This paper appears in the *Proceedings of the Fourteenth National Conference on Artificial Intelligence (AAAI-97)*, Providence, RI, 1997.

Evidence for Invariants in Local Search

David McAllester, Bart Selman, and **Henry Kautz**
AT&T Laboratories
600 Mountain Avenue
Murray Hill, NJ 07974
{dmac, selman, kautz}@research.att.com

Abstract

It is well known that the performance of a stochastic local search procedure depends upon the setting of its noise parameter, and that the optimal setting varies with the problem distribution. It is therefore desirable to develop general priniciples for tuning the procedures. We present two statistical measures of the local search process that allow one to quickly find the optimal noise settings. These properties are independent of the fine details of the local search strategies, and appear to be relatively independent of the structure of the problem domains. We applied these principles to the problem of evaluating new search heuristics, and discovered two promising new strategies.

Introduction

The performance of a stochastic local search procedure critically depends upon the setting of the "noise" parameter that determines the likelihood of escaping from local minima by making non-optimal moves. In simulated annealing (Kirkpatrick *et al.* 1983, Dowsland 1993) this is the temperature; in tabu search (Glover 1986, Glover and Laguna 1993), the tenure (the length of time for which a modified variable is tabu); in GSAT (Selman *et al.* 1992), the random walk parameter; and in WSAT (also called "walksat", Selman *et al.* 1994), the parameter is simply called noise. The optimal noise parameter setting depends both upon characteristics of the problem instances, and on the fine-grained details of the search procedure, which may be influenced by other parameters. It requires considerable effort to find the optimal noise parameter setting for a given problem distribution using trial-and-error. Furthermore, sometimes one is faced with a unique, difficult problem to solve, and therefore cannot tune the noise by solving similar problems. Thus it would be extremely desirable to find a way to set the noise parameter that does not vary with the particular search algorithm or the particular problem instance.

This paper presents empirical evidence that such useful invariants (*i.e.,* properties that hold across strategies and domains) do indeed exist. We first studied six variations of the basic WSAT architecture on a class of hard random problem instances. Based on this study we uncovered two invariants. First, for a given problem class, the "noise level" measured by the *objective function value* (the number of unsatisfied clauses) at the optimal parameter settings was approximately *constant* across solution strategies. We call this the "noise level invariant". We also discovered an even more general principle which shows that the optimal parameter setting is one that approximately minimizes the *ratio* of the objective function's mean to its variance. We will show how this "optimality invariant" can be used to tune the noise parameter for a unique problem instance, without having to first solve that instance or a similiar one. As we will see, the optimal value of the noise parameter for a given strategy can be quickly and accurately estimated by analyzing the statistical properties of several short runs of the strategy.

In order to verify that these invariants are not simply due to special properties of random instances, we then confirmed our findings on highly structured instances from the domains of planning and graph coloring.

The results presented in this paper provide immediate practical guidelines for parameter tuning for WSAT and its variants. We further hypothesize that the same invariants hold across other classes of local search procedures, because the variants of WSAT we considered were in fact based on some of these other procedures: for example, a tabu version, a GSAT-like version, and so on. Confirming this hypothesis will require future work. Our results also suggest that the invariants we observed may hold in general, because the domains we considered were so distinct in other aspects. Along with the presentation of the empirical results we will also discuss intuitive explanations as to why these invariants may hold. The current state of the theory of local search does not allow one to analytically derive the existence of these invariants, and we present the development of such a predictive framework as a challenge to the theory community.

Another practical consequence of our work is that it can be used to help design new local search heuristics. It can be very time-consuming to empirically evaluate a suggested heuristic. Because local heuristics are so sensitive to the setting of their noise parameter, one can only rule out a heuristic if it is tested at its optimal setting. When testing dozens or hundreds of heuristics, however, it is computationally prohibitive to exhaustively test all parameter settings. In our own search for better heuristics, however, the parameter settings determined by the invariants consistently yielded the best performance for each strategy. This allowed us to quickly identify two new heuristics that outperformed all the other variations of WSAT on our test instances.

