Exploiting a Theory of Phase Transitions in Three-Satisfiability Problems

David M. Pennock and **Quentin F. Stout**
Electrical Engineering and Computer Science Department
University of Michigan
Ann Arbor, MI 48109
{dpennock,qstout}@eecs.umich.edu

Abstract

In the past few years there have been several empirical discoveries of phase transitions in constraint satisfaction problems (CSPs), and a growth of interest in the area among the artificial intelligence community. This paper extends a simple analytical theory of phase transitions in three-satisfiability (3-SAT) problems in two directions. First, a more accurate, problem-dependent calculation leads to a new polynomial time probabilistic estimate of the satisfiability of 3-SAT problems called PE-SAT (Probabilistic Estimate SATisfiability algorithm). PE-SAT empirically classifies 3-SAT problems with about 70% accuracy at the hardest region (the so-called crossover point or 50% satisfiable region) of random 3-SAT space. Furthermore, the estimate has a meaningful magnitude such that extreme estimates are much more likely to be correct. Second, the same estimate is used to improve the running time of a backtracking search for a solution to 3-SAT by pruning unlikely branches of the search. The speed-up is achieved at the expense of accuracy—the search remains sound but is no longer complete. The trade-off between speed-up and accuracy is shown to improve as the size of problems increases.

Introduction

Everyone has witnessed phase transitions in the physical sense; for example when the temperature of water rises from below 100°C to above its boiling point.[1] H_2O abruptly transforms from a liquid to a gaseous phase as its temperature crosses this threshold. The defining characteristic of a phase transition is this type of sudden, global change as a particular *global* parameter passes a *critical value*. In this case the global parameter is temperature and the critical value is 100°C.

It is somewhat astonishing that analogous phase transition behavior occurs in abstract man-made structures like graph problems or constraint satisfaction problems (CSPs). In fact, the existence of this type of behavior in random graphs has been known since as early as 1960. However, the empirical discovery of phase transitions in constraint satisfaction and other NP-hard problems, most notably three-satisfiability, is relatively new and has caused an explosion in research on the topic within the artificial intelligence community. The main practical reason for interest in this phenomena lies in the fact that the average-case time complexity for problems near the phase boundary tends to be much worse than for problems away from the boundary.

The logic satisfiability problem (SAT) is the canonical intractable NP-complete problem and involves classifying an arbitrary propositional logic sentence as satisfiable or unsatisfiable. SAT has found countless applications in areas ranging from circuit design to theorem proving. Three-satisfiability (3-SAT) is a subclass of problems where logic sentences are restricted to conjunctions of clauses, each a disjunction of exactly three (different) complemented or uncomplemented literals. Every SAT problem can be cast as a (possibly larger) 3-SAT problem and 3-SAT is also NP-complete. The probability that a random 3-SAT problem is satisfiable has been shown to undergo a sharp phase transition as the ratio of clauses to variables crosses the *critical value* of about 4.2 (Crawford & Auton 1993; Mitchell, Selman, & Levesque 1992). For "large enough" problems[2], below this threshold the probability that a problem is satisfiable is near one while above the threshold the probability that a problem is satisfiable is near zero.

This paper augments a simple analytical theory of phase transitions in random 3-SAT problems in two ways. The simple theory estimates the probability that a random 3-SAT problem is satisfiable by assuming that all clauses are independent. The first extension

[1] Although technically a watched pot never boils, we are assuming the reader is familiar with this particular physical transition.

[2] The sharpness of the phase transition increases as the problem size (number of variables) increases.

is a more accurate calculation that takes into account some of the dependencies between clauses in a particular 3-SAT problem. This probabilistic estimate is used to classify problems as satisfiable or unsatisfiable. The polynomial time procedure, called PE-SAT (Probabilistic Estimate SATisfiability algorithm), empirically achieves about 70% accuracy on problems in the hardest region (at the phase boundary) of random 3-SAT space. The magnitude of the estimate is informative; the higher the estimate the more likely the problem is satisfiable and the lower the estimate the more likely the problem is unsatisfiable. Estimates that are among the 10% most extreme (5% at each extreme) empirically classify problems with about 90% accuracy. The second extension uses the same estimate to improve the speed of a backtracking search for a solution to 3-SAT by pruning branches when the expected number of solutions falls below some threshold. The speed increase is achieved at the expense of accuracy—the search remains sound but is no longer complete. The trade-off between speed-up and accuracy is shown to improve as the size of the problems increase. This extension to backtracking is called PEB-SAT, or Probabilistic Estimate Backtracking SATisfiability algorithm.

