	6	7
				8	8	8	9	9	9	9	9	9	9
				9	9	9	8	8	8	8	8	8	7
				7	7	7	7	7	6	6	6	6	5
				5	5	4	4	3	2	0	0	0	0
	75	2.042	2.061										
	100	1.863	1.884										
	150	1.611	1.629										

Table 2: Locally optimal annealing schedules for the 16-node generalization of the bisection instance given by Strenski and Kirkpatrick. Schedules of length $M > 50$ are qualitatively similar to the 50-step schedules and are not reported.

3.2 Graph Placement

We next study another NP-complete problem, that of *graph placement*. The graph placement problem is defined as follows. Given an edge-weighted graph G with n nodes, a set L of n locations, and the distances between all pairs of locations, determine a one-to-one mapping from the nodes of G onto L which minimizes the weighted sum of distances for the edges of G. Note that this formulation captures minimum-wirelength module placement in VLSI CAD [29], which has historically provided much impetus to research in simulated annealing. As two examples, we note that the original work of Kirkpatrick et al. [23] as well as "the" annealing package (Timberwolf) [41] were both aimed at the graph placement problem.

Consider the six-node instance of the graph placement problem shown in Figure 5. If we choose the neighborhood operator to be a swap of locations for some pair of nodes, we find that there are 17 distinct solutions when symmetries are discounted; Figure 5 shows the global minimum and the unique local minimum configurations with the edge weight $\alpha = 5$. For this example, we have again solved numerically for local optimum annealing schedules of all lengths up to $M = 70$, using both the BSF and WYA criteria.

The results shown in Figure 6 are quite dramatic. The optimum 70-step WYA schedule is monotone decreasing, again as would be expected from the body of results in the current literature. However, the optimum 70-step BSF schedule is *monotone increasing*, and even though it is not very good in terms of the WYA objective, it is clearly superior to the optimal WYA schedule when judged by the "practical" BSF criterion.[8]

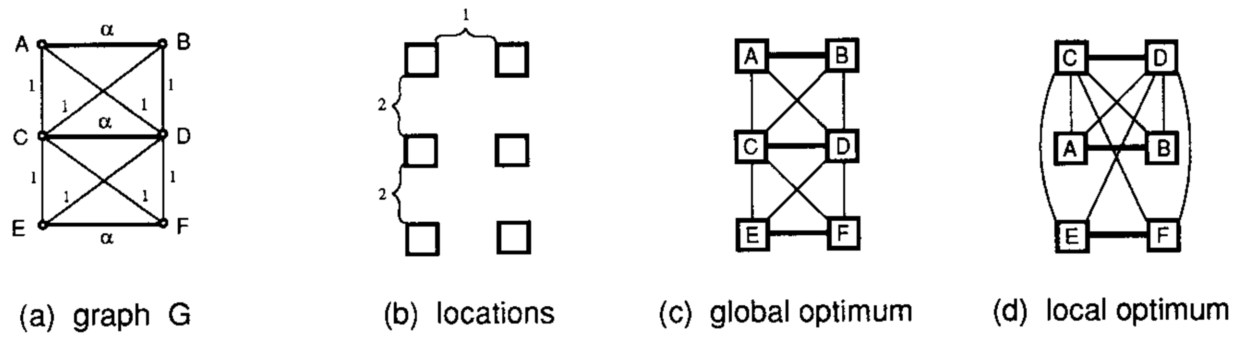


Figure 5: Six-node graph placement problem with edge weights as shown in (a); thick edges have weight $\alpha = 5$. Available locations are in the Manhattan (L_1) plane, as shown in (b). The global optimum and the local optimum solutions are respectively given in (c) and (d).

Finally, we plot in Figure 7 the expected BSF quality of both the optimal WYA schedules and the optimal BSF schedules. Here, the separation between the curves is not as large as for the graph bisection instance, seemingly indicating that the optimal WYA strategy is reasonably good in terms of the BSF criterion.

[8]By way of clarification, it should be noted that in BSF annealing, the last temperature T_{M-1} is irrelevant in the sense that an improving move s' will always be accepted no matter what the value of T_{M-1} might be. We break ties lexicographically and hence simply write $T_M = 0$ for the reported locally optimal BSF schedule.

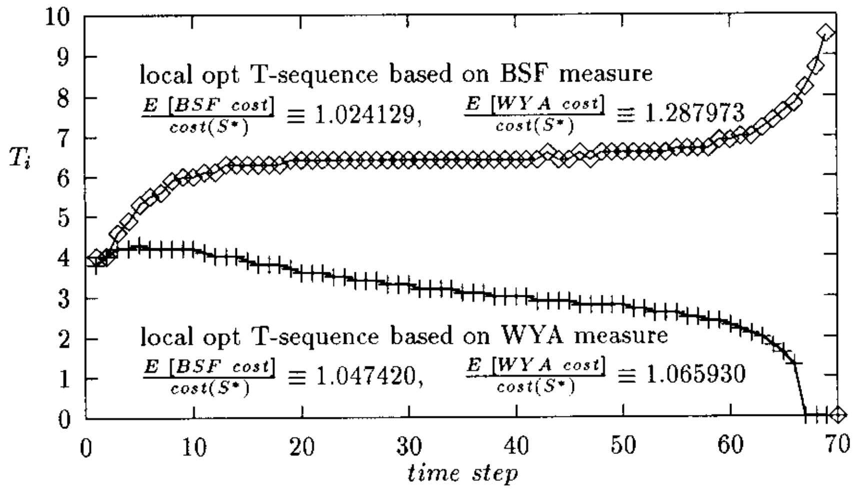


Figure 6: Locally optimal annealing schedules for linear placement example, determined by BSF and WYA measures. $S*$ denotes the optimum solution.