There have been, of course, previous comparative studies of the performance of different local search algorithms for SAT. For example, Gent and Walsh (1993) compared the performance of an "alphabet soup" of variations of GSAT, concluding that one called "HSAT" could solve random problems most quickly. Our aim here is different: we are less interested in finding the best algorithm for random instances than in finding general principles that reveal whether or not different algorithms are in fact searching the same space in approximately the same manner. Parkes and Walser (1996) studied a modified version of WSAT, using GSAT's minimization function at the "p =50%" noise level. They concluded the original WSAT was superior to the modified version. As we shall see, however, the parameter p has different optimal values for different strategies, and in particular is not optimal at 50% for the modified WSAT. Recent work by Battiti and Protasi (1996) is similar in spirit to the present study, in that they develop a general feedback scheme for tuning the noise parameters of local search SAT algorithms. Their calculation is based on the "mean Hamming distance" the algorithm travels in the tail of the search. By contrast, we believe that the statistical measures we employed (described below) more accurately and clearly reveals the optimal noise settings for a variety of algorithms.

Local Search Procedures for Boolean Satisfiability

We consider algorithms for solving Boolean satisfiability problems in conjunctive-normal form (CNF). A formula is a conjuction of clauses; a clause is a disjuction of literals; and a literal is a propositional variable or its negation. In 3SAT, each clause contains exactly three distinct literals. Clauses in a "random 3SAT" formula are generated by choosing three distinct literals uniformly at random, and then negating each or not with equal probability. Mitchell *et al.* (1992) showed that random 3SAT problems are computationally hard when the ratio of clauses to variables in such formula is approximately 4.3.

A local search procedure moves in a search space where each point is a truth assignment to the given variables. A solution is an assignment in which each clause of the formula evaluates to true. The WSAT procedure begins by considering a random truth assignment. It searches for a solution by repeatedly selecting a violated clause at random, and then employing some heuristic to select a variable in that clause to "flip" (change from true to false or vice-versa).

The objective function that local search for SAT attempts to minimize is the total number of unsatisfied clauses. The characteristic of a search strategy that causes it to make moves that are non-optimal — in the sense that the moves increase or fail to decrease the objective function, even when such improving moves are available in the local neighborhood of the current state — is called noise. As noted earlier, noise allows a local search procedure to escape from local optima. Each heuristic described below takes a parameter that can vary the amount of noise in the search. As we will see, the values assumed by this parameter are not directly comparable across strategies: *e.g.*, a value of 0.4 for one strategy may yield a search with more frequent non-improving moves than the search performed by a different strategy with the same parameter value. In fact, the noise level invariant we will describe later can be simply viewed as a normalized way of measuring noise that *is* comparable across strategies.

We considered six heuristics for selecting a variable from *within* a clause. The first four are variations of known procedures, while the last two were created during this study, and are described here for the first time. They are:

G: With probability p pick any variable, otherwise pick a variable that minimizes the total number of unsatisfied clauses. The value p is the noise parameter, which ranges from 0 to 1.

B: With probability p pick any variable, otherwise pick a variable that minimizes the number of clauses that are true in the current state, but that would become false if the flip were made. In the original description of WSAT, this was called "minimizing breaks". Again p is the noise parameter.

SKC: Like the previous, but never make a random move if one with a break-value of 0 exists. Note that when the break-value is 0, then the move is guaranteed to also improve the objective function. This is the original WSAT strategy proposed by Selman, Kautz, and Cohen (1994).

TABU: The strategy is to pick a variable that minimizes the number of unsatisfied clauses. At each step, however, refuse to flip any variable that had been flipped within the past t steps; if all the variables in the chosen unsatisfied clause are tabu, choose a different unsat clause instead. If all variables in all unsatisfied clauses are tabu, then the tabu list is ignored. The tabu list length t is the noise parameter.