A Simple Theory of Phase Transitions in 3-SAT Problems

A phase transition in a constraint satisfaction problem (CSP) is a sharp change in the probability that a problem has a solution; a phase transition can thus only be defined with respect to a specific distribution of problems. Sharp transitions have been observed in many CSPs as some *global* parameter defining the distribution of problems passes a *critical value.* The critical point is often also called the crossover point or the 50% satisfiable point; it is the point where the probability that a legal, satisfying solution exists is 1/2.

For three-satisfiability (3-SAT) a common distribution of problems is the fixed clause length random distribution; given a particular number of variables and clauses any three (different) variables are chosen with equal probability for each clause and each literal is complemented with 50% probability.[3] It has been empirically found that the critical point for random 3-SAT occurs when the ratio of clauses to variables equals about 4.2. More specifically, the authors of (Crawford & Auton 1993) report after extensive testing that the critical point or crossover point for random 3-SAT problems with n variables and m clauses occurs when:

$$m = 4.24n + 6.21 \qquad (1)$$

An important task is to explain this equation with an *analytical* formulation. The first comprehensive explanation of the behavior of phase transitions in 3-SAT problems (and other CSPs) is found in (Williams & Hogg 1994). For the purposes of this paper, however, a simpler formulation similar to that found in (Cheeseman, Kanefsky, & Taylor 1992) and (Kirkpatrick & Selman 1994) will suffice.

Let n denote the number of variables and m denote the number of clauses in a random 3-SAT problem. Let $\vec{\imath} = \{0,1\}^n$ denote an instantiation of all the variables; that is, an n element vector of zeros and ones. Define s to be the probability that a random 3-SAT problem with n variables and m clauses is satisfied by a random instantiation $\vec{\imath}$. Since each clause is a disjunction of three different variables, the probability that any one random clause is satisfied by $\vec{\imath}$ is 7/8. To calculate s, we make an assumption that each clause is independent of the others—this assumption is made purely to simplify the analysis, and is not accompanied by any claim about its accuracy, or to what degree it is an approximation.

$$s = \left(\frac{7}{8}\right)^m$$

This is the probability that the problem is satisfied by a random instantiation $\vec{\imath}$. Now we also make an assumption that each instantiation is independent. Since there are 2^n total possible instantiations, the expected number of solutions, N, is:

$$N = 2^n \left(\frac{7}{8}\right)^m$$

The change in phase occurs when the expected number of solutions crosses from less than one to greater than or equal to one. Solving the equation $N = 1$, we get the location of the phase boundary:

$$m = 5.19n \qquad (2)$$

Next we derive the probability that a random 3-SAT problem is satisfiable, using the same assumptions. The probability that a random problem is *not* satisfied by some random instantiation $\vec{\imath}$ is $1 - (7/8)^m$. Then the probability that the problem is not satisfied for *any* instantiation (i.e. unsatisfiable) is $(1 - (7/8)^m)^{2^n}$. Finally, the probability, S, that a random 3-SAT problem is satisfiable is:

$$S = 1 - \left(1 - \left(\frac{7}{8}\right)^m\right)^{2^n}$$

[3] Technically no two clauses should be exactly the same. Since this happens only rarely, the randomly generated problems used in this paper were not checked for this condition.

We can solve $S = 1/2$ to find the 50% satisfiable point or the crossover point. We can also find the m-intercept at the crossover point by solving $S = 1/2$ when $n = 0$. This leads to:

$$m = 5.19n + 5.19 \quad (3)$$

In the limit Equation 3 is the same as Equation 2. Equation 3 qualitatively matches the empirical relation reported in Equation 1 but the *quantitative* difference is significant enough to indicate that our independence assumptions are too strong. The next section will introduce a more accurate, problem-dependent calculation that takes into account some of the dependencies between clauses.

PE-SAT: A Polynomial Time Probabilistic Estimate for 3-SAT

The above simple calculation of the probability that a 3-SAT problem is satisfiable relied on the simplifying assumption that all clauses are independent. Of course this is not the case; in fact if all clauses were *completely* independent (i.e. had no variables in common) then *every* problem would be satisfiable. By taking into account some of the dependencies between clauses, we can calculate a more accurate estimate of the satisfiability of 3-SAT problems.