However, closer examination of the data shows that again, fairly large "horizontal" displacements arise as M grows larger: for example, the BSF quality of the longest WYA-optimal schedule can be achieved by a BSF-optimal schedule that is over 20% shorter.

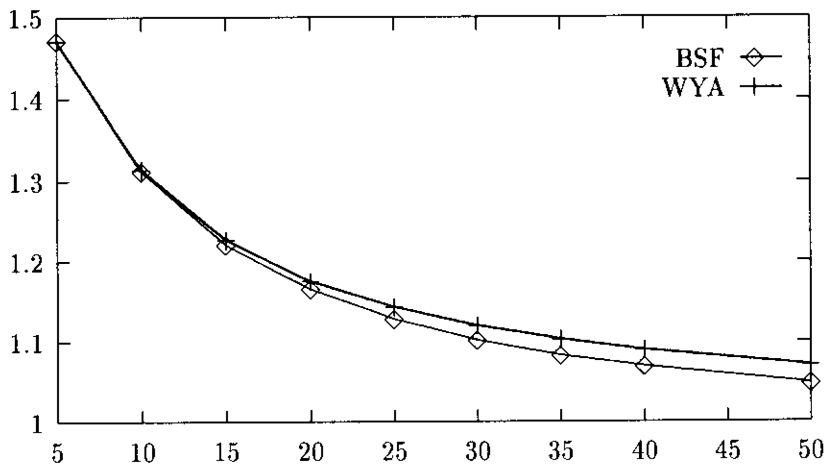


Figure 7: BSF solution qualities for the BSF-optimal and WYA-optimal schedules on the six-node placement instance.

	Number of Steps	Expected BSF Quality	Expected WYA Quality	Schedule									
Optimize BSF	5	1.471	1.473	0.6	1.0	1.2	1.3	0.0					
	10	1.312	1.326	0.9	1.6	2.1	2.5	2.8	3.1	3.2	3.2	3.1	0.0
	15	1.221	1.268	1.6	2.1	2.6	3.1	3.5	3.7	4.0	4.1	4.5	4.6
				4.8	4.8	4.8	4.6	0.0					
	20	1.165	1.255	2.6	2.6	3.1	3.5	3.9	4.2	4.4	4.6	4.8	4.9
				5.1	5.2	5.4	5.5	5.6	5.7	6.0	6.0	6.0	0.0
	25	1.128	1.257	2.6	3.0	3.5	3.9	4.3	4.5	4.8	5.0	5.0	5.2
				5.2	5.4	5.4	5.7	5.7	5.9	5.9	5.9	6.2	6.2
				6.5	6.8	6.8	7.1	0.0					
	30	1.102	1.264	2.9	3.3	3.9	4.1	4.5	4.8	5.0	5.2	5.2	5.4
				5.4	5.7	5.7	5.7	5.9	5.9	5.9	5.9	6.2	6.2
				6.2	6.2	6.5	6.5	6.8	6.8	7.1	7.5	7.9	0.0
	35	1.083	1.270	3.1	3.5	4.0	4.5	4.8	5.0	5.2	5.4	5.4	5.7
				5.7	5.7	5.9	5.9	5.9	5.9	5.9	5.9	6.2	6.2
				6.2	6.2	6.2	6.5	6.5	6.5	6.5	6.8	6.8	7.1
				7.1	7.5	7.9	8.4	0.0					
	40	1.069	1.276	3.3	3.7	4.1	4.6	5.0	5.2	5.4	5.4	5.7	5.7
				5.9	5.9	5.9	5.9	5.9	5.9	6.2	6.2	6.2	6.2
				6.2	6.2	6.2	6.2	6.2	6.5	6.5	6.5	6.5	6.5
				6.8	6.8	6.8	7.1	7.1	7.5	7.9	8.4	8.4	0.0
Optimize WYA	5	1.471	1.471	0.0	0.0	0.0	0.0	0.0					
	10	1.314	1.315	0.9	1.3	1.5	1.5	1.3	0.0	0.0	0.0	0.0	0.0
	15	1.227	1.230	1.9	2.1	2.4	2.5	2.6	2.6	2.4	2.3	2.1	1.8
				1.2	0.0	0.0	0.0	0.0					
	20	1.176	1.182	2.6	2.8	3.0	3.1	3.2	3.2	3.2	3.2	3.1	2.9
				2.8	2.6	2.4	2.1	1.9	1.3	0.0	0.0	0.0	0.0
	25	1.143	1.152	3.0	3.3	3.4	3.5	3.6	3.6	3.6	3.6	3.5	3.5
				3.4	3.3	3.2	3.1	3.0	2.8	2.6	2.4	2.1	1.9
				1.4	0.0	0.0	0.0	0.0					
	30	1.120	1.132	3.4	3.5	3.7	3.9	3.9	3.9	3.9	3.9	3.7	3.7
				3.7	3.6	3.6	3.5	3.4	3.4	3.3	3.2	3.0	2.9
				2.7	2.6	2.3	2.1	1.8	1.4	0.0	0.0	0.0	0.0
	35	1.103	1.117	3.6	3.7	3.9	4.0	4.0	4.0	4.0	4.0	4.0	3.9
				3.9	3.9	3.7	3.7	3.6	3.6	3.5	3.5	3.4	3.3
				3.2	3.2	3.1	2.9	2.8	2.6	2.5	2.3	2.1	1.8
				1.4	0.0	0.0	0.0	0.0					
	40	1.090	1.105	3.7	3.9	4.0	4.1	4.1	4.1	4.1	4.1	4.0	4.0
				4.0	3.9	3.9	3.9	3.7	3.7	3.6	3.6	3.5	3.5
				3.4	3.4	3.3	3.3	3.2	3.1	3.0	2.9	2.8	2.7
				2.6	2.4	2.2	2.0	1.8	1.4	0.0	0.0	0.0	0.0

Table 3: Locally optimal annealing schedules for the 3×2 graph placement problem.

