Critical behavior in the computational cost of satisfiability testing

Bart Selman [a,1], Scott Kirkpatrick [b,c,2]

a *AT&T Bell Laboratories, Murray Hill, NJ 07974, USA*
b *IBM Thomas J. Watson Research Center, Yorktown Heights, NY 10598, USA*
c *Racah Institute of Physics and Center for Neural Computation, Hebrew University, Jerusalem, 91904 Israel*

Abstract

In previous work, we employed finite-size scaling, a method from statistical mechanics, to explore the crossover from the SAT regime of k-SAT, where almost all randomly generated expressions are satisfiable, to the UNSAT regime, where almost all are not. In this work, we extend the experiments to cover critical behavior in the computational cost. We find that the median computational cost takes on a universal form across the transition regime. Finite-size scaling accounts for its dependence on N (the number of variables) and on M (the number of clauses in the k-CNF expression).

We also inquire into the sources of the complexity by studying distributions of computational cost. In the SAT phase we observe an unusually wide range of costs. The median cost increases linearly with N, while the mean is significantly increased over the median by a small fraction of cases in which exponentially large costs are incurred. We show that the large spread in cost of finding assignments is mainly due to the variability of running time of the Davis–Putnam (DP) procedure, used to determine the satisfiability of our expressions. In particular, if we consider a single satisfiable expression and run DP many times, each time randomly relabelling the variables in the expression, the resulting distribution of costs nearly reproduces the distribution of costs encountered by running DP search once on each of many such randomly generated satisfiable expressions. There are intriguing similarities and differences between these effects and kinetic phenomena studied in statistical physics, in glasses and in spin glasses.

Keywords: Dynamical critical phenomena; Phase transition; Finite-size scaling; Satisfiability; k-satisfiability; Computational cost scaling; Complexity

---

[1] E-mail: selman@research.att.com.
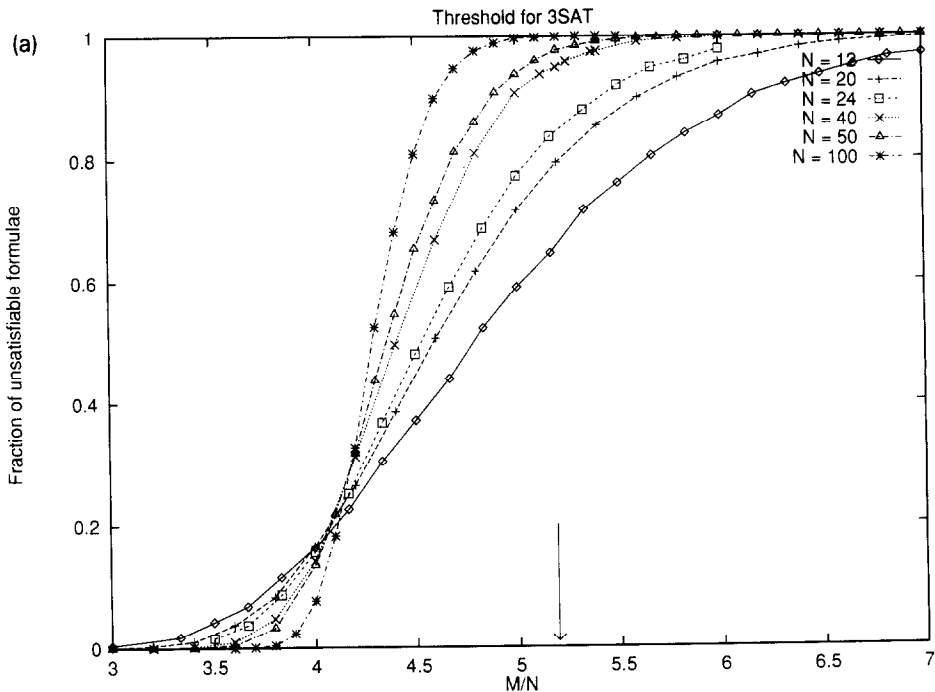[2] Corresponding author. E-mail: kirk@watson.ibm.com.

Fig. 1. (a) Threshold for random 3-SAT. (b) Rescaled data with $\alpha_c = 4.17$ and $\nu = 1.5$. (c) 4-SAT data rescaled with $\alpha_c = 9.25$ and $\nu = 1.25$ (from Kirkpatrick and Selman [16]).