Consider a 3-SAT problem with n variables and m clauses. Define a distribution on the variables such that each instantiation $\vec{\imath}$ occurs with equal probability (each variable is zero with probability 1/2 and one with probability 1/2). Let C_i be the event that the ith clause in the sentence is true. Then $C_1 C_2 \cdots C_m$ is the event that all m clauses are true, and $s = \Pr(C_1 C_2 \cdots C_m)$ is the probability that the problem is satisfied (all clauses are true) by a random instantiation $\vec{\imath}$. By the definition of conditional probability,

$$\begin{aligned} s &= \Pr(C_1 C_2 \cdots C_m) \\ &= \prod_{i=1}^{m} \Pr(C_i | C_{i+1} C_{i+2} \cdots C_m) \end{aligned} \quad (4)$$

In order to compress the notation, define $R_i \equiv C_{i+1} C_{i+2} \cdots C_m$ to be the conjunction of the "rest" of the clauses after the ith clause. Each clause C_i is a disjunction of exactly three literals; denote these literals as x_i, y_i, and z_i. Without loss of generality, assume that each of these literals is uncomplemented. Now we can rewrite s as:

$$\begin{aligned} s &= \prod_{i=1}^{m} \Pr(x_i + y_i + z_i | R_i) \\ &= \prod_{i=1}^{m} \Big[\Pr(\bar{x}_i \bar{y}_i z_i | R_i) + \Pr(\bar{x}_i y_i \bar{z}_i | R_i) + \\ &\qquad \Pr(\bar{x}_i y_i z_i | R_i) + \Pr(x_i \bar{y}_i \bar{z}_i | R_i) + \Pr(x_i \bar{y}_i z_i | R_i) + \\ &\qquad \Pr(x_i y_i \bar{z}_i | R_i) + \Pr(x_i y_i z_i | R_i) \Big] \end{aligned}$$

Next we apply Bayes' Rule,

$$s = \prod_{i=1}^{m} \frac{\frac{1}{8}\Pr(R_i | \bar{x}_i \bar{y}_i z_i) + \cdots + \frac{1}{8}\Pr(R_i | x_i y_i z_i)}{\Pr(R_i)}$$

and expand the denominator by cases:

$$s = \prod_{i=1}^{m} \frac{\frac{1}{8}\Pr(R_i | \bar{x}_i \bar{y}_i z_i) + \cdots + \frac{1}{8}\Pr(R_i | x_i y_i z_i)}{\frac{1}{8}\Pr(R_i | \bar{x}_i \bar{y}_i \bar{z}_i) + \cdots + \frac{1}{8}\Pr(R_i | x_i y_i z_i)} \quad (5)$$

The numerator of Equation 5 consists of seven terms and the denominator consists of *the same* seven terms plus one more. Up until this point in our derivation the equations have been exact; now, we make an independence assumption. Every term in Equation 5 is of the form $\Pr(R_i | x_i y_i z_i) = \Pr(C_{i+1} C_{i+2} \cdots C_m | x_i y_i z_i)$. For each of these terms we make the following approximation:

$$\begin{aligned} &\Pr(C_{i+1} C_{i+2} \cdots C_m | x_i y_i z_i) \approx \\ &\qquad \Pr(C_{i+1} | x_i y_i z_i) \cdots \Pr(C_m | x_i y_i z_i) \end{aligned} \quad (6)$$

Each probability term on the right hand side of Equation 6 can be easily evaluated. For example $\Pr(x + a + b | xyz) = 1$ since x is given to be true; similarly $\Pr(\bar{x} + a + b | xyz) = 3/4$ since $\bar{x}$ is given to be false and thus $(\bar{x} + a + b)$ is true if and only if either a or b is true.

Given a particular 3-SAT problem, Equation 5 coupled with the approximation defined in Equation 6 can be evaluated in $O(m^2)$ time. In this way we can calculate a better approximation of s than presented in the previous section. This problem-dependent calculation takes into account first-order dependencies between clauses that were ignored in that section.

We can also find an even more accurate estimate to s by taking into account *second* order dependencies between clauses. This is done by recursively expanding each term in Equation 5 using Equation 4 and making an independence assumption only after this double expansion. This "second-order" estimate can be evaluated in $O(m^3)$ time.