3.3 The Traveling Salesman Problem

Our third set of experiments was performed on a small instance of the traveling salesman problem (TSP) with $n = 6$ cities. In addition to being one of the most well-studied problems in the combinatorial optimization literature [28], the TSP has received attention in various application areas of electronic design automation. These areas include mask lithography, plotting, PCB drilling (see, e.g., [30]), and daisy-chain signal routing; more recently, the TSP has proved critical to the efficient probe-testing of MCM substrates (e.g., [50] and others).

Here, we study a six-city TSP instance that is embedded in the Manhattan plane with city coordinates $A = (0,0)$, $B = (100,0)$, $C = (100,200)$, $D = (0,200)$, $E = (40,95)$ and $F = (40,105)$. In this instance, $|S| = 5!/2 = 60$, and there exists one globally optimal solution (ABCDEF(A)), along with three other locally optimal solutions (ABFECD(A)), (ABECDF(A)), and (ABFCDE(A)). We use the Lin 2-opt neighborhood operator that is usual in studies of the TSP [28]: a 2-opt move deletes two edges of the current solution s_i and

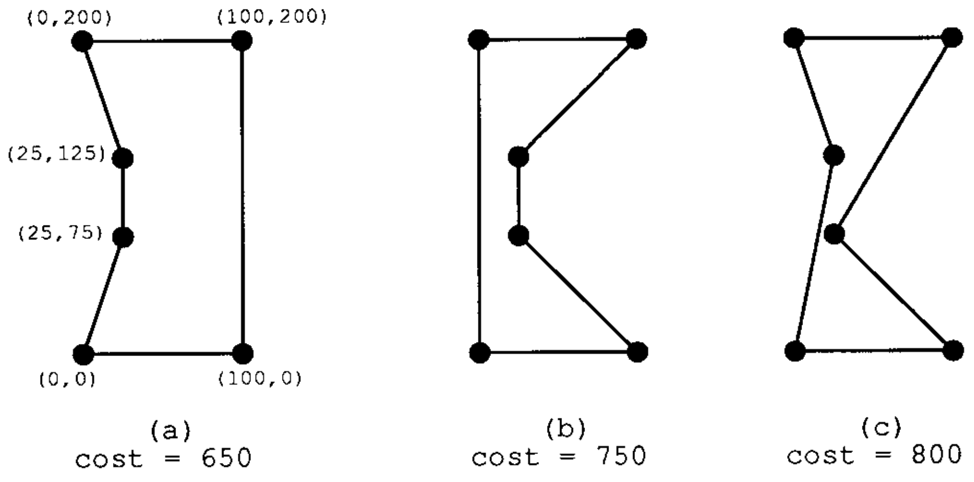


Figure 8: Traveling Salesman Problem (TSP) instance with 6 cities. The global minimum (a) and two local minima are depicted. The remaining local minimum is symmetric to (c), and is not shown.

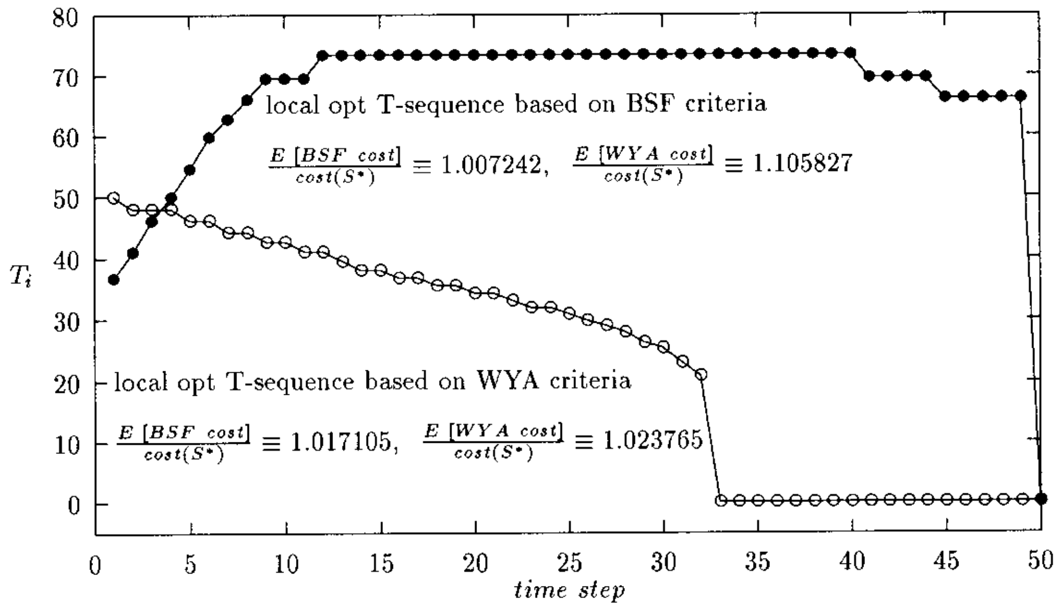


Figure 9: Locally optimal annealing schedules for 6-city TSP example, determined by BSF and WYA measures. $S*$ denotes the optimal TSP solution.

then reconnects the two resulting paths into the "other" tour. The neighborhood size is $|N| = C(6,2) - 6 = 9$ in a six-city instance: we consider swaps of all pairs of edges except for adjacent pairs, since these will yield no change in the tour. For this example, we have again solved numerically for locally optimum annealing schedules of all lengths up to $M = 50$, using both the BSF and WYA criteria. Again, the results shown in Figure 9 provide a stark contrast: the optimum 50-step WYA schedule is monotone decreasing (again, as would be expected from the body of results in the current literature), while the optimum 50-step BSF schedule is nearly *monotone increasing.* As with all of our previous examples, even though the optimum BSF is hopeless when measured by the WYA objective (expectation of 1.1058 versus 1.0238 times optimal), it is clearly superior to the optimum WYA schedule (1.0072 versus 1.0171 times optimal) when judged by the "real-life" BSF criterion.