NOVELTY: This strategy sorts the variables by the total number of unsatisfied clauses, as does G, but breaking ties in favor of the least recently flipped variable. Consider the best and second-best variable under this sort. If the best variable is not the most recently flipped variable in the clause, then select it. Otherwise, with probability p select the second-best variable, and with probability $1-p$ select the best variable.

R_NOVELTY: This is the same as NOVELTY, except in the case where the best variable is the most recently flipped one. In this case, let n be the difference in the objective function between the best and second-best variable. (Note that $n \geq 1$.) There are then four cases:

1. When $p < 0.5$ and $n > 1$, pick the best.
2. When $p < 0.5$ and $n = 1$, then with probability $2p$ pick the second-best, otherwise pick the best.
3. When $p \geq 0.5$ and $n = 1$, pick the second best.
4. When $p \geq 0.5$ and $n > 1$, then with probability $2(p - 0.5)$ pick the second-best, otherwise pick the best.

The intuition behind NOVELTY is that one wants to avoid repeatedly flipping the same variable back and forth. The intuition behind R_NOVELTY is that the objective function should influence the choice between the best and second-best variable — a large difference in the objective function favors the best. Note that R_NOVELTY is nearly deterministic. To break deterministic loops in the search, every 100 flips the strategy selects a random variable from the clause. Although few flips involve non-determinism, as we shall see the performance of R_NOVELTY is still quite sensitive to the setting of the parameter p.

The Noise Level Invariant

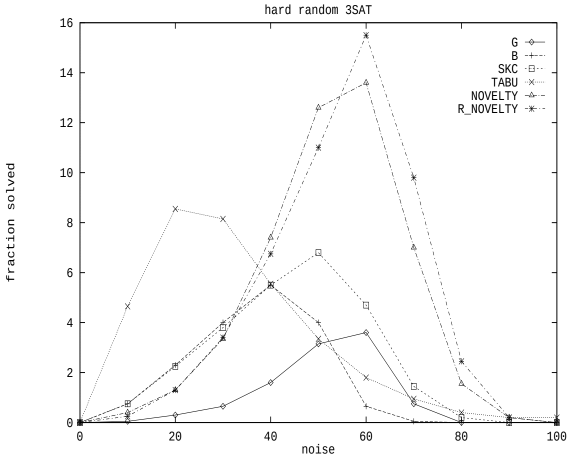

Figure 1: Sensitivity to noise.

As we have discussed, noise in local search can be controlled by a parameter specifying the probability of a locally non-optimal move, as in strategies G, B, and SKC, NOVELTY, and R_NOVELTY. Tabu procedures instead take a parameter specifying the length of the tabu list. Searches with short tabu lists are more susceptible to local minima (*i.e.* are less noisy) than searches with long tabu lists.

In Figure 1 we show the results of a series of runs of the different strategies as a function of the setting of the noise parameter, on a collection of 400 variable hard random 3SAT instances. The horizontal access is the probability of a random move. The tabu length ranged from 0 to 20, and was normalized in the graph to the range of 0 to 100. The vertical axis specifies the percentage of instances solved. Each data point represents 16,000 runs with a different formula each run, where the maximum number of flips per run is fixed at 10,000.

We have plotted the value of the noise parameter versus the fraction of the instances that were solved. For example, R_NOVELTY solved almost 16% of the problem instances when p was set to 60%. Considering the fraction solved with a fixed number of flips allows us to gather accurate statistics on the effectiveness of each strategy. If instead we tried to solve every instance, we would face the problem of dealing with the high variation in the run-time of stochastic procedures — for example, a few runs could require millions of flips, simply by chance — and the problem of dealing with runs that never converged.

As is clear from the figure, the performance of each strategy varies greatly depending on the setting of the noise parameter. For example, running R_NOVELTY at a noise level of 40% instead of 60% degrades its performance by more than 50%. Furthermore, the optimal performance appears at different parameter settings for each strategy. This immediately suggests that in comparing strategies one has to carefully optimize the parameter setting for each, and that even minor changes to a strategy require that the parameters be appropriately re-adjusted.