In computer experiments the estimate of s from Equation 5, coupled with the independence assumption defined in Equation 6, was calculated for one thousand random 3-SAT problems at the crossover point for varying size problems. Figure 1 shows the empirical correlation between the estimate for s (the first-order estimate) and the satisfiability of problems with 80 variables at the crossover point. The problems were

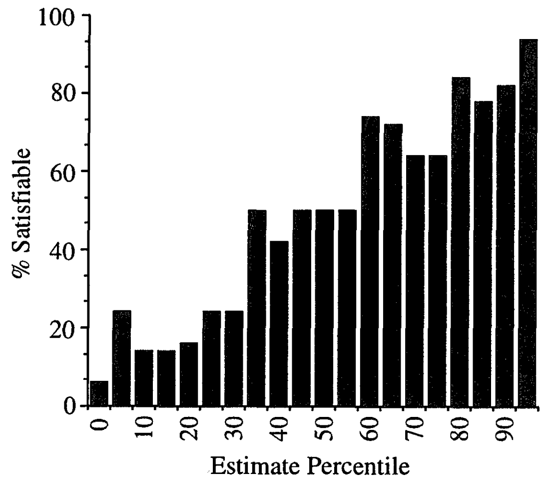

Figure 1: Empirical correlation between the first-order estimate and the satisfiability of 1000 random problems with 80 variables at the crossover point.

sorted according to their estimate values; the graph shows the percentage of problems that are satisfiable in each estimate percentile. Similar correlations occur for other size problems.

The numeric estimate for s can be used to classify problems as satisfiable or unsatisfiable by finding the median of all 1000 estimates at the crossover point and using this as a threshold. Estimates below the threshold are classified as unsatisfiable and estimates above the threshold are classified as satisfiable. It is significant that the threshold can be determined without actually solving the problems *exactly* (an intractable endeavor for large enough problems), as long as the location of the crossover point is known. This first-order estimation procedure is called PE-SAT. The same procedure, carried out using the more accurate, but more computationally expensive, second-order calculation is called $PE^{(2)}$-SAT.

To test the accuracy of PE-SAT and $PE^{(2)}$-SAT, the estimated answer is compared with the exact answer to each problem (satisfiable or unsatisfiable) and found on average to be 68.6% correct for the first order approximation and 72.2% correct for the second order approximation at the crossover point. To show that the magnitude of the estimate is meaningful, the smallest 10% and largest 10% estimates are compared with their exact solutions. These 20% most extreme estimates are found to be on average 85.2% correct for PE-SAT and 90.6% correct for $PE^{(2)}$-SAT. Similarly, the 10% most extreme estimates (5% at each extreme) are on average 87.6% correct for PE-SAT and 92.2% correct for $PE^{(2)}$-SAT. All of the results are summarized in Tables 1 and 2. In our experiments, the accuracy does *not* fall as the size of problems increase.

# of vars	# of clauses	Overall accuracy	20% extreme	10% extreme
30	135	67.1%	84.5%	82.0%
40	175	67.9%	89.5%	92.0%
50	218	69.4%	83.5%	85.0%
60	260	66.7%	82.0%	85.0%
80	345	72.0%	86.5%	94.0%
Totals		68.6%	85.2%	87.6%

Table 1: Accuracy of PE-SAT. 1000 randomly generated problems were tested for each row. "20% extreme" is the accuracy of the 100 smallest and 100 largest estimates. "10% extreme" is the accuracy of the 50 smallest and 50 largest estimates.

# of vars	# of clauses	Overall accuracy	20% extreme	10% extreme
30	135	73.3%	90.0%	93.0%
40	175	71.1%	94.0%	94.0%
50	218	71.6%	89.0%	91.0%
60	260	69.1%	88.5%	91.0%
80	345	75.8%	91.5%	92.0%
Totals		72.2%	90.6%	92.2%

Table 2: Accuracy of $PE^{(2)}$-SAT.

Perhaps the most glaring "missing piece" in the PE-SAT algorithm is a more direct way of deriving the threshold value used for classification—currently a large number of estimates must be calculated and the median used as the threshold. In addition, PE-SAT is only capable of approximating the 3-SAT binary decision problem and does *not* return a satisfying instantiation if one exists. PE-SAT has only been tested on randomly generated 3-SAT problems, and not much can be said about "practical" problems yet. Also note that the most naive algorithm imaginable (always guess "satisfiable") will achieve 50% accuracy at the crossover point.

Improving the Speed of a 3-SAT Backtracking Search

The estimation procedure for 3-SAT discussed above, called PE-SAT, cannot find a legal instantiation of the variables; it can only classify problems as satisfiable or unsatisfiable. This binary decision problem is still NP-complete and is acceptable for some applications of 3-SAT; however many applications require a verifiable solution to be returned, if one exists. Backtracking is one such algorithm and many researchers have investigated varying flavors of backtracking searches to solve 3-SAT problems.