	Number of Steps	Expected BSF Quality	Expected WYA Quality	Schedule									
Optimize BSF	5	1.153	1.154	0	8	12	15	0					
	10	1.078	1.090	16	18	20	22	25	28	32	35	38	0
	15	1.049	1.085	23	25	29	33	37	39	43	44	46	48
				48	48	50	52	0					
	20	1.034	1.091	27	31	35	39	44	48	50	52	55	55
				57	57	57	57	57	55	57	57	60	0
	25	1.025	1.096	30	34	39	44	48	52	55	57	60	60
				63	63	63	63	63	63	63	63	63	60
				60	60	60	63	0					
	30	1.019	1.100	33	37	41	46	50	55	57	60	63	66
				66	66	66	66	70	66	60	66	66	66
				66	66	66	63	63	63	63	63	63	0
	35	1.015	1.102	34	38	43	48	52	57	60	63	66	66
				70	70	70	70	70	70	70	70	70	70
				70	70	70	70	70	70	70	66	66	66
				63	63	63	66	0					
	40	1.011	1.104	35	39	44	50	55	57	64	66	66	70
				70	70	70	70	70	73	73	73	73	70
				70	73	70	73	70	73	70	73	70	70
				70	70	70	66	66	66	66	63	66	0
	50	1.007	1.106	37	41	46	50	55	60	63	66	70	70
				70	73	73	73	73	73	73	73	73	73
				73	73	73	73	73	73	73	73	73	73
				73	73	73	73	73	73	73	73	73	73
				70	70	70	70	66	66	66	66	66	0
Optimize WYA	5	1.154	1.154	0	0	0	0	0					
	10	1.079	1.079	14	12	10	0	0	0	0	0	0	0
	15	1.052	1.053	22	20	19	16	0	0	0	0	0	0
				0	0	0	0	0					
	20	1.040	1.041	33	32	31	29	27	24	20	0	0	0
				0	0	0	0	0	0	0	0	0	0
	25	1.033	1.035	41	41	39	38	37	34	32	30	28	25
				20	0	0	0	0	0	0	0	0	0
				0	0	0	0	0					
	30	1.028	1.031	46	44	44	43	41	41	39	37	35	34
				32	30	27	25	21	0	0	0	0	0
				0	0	0	0	0	0	0	0	0	0
	35	1.024	1.028	48	48	46	46	44	43	43	41	39	38
				37	35	34	32	31	29	27	25	0	0
				0	0	0	0	0	0	0	0	0	0
				0	0	0	0	0					
	40	1.021	1.024	50	48	48	48	46	46	44	44	43	43
				34	33	32	32	31	30	29	28	26	25
				23	21	0	0	0	0	0	0	0	0
				0	0	0	0	0	0	0	0	0	0

Table 4: Locally optimal annealing schedules for the 6-city TSP instance.

3.4 A Scaling Model for Large-Scale Optimizations

For all three problem classes above, we have obtained clear differences between the optimal BSF and optimal WYA annealing schedules. However, the significance of these experiments may possibly be limited by scaling effects: the small instances may somehow fail to contain the attributes that make larger instances hard to solve, or the phenomena that we observe might be manifested only when the schedule length M is large relative to the size of the solution space. To help assess whether our results are due to such scaling artifacts, we have also tested the BSF-WYA distinction on a *levels* model which captures the scaling structure of large-scale optimization cost surfaces.

Our levels model abstracts large solution spaces by grouping the solutions into a small number of *classes*, each of which is associated with a particular range of solution costs and/or a particular type of local structure in the cost surface. The levels model captures several key attributes that are common to the various structural models proposed in the literature to describe large-scale optimization cost surfaces. In particular, we have attempted to capture the following statistics that have been observed in actual cost surfaces:

1. Aarts et al. [1] [25] have postulated that the distribution of solutions near the global optimum is distributed exponentially in relation to their distance from the global optimum. In other words, if C_{opt} is the cost of the global optimum, then for some constant $\gamma > 0$, the number of states with cost C, is given by
$$w(C) \propto exp((C - C_{opt})\gamma).$$

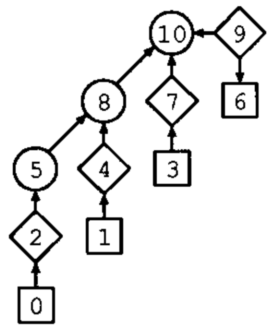
2. Aarts et al. [1] [25] also postulate that the set of solution costs $f(s)$ over the entire solution space S will follow a normal distribution. Since the best simulated annealing schedule will intuitively spend most of its time in the region of relatively good solutions (what we call the "Energy Landscape" below), we may assume that the number of solutions with a given cost will follow an exponential distribution in this region of interest. Note that this assumption is also in agreement with the model proposed by Baum in [4].

3. The model of Baum [4] also implies that near the optimum solution, the numbers of local minima with given solution costs will also follow an exponential distribution. There are perhaps more sophisticated models that can be coerced to yield distributions for the quality of local minima in the cost surface, but we have not yet performed such analyses.[9]

4. A number of authors, ranging from [13] to [44], use the notion of "basins of attraction" to describe the progress of the annealing algorithm within the cost surface. The term "basin" connotes the fact that from any given solution, only a limited subset of the local minima in the solution space may be reached without resorting to uphill moves.

To capture these relative incidences of both solution costs and locally minimum solution costs, our levels model is as exemplified in Figure 10. The figure illustrates the structure of the 5-level version of our model.