Given the preceding observation, the question arises: is there a better characterization of the noise level, which is less sensitive to the details of individual strategies? We examined a number of different measures of the behavior of the local search strategies. Let us define the *normalized noise level* of a search procedure on a given problem instance as the the *mean value of the objective function* during a run on that instance. Then we can observe that the *optimal* normalized noise level is approximately *constant* across strategies. This is illustrated in Figure 2. In other words, when the noise parameter is optimally tuned for each strategy, then the mean number of unsatisfied clauses during a run is approximately the same across strategies.

We call this phenomena the *noise level invariant*. It provides a useful tool for designing and tuning local search methods: Once we have determined the mean violation count giving the optimal performance for a single strategy

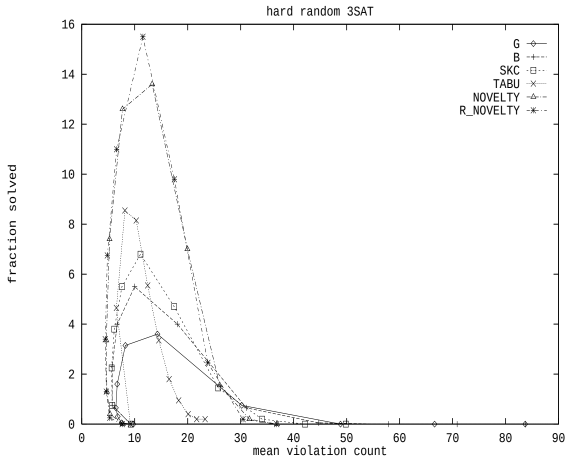

Figure 2: Strategy invariance of normalized noise level on random formulas.

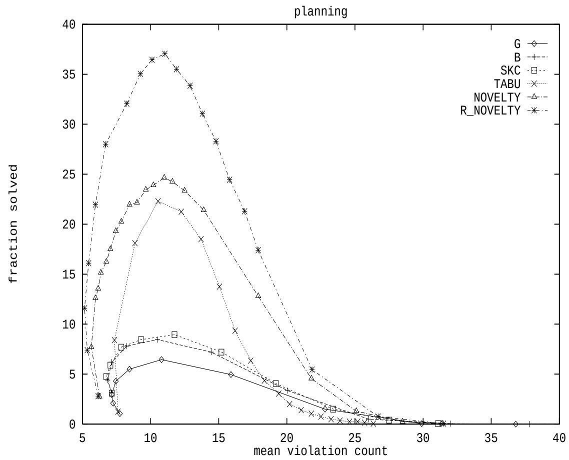

Figure 3: Strategy invariance of normalized noise level on a planning formula.

over a given distribution of problems, we can then simply tune other strategies to run at the same mean violation count, in the knowledge that this will give us close to the optimal performance.

After hypothesizing the existence of this invariant based on our study of random formulas, we wished to see whether it also held for classes of real-world, structured satisfiability problems. Figures 3 and 4 present confirming evidence.

Figure 3 is based on solving a satisfiablity problem that encodes a blocks-world planning problem (instance "bw_large.a", from (Kautz and Selman 1996)). The original problem is to find a 6-step plan that solves a planning problem involving 9 blocks, where each step moves a block (*i.e.*, a pickup followed by a putdown). After the problem is encoded and then simplified by unit propagation, it contains 459 variables and 4675 clauses. Each stochastic procedure was run 16,000 times, with a different random seed for each run, at each data point. (Note that this is unlike the case with the random formulas, where a different formula was generated for each try. Of course, the entire point of this exercise was to test our hypothesis on a real structured problem, not on a collection of randomly-generated instanced. We wanted to make sure that the observed invariant was not simply due to some statistical property of random formulas.)