1. Introduction

In [16], we considered threshold phenomena occurring in randomly generated Boolean expressions, and showed that the threshold has characteristics typical of a phase transition in the statistical mechanics of disordered materials. The problem considered is a classic, usually called "k-satisfiability" or k-SAT. (See [22] for some history.) The expressions considered in k-SAT are Boolean formulas, generated at random in conjunctive normal form (CNF), that is as the AND of M "clauses", with each clause the OR of k Boolean variables, and each variable selected at random from a set of N variables. Each variable selected is negated with 50 per cent probability.

The ratio $\alpha = M/N$ determines what fraction of the randomly generated formulas is satisfiable. At low ratios, almost all formulas are satisfiable, whereas at high ratios almost all formulas are unsatisfiable [6,20,22]. Fig. 1(a) shows the threshold function for 3-SAT. The curves are determined empirically. Note that the threshold sharpens up for higher values of N, which is characteristic of threshold phenomena in general.

Threshold functions in combinatorics have been introduced as surfaces in some parameter spaces which separate different behaviors. For example, Bollobás [2] showed that for the random graph ensemble with N vertices and M edges, as long as

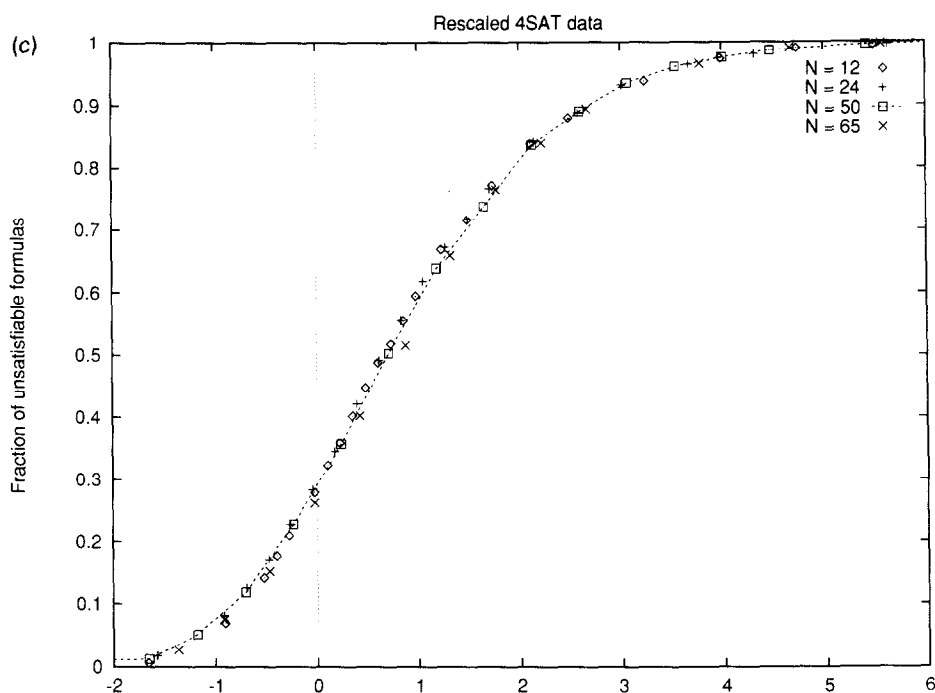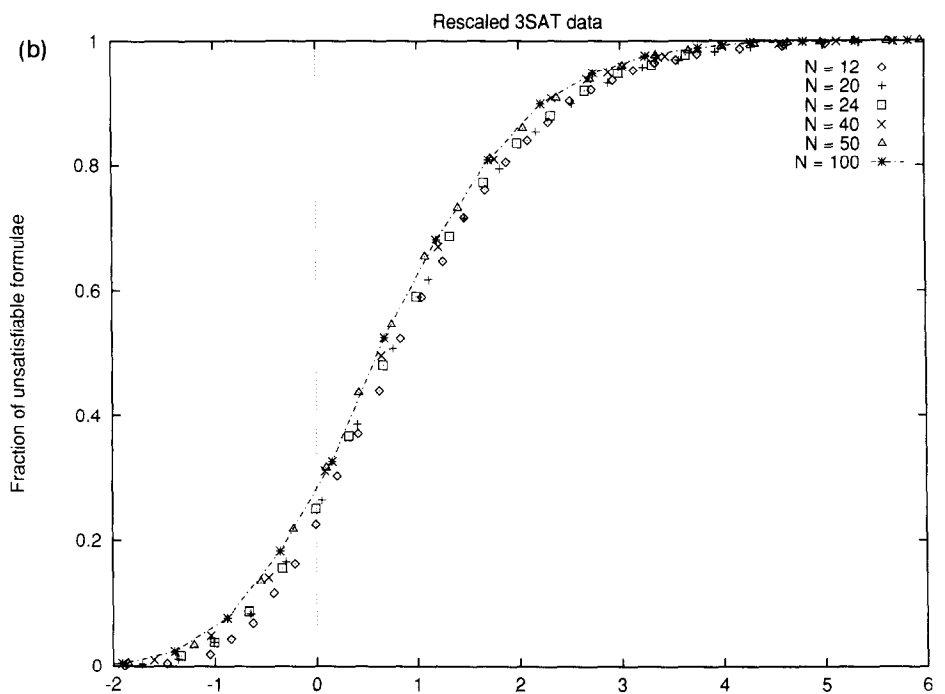$$M - N/2 \quad > \text{ or } < \quad \lambda N^{2/3}$$