[9]In particular, the fractal scaling model of Sorkin [44] (also see [20] and [48]) seems promising, since it has been found to closely match the correlations of real optimization cost surfaces. The autocorrelations implicit in Sorkin's model (which views cost surfaces as high-dimensional fractional Brownian motions) can yield distributions of local minimum solution costs in the region of the global optimum solution. Kirkpatrick and Toulouse [24] give evidence that local optima of combinatorial problems (specifically, the TSP) are embedded in an ultrametric space under a very natural distance function. By results of Baldi and Baum [3], the ultrametricity of local optima can imply very tight bounds on the number of local minimum solutions that can exist in the cost surface. However, the ultrametric assumption does not easily yield a distribution on the *costs* of these local minima.



cost	total number of solutions	□ number of local minima	◇ number of closest neighbors
4	α^4	β^4	$(\alpha-\beta)\beta^3$
3	α^3	β^3	$(\alpha-\beta)\beta^2$
2	α^2	β^2	$(\alpha-\beta)\beta$
1	α	β	$(\alpha-\beta)$
0	1	1	0

Figure 10: Levels model with five different levels. Each square, diamond, or circle represents a class of interchangeable (i.e., identical) states. The squares represent local minima; the diamonds represent solutions that are within a "basin of attraction", and the circles represent solutions that are neither locally minimum nor within any basin of attraction. Besides α and β, a further parameter, j, determines neighborhood structure by limiting the span of any single uphill move. Two classes can be neighbors only if there is a directed path of length j or less between them. Note that this includes directed paths of length zero, i.e., self-loops in the solution space.

Each solution in the solution space has an integer cost, corresponding to its *level*, lying in the range 0 to 4. In the model there are eleven classes of solutions; all solutions in a given class have identical properties, and induce only a single column and row in the (11×11) transition matrices that are used in our methodology of Section 2.2 above. The details of any instance of the model are determined by three parameters α, β, and j:

- α determines the number of solutions in a level i: if S_i is the set of solutions at level i, then $|S_i| = \alpha^i$.
- $\beta < \alpha$ determines the number of local minima (shown as squares in Figure 10), $|LM_i|$, at level i. The parameter β also determines the number $|CN_i|$ of solutions at level i (denoted by diamonds in the Figure) which are "closest neighbors" to local minima at level $i-1$. $|LM_i| = \beta^i$, and $|CN_i| = \beta^{i-1}(|S_{i-1}| - |LM_{i-1}|)$.
- j limits the size of a "jump" that can be made in a single step of the algorithm. To be specific: two classes are neighbors exactly when there is a directed path of length j or less between them. This includes directed paths of length zero, so that self-loops in the solution space are allowed.

As an example, if $j = 3$, then groups 0 and 8 in Figure 10 are neighbors, but groups 6 and 10 are not. Again, we point out that our model allows "basins of attraction" for each local minimum: note that any path of moves from a local minimum or its "closest neighbor" class must make a disimproving move in order to reach a group in a different column (i.e., basin of attraction).

To establish the transition probabilities between states (classes) in our model, we use the notion of a *neighbor selection matrix*, denoted by R, which is the transition matrix for temperature $T = +\infty$. Our aim is to enforce the bidirectional nature of all adjacencies in the neighborhood structure when the annealing algorithm is executed. To this end, if we let $|c_i|$ denote the number of solutions in a class c_i, and use $N(c_i)$ to denote the set of classes that are neighbors of c_i,

$$R_{ik} * |c_k| = R_{ki} * |c_i|,\ \forall c_k \in N(c_i)$$

(Note that this is a very reasonable way of enforcing bidirectional transitions; it also holds for any neighborhood structure with constant neigborhood size $|N|$ that can be described by an undirected graph.) We complete the neighbor selection matrix by defining $N(c_i)^+$ to be the neighbors of group c_i with cost greater than or equal to the cost of c_i, and similarly defining $N(c_i)^-$ to be the set of neighbors of c_i with lower cost. The neighbor selection matrix R is then completed by using the balance constraint above and the following recursive formula:

$$\forall k \in N(c_i)^+,\ R_{ik} = \frac{(1 - \sum_{l \in N(c_i)^-} R_{il})}{|N(c_i)^+|}$$

In other words, the possible transitions from c_i that are not "used up" by previously defined transitions to $N(c_i)^-$ are evenly divided among groups of solutions in $N(c_i)^+$.

Table 5 shows the locally optimal BSF and WYA schedules for our levels model, computed with respect to various combinations of the parameters α and β, and using $j = 3$ always. These optimal schedules reflect the "regimes" observed for the combinatorial problems in the three previous subsections: for example, we can readily identify highly periodic (or iterated descent-like) BSF solutions, as well as a BSF schedule that appears to be similar to "warming". The BSF schedules become more random as M increases, suggesting that the best marginal use of the extra steps (in terms of optimizing BSF solution quality!) is attained by simply "wandering" around the cost surface. In contrast, WYA schedules are very conservative, since they are constrained by the desire to end up in a good solution at the M^{th} time step. As with the examples of the previous subsections, we observe marked differences in BSF quality between the BSF-optimal and WYA-optimal schedules. Ongoing work is aimed at refining the levels model and calculation of longer BSF- and WYA-optimal schedules for a greater number of levels (i.e., a more "fine-grain" model).

3.5 Extensions To A Real-Scale Placement Instance

At this point, we can only point to very limited computational experience, mostly because detailed experimentation with annealing on real-scale benchmarks requires inordinate CPU resources. For example, we have implemented various annealing schedules for placement of the small ILLIAC IV benchmark circuit IC67, which has 67 modules. For our experiments, we used the sum of net bounding-box semiperimeters as the objective function, and placed the modules into a fixed (square with three extra slots) array of slots.