Figure 4 shows the noise level invariant on a SAT encoding of a graph coloring problem. The instance is based on an 18 coloring of a 125-node graph (Johnson *et al.* 1991). This formula contains 2,250 variables and 70,163 clauses. Because this formula is so large, we could not perform as many runs for each data point as in the previous experiments. Each point is based on just 1,000 samples. The explains the somewhat irregular nature of the curves.

The noise level invariant does not imply that all strategies are equivalent in terms of their optimal performance level. We have informally experimented with a large number of heuristics for selecting the variable to change in the WSAT program for solving Boolean satisfiability problems. The noise level invariant allowed us to quickly evaluate more than 50 variations of WSAT, while being confident that each was tested at its optimal noise level. This led to the development of the NOVELTY and R_NOVELTY strategies, which consistently outperform the other variants, by roughly a factor of two.

The Optimality Invariant

The noise level invariant gives us some handle on dealing with the noise sensitivity of local search procedures. In order to use it, however, one needs to be able to gather statistics on the success rate of at least one strategy across a sample of a given problem distribution. In practice we are often faced with the need to solve a particular novel problem instance. Furthermore, this instance can be extremely hard, and solving it even once may require a large amount of computation even at the (yet unknown) optimal noise setting. What is desirable, therefore, is a way of quickly predicting the setting of the noise parameter for a single problem instance, without actually having to solve it.

Fortunately, our empirical study of noise sensitivity has yielded a preliminary principle for setting noise parameters based on *statistical properties* of the search. More specifically, we make many short runs of the search procedure. We record the final value of the objective function for each run and the variance of the values over that run. We then take the average of these values over the runs.

At low noise levels (running too "cold"), the mean value of the objective function is small — *i.e.*, we are reach-

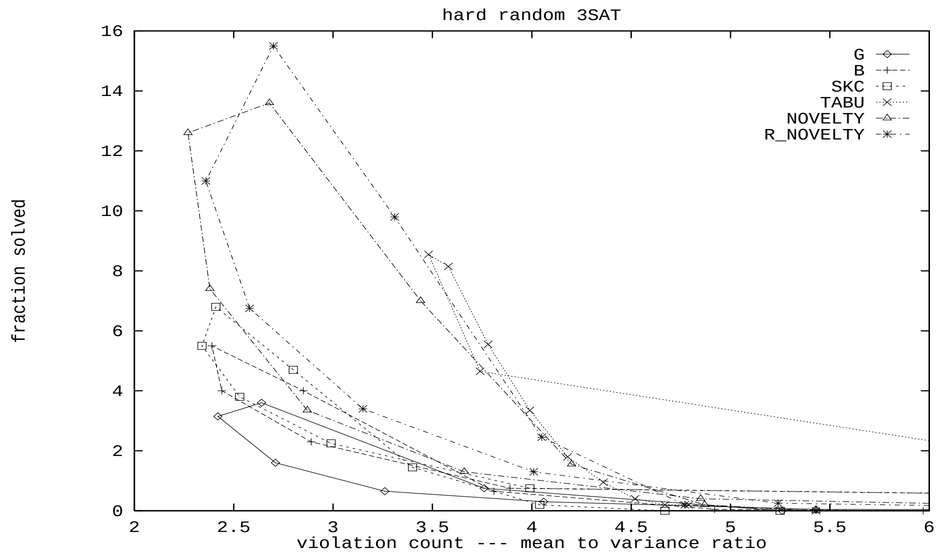

Figure 5: Tuning noise on random instances.