One standard backtracking algorithm for SAT is the Davis-Putnam procedure with unit propagation. This is a recursive procedure that implements a depth-first search through possible variable instantiations until a satisfying assignment is found, or all possibilities are exhausted. Unit propagation is used when a clause contains only one uninstantiated literal and all other literals in the clause have been set to zero. In this case the singleton variable is set to the appropriate value in order to make the literal (and its clause) true. One unit propagation may in turn lead to others. When all possible unit propagations are exhausted, a new variable is chosen to branch on. By choosing variables in most-constraining-first order and values for variables in least-constraining-first order, we can further improve the running time of Davis-Putnam.

By exploiting the same probabilistic estimate used in PE-SAT, we can achieve another improvement in search time, this time at the expense of accuracy. Denote a particular 3-SAT problem as $F(v_1, v_2, \ldots, v_n)$, a propositional logic function of n variables, consisting of a conjunction of clauses, each clause a disjunction of three literals. At any particular level of recursion we have a partial instantiation of the variables $\vec{i_p} = \{0, 1, *\}^n$ where $*$ denotes an uninstantiated variable. Then $F(\vec{i_p})$ is a propositional function, possibly reduced from the top-level function. We can calculate an estimate to s, the probability that $F(\vec{i_p})$ is satisfied by a random instantiation of the remaining (uninstantiated) variables, by using Equation 5 coupled with the independence assumption defined in Equation 6. Let n' be the number of variables still uninstantiated. Then the expected number of legal solutions to $F(\vec{i_p})$ is:

$$N = (2^{n'})s$$

At any particular level of the recursion, if subproblem $F(\vec{i_p})$ has less than one expected solution, it is likely to be unsatisfiable; we can prune the search and back up to the previous choice point with only a small chance of missing a legal solution. In this way we can reduce the search space (improve the running time) at the expense of accuracy. We can vary the trade-off between speed and accuracy by changing the threshold number of expected solutions used to prune the search. For example, if we prune all subproblems $F(\vec{i_p})$ with less than *four* expected solutions (instead of one), we will achieve an even greater speed-up, but have a higher chance of missing all the legal solutions. This extension to backtracking is called PEB-SAT.

Figure 2 shows some empirical results comparing PEB-SAT with the ordinary Davis-Putnam procedure.

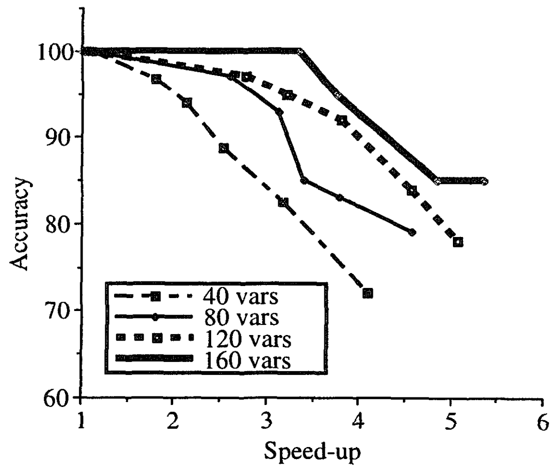

Figure 2: Accuracy versus speed-up trade-off points for PEB-SAT. Speed-up is measured in terms of the number of recursive calls, or the number of nodes traversed in the search.

In both algorithms, unit propagation, most-constraining variable and least-constraining value heuristics are used. The graph plots the speed-up/accuracy trade-off points for randomly generated problems of size 40, 80, 120, and 160 variables at the crossover point. Each point in the graph for 40 variable problems is an average of 1000 problems; each point for 80 and 120 variable problems is an average of 100; each point for 160 variable problems is an average of 20. Each trade-off point for the same size problems is calculated using the same set of random problems. Note that that the trade-off between speed and accuracy improves as the problem size increases. The speed-up factors presented in Figure 2 are in terms of the number of recursive calls, or the number of nodes traversed in the search tree.

PEB-SAT is a sound, but *not* complete solution method for 3-SAT. If PEB-SAT returns "satisfiable", then it is guaranteed to be correct; however if PEB-SAT returns "unsatisfiable", it may be incorrect. This drawback is the same as for a hill-climbing 3-SAT algorithm such as GSAT. It would be instructive to compare the speed-up/accuracy trade-off curves for PEB-SAT and GSAT.