α	β	Optimality Criterion	Number of Steps	Expected BSF Quality	Expected WYA Quality	Schedule									
2.0	1.2	BSF	40	.485	1.418	0	0	0	0	0	0	0	3.6	20	20
						20	20	20	0	20	0	20	0	20	0
						20	0	20	0	20	0	20	1.6	20	20
						20	20	0	0	0	0	0	0	0	0
2.0	1.2		60	.279	1.406	0	0	0	0	0	0	0	2.6	3.4	20
						20	20	20	0	20	0	20	0	20	1
						1.6	20	0	20	0	20	0	20	0	20
						0	20	0	20	0	20	0	20	0	20
						0	20	0	20	0	2.2	0	20	20	20
						20	20	0	0	0	0	0	0	0	0
		WYA	40	.704	.835	0	0	0	0	0	1	1	1	1	1
						1	1	.8	1	.8	1	.8	1	.8	1
						.8	1	.8	1.8	0	0	0	0	0	0
						0	0	0	0	0	0	0	0	0	0
			60	.522	.705	0	0	0	0	0	1	1	1	1	1
						1	1	.8	1	.8	1	.8	1	.8	.8
						.8	.8	.8	.8	.8	.8	.8	.8	.8	.8
						.8	.8	.8	.8	.8	.8	.8	.8	.8	.6
						.8	.8	.8	.8	.8	.6	0	0	0	0
						0	0	0	0	0	0	0	0	0	0
2.0	1.5	BSF	40	.865	1.826	0	0	0	0	20	0	20	0	20	20
						20	0	20	0	20	20	20	20	0	20
						0	20	0	20	0	20	0	20	0	20
						20	20	20	20	0	0	0	0	0	0
			60	.706	1.886	0	0	0	0	20	0	20	0	20	0
						20	0	20	0	20	0	20	0	20	0
						20	0	20	0	20	0	20	0	20	0
						20	0	20	0	20	0	20	0	20	0
						20	0	20	0	20	0	20	0	20	0
						20	0	20	20	20	0	0	0	0	0
		WYA	40	.999	1.167	0	0	0	0	20	0	0	0	20	0
						0	0	2.8	0	0	0	20	0	0	0
						0	0	0	0	9.4	0	0	0	0	0
						0	0	0	0	0	0	0	0	0	0
		WYA	60	.876	1.089	0	0	0	0	20	0	0	0	20	0
						0	0	4	0	0	0	4.6	0	0	0
						3.8	0	0	0	2.6	0	0	0	3.2	0
						0	0	2	0	0	0	3	0	0	0
						1.4	0	0	0	2.4	0	0	0	0	0
						0	0	0	0	0	0	0	0	0	0
2.5	1.2	BSF	40	.534	1.058	0	0	0	0	0	0	0	0	0	0
						0	0	0	0	.6	1.4	9.6	20	20	20
						20	20	20	0	0	0	0	0	0	0
						0	0	0	0	0	0	0	0	0	0
		WYA	40	.674	.723	0	0	0	0	0	0	0	.6	.6	.8
						.8	.8	.8	.8	.8	.8	.8	.8	.6	.4
						0	0	0	0	0	0	0	0	0	0
						0	0	0	0	0	0	0	0	0	0
2.5	1.5	BSF	40	.703	1.377	0	0	0	0	0	0	0	0	0	0
						0	0	.6	1.2	2.4	11.4	20	20	20	8
						8.2	20	20	20	20	20	0	0	0	0
						0	0	0	0	0	0	0	0	0	0
		WYA	40	.877	.952	0	0	0	0	0	0	0	.6	.8	.8
						.8	.8	.8	.8	.8	.8	.8	.8	.8	.8
						.8	.6	.4	0	0	0	0	0	0	0
						0	0	0	0	0	0	0	0	0	0
2.5	2.0	BSF	40	1.158	2.070	0	0	0	0	0	0	0	0	0	20
						0	20	0	20	20	20	20	20	20	0
						0	0	0	0	0	20	0	20	20	20
						20	20	0	0	0	0	0	0	0	0
		WYA	40	1.275	1.365	0	0	0	0	3.2	0	0	0	20	0
						0	0	0	0	0	0	20	0	0	0
						0	0	0	0	0	0	0	0	0	0
						0	0	0	0	0	0	0	0	0	0

Table 5: Locally optimal annealing schedules, computed with respect to our levels model in Figure 6. Note that 20 is the highest temperature allowed and may be assumed to represent a temperature of $+\infty$.

Our first trials compared over 50 different linear and logarithmic temperature schedules with M ranging from 10,000 to 100,000. Each schedule was parameterized by either a T_0 value (for logarithmic cooling) or by a (T_0, T_M) pair (for linear schedules), and was run from 15 random starting solutions. Surprisingly, careful study showed that linear cooling schedules were clearly better than logarithmic cooling for this benchmark. Also surprising were the degree of variation in solution quality with the parameterization of the schedule, and the fact that the best (linear) schedule for all M values consistently started with approximately the same temperature, and cooled to $T_M = 0$. Warming schedules and constant-temperature schedules were generally not competitive with the linear cooling ones. It is not clear why our results for small examples were not substantiated by this medium-sized example; we suspect that the lengths of our schedules are still so short that essentially greedy methods are still optimal, but there may also be a fundamental difference between our small examples and larger problems. In any case, what kinds of schedules are most efficient using the best-so-far criterion for real problems remains an important open question.

It should be noted that even with the IC67 benchmark we have found some signs of promise in the BSF perspective. For example, the "horizontal offset" between BSF quality of optimal WYA and BSF schedules suggests that a BSF-motivated annealing schedule will reach better solutions faster than traditional cooling. Another intuition is that the WYA objective of minimizing $f(s_M)$ forces WYA schedules to be far too conservative (cf. the results pertaining to the levels model in Section 3.4). Thus, even a simple non-monotone heuristic, e.g., which invokes iterated-descent methods after taking the time to ensure a visit to at least one local optimum, may be more successful than generic cooling. With this in mind, we tested periodic schedules with all $T_i \in \{0, \infty\}$ which were designed in a hierarchical fashion: given a "period" N and a constant factor k, we use $T_i = 0$ as long as i is not divisible by N. If i is divisible by N, then we execute one move at temperature $+\infty$. If i is divisible by $k * N$, then we execute two moves at ∞, and in general, if i is divisible by $k^r * N$, we execute $r + 1$ consecutive moves at $+\infty$.