Fig. 1 — continued.

for any λ, then almost every graph in the ensemble is ($>$) connected over $O(N)$ vertices, or ($<$) disconnected into finite clusters. The threshold function $\lambda N^{2/3}$ captures precisely the changing scale over which phase-transition-like effects occur as N increases. In terms of the ratio, $\alpha = M/N$, this range narrows as $N^{-1/3}$. In time, λ came to be used to parameterize the critical region itself, by setting $\lambda N^{2/3}$ equal to $M - N/2$. See especially Spencer [25] or Janson et al. [13], who succeeded in calculating many details of distributions of functions induced on the random graph ensemble in terms of λ. Note that we can express λ in terms of the other quantities as

$$\lambda = N^{1/3}(\alpha - 1/2).$$

In the k-SAT problem, N is the number of Boolean variables, and M is the number of clauses (each the OR of k randomly selected variables) in a CNF expression. An ensemble of randomly generated expressions results when $\alpha = M/N$ is held constant. We similarly define a scale-invariant parameter, y, which measures distance from the threshold in a way which becomes independent of N for sufficiently large N, or sufficiently close to the threshold. (See [17], for a derivation.) We plot all nonsingular quantities measured experimentally against

$$y = N^{1/\nu}(\alpha - \alpha_c)/\alpha_c,$$

where ν and α_c are constants. In [16], we used this rescaling approach to identify an invariant function characterizing the crossover from almost always satisfiable to almost always unsatisfiable at finite N. $F(M,N)$, the fraction of expressions which are unsatisfiable, was reduced to a function of y alone. While the invariant function depends in detail on k for $k = 2$, 3 or 4, for larger values of k, $f(y)$ approaches the limiting form (first obtained by Troyansky),

$$f(y) = e^{-2^{-y}}.$$

Figs. 1(b) and 1(c) give the universal form $f(y)$ resulting from rescaling data for 3-SAT and 4-SAT. Note that in our rescaling approach, the exponent ($1/\nu$) and the critical ratio α_c must be determined empirically and are subject to uncertainty about how much of the data we should try to fit, especially when the computations are costly and N cannot be very large. We have normalized by α_c in order to permit the comparison of models with different thresholds, e.g., different k.

The threshold phenomenon in k-SAT is of particular interest because the computationally hardest instances cluster in the transition region. Mitchell et al. [22] consider the median cost of determining satisfiability as a function of the ratio α. They observe an easy–hard–easy pattern. For (random) formulas with α well below or well above the threshold it is relatively easy to determine satisfiability. Around the threshold the cost increases dramatically. Thus, when plotting the median computational cost, one obtains a curve that peaks around the threshold. The peak sharpens up for higher values of N. Mitchell et al. [22] were unable to determine the mean cost accurately, hence their concentration on the median. See [4] for closely related work on random graph problems.

In the first part of this paper, we will show that finite-size scaling can be used to identify a universal form for the median computational cost in the transition region. This means that our rescaling approach again captures the changing scale in the phase-transition-like area in k-SAT, and may thus also be useful in characterizing the dependency on N of other properties at or near the phase transition in combinatorial problems. In the second part of the paper, we will study the distribution of the computational cost in more detail. First, we will show that below the threshold, one can find formulas that appear to be extremely hard. This is consistent with results reported by Hogg and Williams [11] for coloring random graphs and by Gent and Walsh [9] for variable-clause-length random SAT problems. However, we find that the supposedly hard instances are easily solved after a simple renaming of the variables. The apparent hardness of such instances thus appears due to interaction between our systematic backtrack-style satisfiability procedure and our random instances. (See [1] for closely related observations.) In a further experiments, we will show that the distribution of computational costs for a random sample of different formulas is *almost identical* to the distribution of computational costs of running many times on the same formula, each time with its variables randomly relabelled. Our results imply that for any formula below the threshold there are random relabellings that appear to make the formula very hard, and that any two such formulas behave very similarly when solved several times under random relabellings. Given the existence of relabellings that lead to relatively small computational costs, it also follows that there can be a dramatic payoff in the parallel execution of a systematic search procedure on multiple random relabellings of a problem instance. For related work on parallel approaches to solving instances in the phase transition area, see Hogg and Williams [12].