Related Work

In (Huberman & Hogg 1987), the authors predicted that phase transitions would become an important feature of study in many AI systems.[4] Several researchers

[4]A web page at Xerox Parc explores issues relating to phase transitions in CSPs: `ftp://parcftp.xerox.com/pub/dynamics/constraints.html`.

have reported experimental results concerning the behavior of phase transitions in CSPs and other NP-hard problems (Mitchell, Selman, & Levesque 1992; Crawford & Auton 1993; Cheeseman, Kanefsky, & Taylor 1991). Analytical results similar to the "simple" theory presented above, for 3-SAT and graph coloring problems, were derived in (Cheeseman, Kanefsky, & Taylor 1992). Williams and Hogg introduced the first comprehensive theoretical model of phase transitions in CSPs (Williams & Hogg 1994). An accurate analytical model for random K-SAT is presented in (Yugami 1995). Some researchers have made "practical" use of the properties of phase transitions in order to improve search algorithms (Zhang & Pemberton 1994; Hogg & Williams 1994).

An alternate formula for estimating 3-SAT has been derived by Sandholm using empirical methods (Sandholm 1994). This estimate can be computed in linear time, and achieves about 60% accuracy at the crossover region.

Percolation theory involves the study of phase transitions in lattice models of physical systems (Stauffer & Aharony 1994) and insight from this field may shed light on phase transitions in CSPs (Kirkpatrick & Selman 1994). The Bethe lattice or Caylee tree, a well understood structure from percolation theory, may be a good analog to the search tree of a CSP.

Conclusions

We have presented two extensions to a simple analysis of the probability that a random 3-SAT problem is satisfiable. The first is a more accurate probability estimate that takes into account some of the dependencies between clauses in a particular 3-SAT problem. This estimate can be calculated in polynomial time and can be used to classify problems as satisfiable or unsatisfiable. The procedure, called PE-SAT, empirically achieves about 70% accuracy at the crossover point, typically the region containing the hardest problems to solve. Furthermore, the problems that yield the 10% most extreme estimates are classified with about 90% accuracy. The second extension uses the same estimate to heuristically prune a backtracking search for a solution to 3-SAT—the search is pruned when the expected number of solutions falls below some threshold. The algorithm, called PEB-SAT, trades off accuracy for speed; it remains sound but is no longer complete.

A general goal for future research would be to extend these results to encompass larger classes of CSPs and other NP-hard problems. The existence of phase transitions has only recently been uncovered in a wide variety of ubiquitous computer science problems. The question of how exactly to exploit the phenomena for better algorithms seems ripe with potential for discovery and exploration.

References

Cheeseman, P.; Kanefsky, B.; and Taylor, W. M. 1991. Where the really hard problems are. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence*, 331–337.

Cheeseman, P.; Kanefsky, B.; and Taylor, W. M. 1992. Computational complexity and phase transitions. In *Workshop on Physics and Computation*, 63–68.

Crawford, J. M., and Auton, L. D. 1993. Experimental results on the crossover point in satisfiability problems. In *Proceedings of the 11th National Conference on Artificial Intelligence*, 21–27. AAAI Press.

Hogg, T., and Williams, C. P. 1994. Expected gains from parallelizing constraint solving for hard problems. In *Proceedings of the 12th National Conference on Artificial Intelligence*, 331–336. AAAI Press.

Huberman, B. A., and Hogg, T. 1987. Phase transitions in artificial intelligence systems. *Artificial Intelligence* 33:155–171.

Kirkpatrick, S., and Selman, B. 1994. Critical behavior in the satisfiability of random boolean expressions. *Science* 264:1297–1301.

Mitchell, D.; Selman, B.; and Levesque, H. 1992. Hard and easy distributions of sat problems. In *Proceedings of the 10th National Conference on Artificial Intelligence*, 459–465. AAAI Press.

Sandholm, T. W. 1994. A new order parameter for 3SAT. In *Working Notes of the 12th National Conference on Artificial Intelligence Workshop on Experimental Evaluation of Reasoning and Search Methods*, 57–63.

Stauffer, D., and Aharony, A. 1994. *Introduction to Percolation Theory.* London: Taylor & Francis.

Williams, C. P., and Hogg, T. 1994. Exploiting the deep structure of constraint problems. *Artificial Intelligence* 70:73–117.

Yugami, N. 1995. Theoretical analysis of Davis-Putnam procedure and propositional satisfiability. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, 282–288.

Zhang, W., and Pemberton, J. C. 1994. Epsilon-transformation: Exploiting phase transitions to solve combinatorial optimization problems - initial results. In *Proceedings of the 12th National Conference on Artificial Intelligence*, 895–900. AAAI Press.