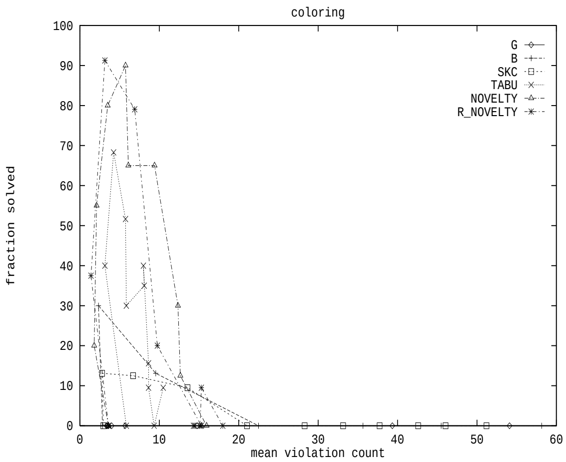

Figure 4: Strategy invariance of normalized noise level on a graph coloring formula.

ing states with low numbers of unsatisfied clauses. However, the variance is also very small; so small, in fact, that the algorithm seldom reaches a state with zero unsatisfied clauses. When this occurs, the algorithm is stuck in a deep local minima. On the other hand, at high noise levels (running too "hot"), the variance is large, but the average number of unsatisfied clauses is even larger. Once again, the algorithm is unlikely to reach a state with zero unsatisfied clauses.

Therefore, we need to find the proper balance between the mean and variance. Our experiments show that the *ratio* of the mean to the variance provides a useful balance. In fact, optimal performance is obtained when the noise value is slightly above that at which the ratio is minimized. We call this observation the *optimality invariant*. Furthermore, this invariant holds for all the variations of WSAT we considered.

To illustrate the principle, In Figure 5 we present the fraction of problems solved as a function of the mean to variance ratio on a collection of hard random problem instances. For each strategy the data points form a loop. Traversing the loop in a clockwise direction starting from the lower right hand corner corresponds to increasing the noise level from 0 to its maximum value. As we see, at some point during this traversal one reaches a minimum value of the mean to variance ratio. For example, the R_NOVELTY strategy (the highest curve) has a minimum mean to variance ratio of around 2.5. At that point it solves about 11% of the instances. By raising the noise somewhat further, and thus increasing again the mean to variance ratio, we reach the peak performance of 15% at a ratio of 2.8. We observe the same pattern for all strategies. In our experiments we found the optimal performance when the ratio is about 10% higher than its minimum value.

Figures 6 and 7 again confirm this observation on the planning and graph coloring instances. Again we see that all the curves reach their peak slightly to the right of the minimal mean to variance ratio.

We should stress again that measuring the mean to variance ratio does not actually require solving the problem instance. We can measure the ratio at each noise value by simply doing several short runs where we compute the mean and the variance of the violation count during the run. Then, by repeating this procedure at different noise parameter set-

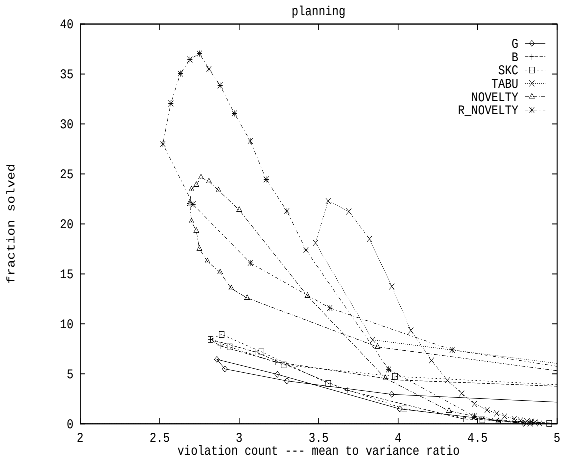

Figure 6: Tuning noise on a planning instance.

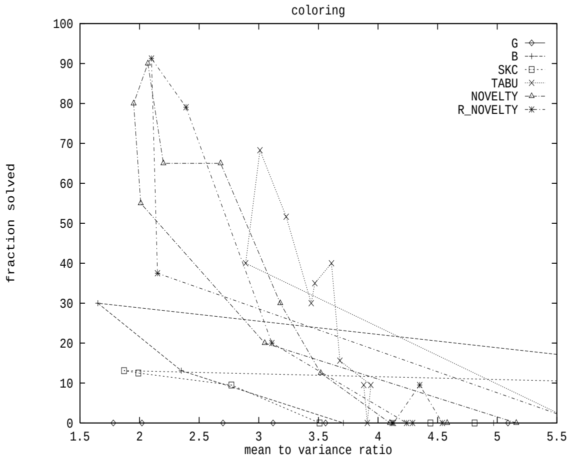

Figure 7: Tuning noise on a graph-coloring instance.

tings we can determine the settings necessary to obtain a mean to variance ratio that is 10% above its minimum. This gives a noise parameter setting at which we can then invest a significant amount of computational effort in order to actually solve the instance.