Our experiments showed these hierarchical/periodic schedules also to be unsuccessful versus linear cooling, exept when a "hybrid" methodology was applied: first linear cooling was run for 85% of the schedule and then a periodic schedule for the remaining portion. Our motivation for such a schedule was that since linear cooling is so conservative, the last part of its schedule might be profitably used to search in the area of the BSF solution enough to visit as many other local minima as possible. In other words, we investigated the simplest consequence of the BSF criterion: it is unnecessary to quench at the end of a schedule to minimize the cost of the very last state visited. In this set of experiments, small improvements over the best cooling schedules were obtained (however, recall that these cooling schedules were the best over a large number of schedules examined).

4 Conclusions

At this stage in our work, many new questions are opened up even as we try to address any single issue. The contributions of this work have been the demonstration that best-so-far analysis points to completely new hill-climbing regimes for further investigation, and the opening of an entirely new theoretical front (e.g., with respect to Markov analysis of best-so-far annealing). On the other hand, the main unanswered question is whether knowing that non-cooling schedules are better than cooling schedules will ever yield practical speedups or performance improvements to large-scale hill-climbing optimizations.

For the examples that we have tested, the best-so-far criterion has led to dramatic changes in our view of how the SA algorithm should be applied. We have found that the best temperature schedules are no longer monotone cooling, but rather periodic or even warming; moreover, this is such a pervasive phenomenon that we are literally forced to discard the original physical "annealing" analogy. The results of Section 3 point to a number of possible hill-climbing "regimes" which may be closely tied to measures of "reachability" within the neighborhood structure (recall the qualitative difference between optimal BSF schedules for the 8- and 16-node bisection instances). We also believe that adaptive methods, and methods that are tuned to statistical parameters of optimization cost surfaces [44] [20] will provide important research directions.

References

[1] E. H. L. Aarts, J. H. M. Korst and P. J. M. van Laarhoven, "Quantitative Analysis of the Statistical Cooling Algorithm", submitted to *Philips J. of Research*, 1987.

[2] E. H. L. Aarts and J. Korst. *Simulated Annealing and Boltzmann Machines: a Stochastic Approach to Combinatorial Optimization and Neural Computing*, Wiley, 1989.

[3] P. Baldi and E. B. Baum, "Bounds on the Size of Ultrametric Structures", *Physical Review Letters* 56(15) (1986), pp. 1598-1600.

[4] E. B. Baum, "Iterated Descent: A Better Algorithm for Local Search in Combinatorial Optimization Problems", *Proc. Neural Information Processing Systems* , D. Touretzky, ed., 1987.

[5] T. N. Bui, S. Chauduri, F. T. Leighton, and M. Sipser, "Graph Bisection Algorithms with Good Average Case Behavior", *Combinatorica* 7(2):171-191, 1987.

[6] V. Cerny, "Thermodynamical Approach to the Traveling Salesman Problem: an Efficient Simulation Algorithm", *J. Optimization Theory and Applications* 45(1) (1985), pp. 41-51.

[7] C.M Fiduccia and R.M. Mattheyses, "A Linear Time Heuristic for Improving Network Partitions", *ACM/IEEE Design Automation Conf.*, 1982, pp. 175-181.

[8] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, New York: W. H. Freeman, 1979.

[9] F. Glover, "Tabu Search - Part I", *ORSA J. Computing* 1 (1989), pp. 190-206.

[10] J. W. Greene and K. J. Supowit, "Simulated Annealing Without Rejected Moves", *Proc. IEEE Intl. Conf. on Computer-Aided Design*, 1984, pp. 658-663.

[11] L. K. Grover, "A New Simulated Annealing Algorithm for Standard Cell Placement", *Proc. IEEE Intl. Conf. on Computer-Aided Design*, November 1986, pp. 378-380.

[12] L. K. Grover, "Standard Cell Placement Using Simulated Sintering", *Proc. ACM/IEEE Design Automation Conf.*, 1987, pp. 56-59.

[13] B. Hajek, "Cooling Schedules for Optimal Annealing", *Math. Oper. Res.* , submitted, 1985.

[14] B. Hajek and G. Sasaki, "Simulated Annealing - To Cool or Not", *Systems and Control Letters* 12 (1989), pp. 443-447.

[15] M. Jerrum and A. Sinclair, "Conductance and the Rapid Mixing Property for Markov Chains: the Approximation of the Permanent Resolved", *Proc. ACM Symp. on Theory of Computing*, May 1988, pp. 235-244.

[16] D. S. Johnson, C. H. Papadimitriou and M. Yannakakis, "How Easy is Local Search?", *J. Computer and Systems Science* 37(1) (1988), pp. 79-100.

[17] D. S. Johnson, C. R. Aragon, L. A. McGeoch and C. Schevon, "Optimization by Simulated Annealing: an Experimental Evaluation; Part I, Graph Partitioning", *Operations Research* 37 (1989), pp.865-892.

[18] D. S. Johnson, C. R. Aragon, L. A. McGeoch and C. Schevon, "Optimization by Simulated Annealing: an Experimental Evaluation; Part III, The Traveling Salesman Problem", *Operations Research*, to appear.

[19] D. S. Johnson, "Local Optimization and the Traveling Salesman Problem" *Proc. 17th Intl. Colloquium on Automata, Languages and Programming*, 1990, pp. 446-460.