Another way of solving a unique problem instance is to start a run at an arbitrary noise level. Then, one can measure the mean to variance ratio during the run, and dynamically adjust the noise level to obtain optimal performance. We are currently experimenting with such a *self-tuning* version of WSAT.

Conclusions

We presented two statistical measures of the progress of local search algorithms that allow one to quickly find optimal noise settings. First, we showed that the optimal mean value of the objective function is approximately constant across different local search strategies. Second, we showed that one can optimize the performance of a local search procedure by measuring the ratio of the mean to variance of the objective function during a run. The second measure allows one to find good noise level settings for previously unseen and unsolved problem classes. Finally, we applied these principles to the task of evaluating new local search heuristics, and as a result discovered two new heuristics that significantly outperformed other versions of WSAT on all the test data.

References

Battiti, R., and Protasi, M. (1996). Reactive Search, a history-based heuristic for MAX-SAT. Technical report, Dipartimento di Matematica, Univ. of Trento, Italy.

Connolly, D.T. (1990) An improved annealing scheme for the QAP. *EJOR*, 46:93–100, 1990.

Dowsland, K. A. (1988) Simulated annealing. In *Modern Heuristic Techniques for Combinatorial Problems*, C. R. Reeves (Ed.), John Wiley & Sons, 20–69, 1988.

Gent, I., and Walsh, T. (1993) Towards an understanding of hill-climbing procedures for SAT. *Proc. AAAI-93*, 28–33, 1993.

Glover, F. (1993) Future path for integer programming and links to artificial intelligence. *Computers & Ops. Res.* , 5:533–549, 1993.

Glover, F. and Laguna, M. (1996) Tabu search. In *Modern Heuristic Techniques for Combinatorial Problems* , C. R. Reeves (Ed.), John Wiley & Sons, 70–150, 1996.

Hajek, B. (1988). Cooling schedules for optimal annealing. *Math. Oper. Res.* 13:311–329, 1988.

Johnson, D.S., Aragon, C.R., McGeoch, L.A., and Schevon, C. (1991) Optimization by simulated annealing: an experimental evaluation; part ii, graph coloring and number partioning. *Operations Research*, 39(3):378–406, 1991.

Kautz, H., and Selman, B. (1996). Pushing the envelope: planning, propositional logic, and stochastic search. *Proc. AAAI-96*, Portland, OR, 1996.

Kirkpatrick, S. , Gelatt, C.D., and Vecchi, M.P. (1983) Optimization by simulated annealing. *Science*, 220:671–680, 1983.

Mitchell, D., Selman, B., and Levesque, H.J. (1992). Hard and easy distributions of SAT problems. *Proc. AAAI-92,* San Jose, CA, 459–465, 1992.

Parkes, A.J. and Walser, J.P. (1996) Tuning local search for satisfiability testing. *Proc. AAAI-96*, Portland, OR, 356–362, 1996.

Selman, B. and Kautz, H.A. (1993) An empirical study of greedy local search strategies for satisfiability testing. *Proc. AAAI-93,* Wash., DC, 46–51, 1993.

Selman, B. , Kautz, H., and Cohen, B. (1994). Noise Strategies for Local Search. *Proc. AAAI-94*, Seattle, WA, 337–343, 1994.

Selman, B., Levesque, H.J., and Mitchell, D. (1992) A new method for solving hard satisfiability problems. *Proc. AAAI-92*, San Jose, CA, 440–446, 1992.