[20] A. B. Kahng and G. Robins, "On Structure and Randomness in Practical Optimization", *UCLA Computer Science Department Annual*, 1990, pp. 23-38.

[21] S. Kauffman and S. Levin. "Towards a General Theory of Adaptive Walks on Rugged Landscapes". *Journal of Theoretical Biology*, 128:11–45, 1987.

[22] B. W. Kernighan and S. Lin, "An Efficient Heuristic Procedure for Partitioning Graphs", *Bell System Tech. J.* Feb. 1970, pp. 291-307.

[23] S. Kirkpatrick, Jr. C. D. Gelatt, and M. Vecchi. "Optimization by Simulated Annealing". *Science*, 220(4598):671–680, May 1983.

[24] S. Kirkpatrick and G. Toulouse. "Configuration Space Analysis of Traveling Salesman Problems". *Journal de Physique*, 46:1277–1292, 1985.

[25] P. J. M. Laarhoven and E. H. L. Aarts, *Simulated Annealing : Theory and Applications*, Boston, D. Reidel, 1987.

[26] J. Lam and J. M. Delosme, "Performance of a New Annealing Schedule", *Proc. ACM/IEEE Design Automation Conf.*, 1988, pp. 306-311.

[27] J. B. Lasserre, P. P. Varaiya and J. Walrand, "Simulated Annealing, Random Search, MultiStart or SAD?", *Systems and Control Letters* 8 (1987), pp. 297-301.

[28] E. L. Lawler, J. K. Lenstra, A. Rinnooy-Kan and D. Shmoys, *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*, Chichester: Wiley, 1985.

[29] T. Lengauer, *Combinatorial Algorithms for Integrated Circuit Layout*, Berlin: Wiley-Teubner, 1990.

[30] J. D. Litke, "Minimizing PWB NC Drilling", *Proc. ACM/IEEE Design Automation Conf.*, 1983, pp. 444-447.

[31] M. Lundy and A. Mees, "Convergence of an Annealing Algorithm", *Math. Programming* 34 (1986), pp. 111-124.

[32] M. Mezard and M. A. Virasoro, "The Microstructure of Ultrametricity", *J. Physique* 46 (1985), pp. 1293-1307.

[33] M. Mihail, "Conductance and Convergence of Markov Chains – A Combinatorial Treatment of Expanders", *Proc. IEEE Symp. on Foundations of Computer Science*, November 1989, pp. 526-531.

[34] D. Mitra, F. Romeo and A. Sangiovanni-Vincentelli, "Convergence and Finite-Time Behavior of Simulated Annealing", *Adv. in Applied Probability* 18(1986), pp. 747-771.

[35] J. Nulton, P. Salamon, B. Andresen and A. Qi, "Quasistatic Processes as Step Equilibrations", *J. Chem. Physics* 83(1) (1985), pp. 334-338.

[36] J. Pearl, *Heuristics: Intelligent Search Strategies for Computer Problem Solving*, Reading, MA: Addison-Wesley, 1984.

[37] F. Romeo and A. Sangiovanni-Vincentelli, "Probabilistic Hill Climbing Algorithms: Properties and Applications", *Proc. Chapel Hill Conf. on VLSI* , 1985, pp. 393-417.

[38] J. Rose and W. Klebsch, "Temperature Measurement and Equilibrium Dynamics of Simulated Annealing Placements", *IEEE Trans. on CAD* 9(2) (1990), pp. 253-259.

[39] P. Salamon, K. H. Hoffmann, J. R. Harland and J. D. Nulton, "An Information Theoretic Bound on the Performance of Simulated Annealing Algorithms", *draft*, 1988.

[40] F. Schoen, "Stochastic Techniques for Global Optimization: A Survey of Recent Advances", *J. Global Optimization* 1 (1991), pp. 207-228.

[41] C. Sechen and A. Sangiovanni-Vincentelli, "The Timberwolf Placement and Routing Package", *IEEE J. of Solid-State Circuits* 20(2) (1985), pp. 510-522.

[42] P. Sibani, J. M. Pedersen, K. H. Horrmann and P. Salamon, "Monte Carlo Dynamics of Optimization Problems: A Scaling Description", *draft* , May 1990.

[43] A. Sinclair and M. Jerrum, "Approximate Counting, Uniform Generation and Rapidly Mixing Markov Chains", *Information and Computation* 82 (1989), pp. 93-133.

[44] G. B. Sorkin, "Efficient Simulated Annealing on Fractal Energy Landscapes", *Algorithmica* 6 (1991), pp. 367-418.

[45] G. B. Sorkin, "Simulated Annealing on Fractals: Theoretical Analysis and Relevance for Combinatorial Optimization", *Proc. Sixth MIT Conf. on Adv. Research in VLSI* , March 1990, pp. 331-351.

[46] P. Strenski and S. Kirkpatrick. "Analysis of Finite Length Annealing Schedules". *Algorithmica* 6 (1991), pages 346-366.

[47] L. Tao and Y. C. Zhao, "Multi-Way Graph Partition by Stochastic Probe", *technical report* CSD-91-01, Concordia University, December 1991.

[48] E. Weinberger, "Measuring Correlations in Energy Landscapes and Why It Matters", *Information Dynamics*, H. Atmanspacher and H. Scheingraber, eds., New York: Plenum, 1991, pp. 185-193.

[49] D. F. Wong, H. W. Leong and C. L. Liu, *Simulated Annealing for VLSI Design*, Boston: Kluwer Academic, 1988.

[50] S. Z. Yao, C. K. Cheng, N. C. Chou and T. C. Hu, "An Optimal Probe Testing Algorithm for the Connectivity Verification of MCM Substrates", *Proc. IEEE Intl. Conf. on Computer-Aided Design*, Santa Clara, November 1992.