2. Determining satisfiability

We used the Davis–Putnam (DP) procedure [7] to determine the satisfiability of expressions. DP is typical of depth-first recursive search algorithms employed in solving combinatorial problems. The procedure consist of the following steps. Consider an expression consisting of a set of clauses Σ defined over a set of variables V.

- If Σ is empty, return "satisfiable".
- If Σ contains an empty clause, return "unsatisfiable".
- *Unit-Clause Rule.* If Σ contains a unit clause C, assign to the variable mentioned the truth value which satisfies C, and return the result of calling DP on the simplified expression.
- *Splitting Rule.* Select from V a variable v which has not been assigned a truth value. Assign it a value, and call DP on the simplified expression. If this call returns "satisfiable", then return "satisfiable" Otherwise, set v to the opposite value, and return the result of calling DP on the re-simplified expression.

The basic Davis–Putnam procedure is surprisingly efficient. Much work has been done trying to improve the procedure. Most of this work concentrates on finding the best sequence of variables to branch on in the splitting rule, in the hope of reducing the overall search space. From the many alternatives that were explored, only one was

shown to result in a significant improvement of the basic procedure. The idea is to generalize the Unit-Clause Rule, by branching on the variable that occurs the most in the shortest remaining clauses. In two recent extensive studies, this extension of DP proved to be the most efficient sound and complete method to date [3, 14]. Unless indicated otherwise, our data below is all produced with this extended version of DP. The code was developed by Crawford and Auton [6].

Note that this heuristic is similar in spirit to the methods used by Gent and Walsh, and by Hogg and Williams on their respective problems. Also note that there is an asymmetry in the treatment of (satisfiable) SAT and (unsatisfiable) UNSAT expressions. An expression is found to be SAT as soon as one satisfying truth assignment has been found. In the extreme case, this may not require any backtracking. To prove an expression UNSAT, however, requires DP to do an exhaustive exploration of assignments to the first subset of variables large enough to prove that the expression cannot be satisfied. This will, as a rule, involve much more work.

3. Universal form for median cost

In Fig. 2 we present the median cost of determining satisfiability as a function of α for random k-SAT instances ($k = 3, 4, 5$). Each data point is based on a sample of 10,000 formulas. As a measure of the computational cost, we give the number of DP calls, which is the number of times the Splitting Rule was used.

The curves confirm the pattern as reported for 3-SAT in [22]. Note how the curves sharpen up for higher values of N, and costs increase very strongly with N. In Fig. 3, we plot the increase with increasing N (just the maxima). We find that for all three values of k, the max cost increases with N as $e^{(N/c)}$, where $c \approx 11.1$ for $k = 3$; 4.4 for $k = 4$; and 2.9 for $k = 5$. Fig. 3 also includes data for the basic version of DP (without the preference for splitting on variables in short clauses) on 3-SAT formulas. In this case, $c \approx 4.3$. c can be loosely interpreted as the number of bits set by the Unit Clause rule each time a bit is set by assignment in the splitting rule. This confirms that the heuristic used in our extended DP indeed results in a significant reduction in the growth rate of the cost function.

In order to compare our cost curves in Fig. 2 for different values of N, we normalized the curves to have 1 as the maximum cost. See Fig. 4 for the result. Note again how the curves sharpen up with increasing values of N. We also see that the maxima in the curves shift to somewhat lower values of α with increasing N. We now apply our finite-size scaling approach. We replot each set of data against the rescaled parameter, y, which gives us a single universal curve for each value of k, seen in Figs. 5(a), 5(b), and 5(c). The fit is best in the critical region (around $y = 0$), as is to be expected when using a finite-size scaling approach. The k-dependence of the limiting curve is very slight. To obtain the best fits, we have adjusted the values of α_c and ν slightly from the values reported in [16], remaining in each case within the original error bars. The peak in each rescaled curve occurs at $y = 0.9 \pm 0.1$, which is to the right of the critical point ($y = 0$). This is consistent with the often-repeated observation that the hardest cases occur near $f(y) = 0.5$ in our observed crossover functions. From Fig. 1, we see
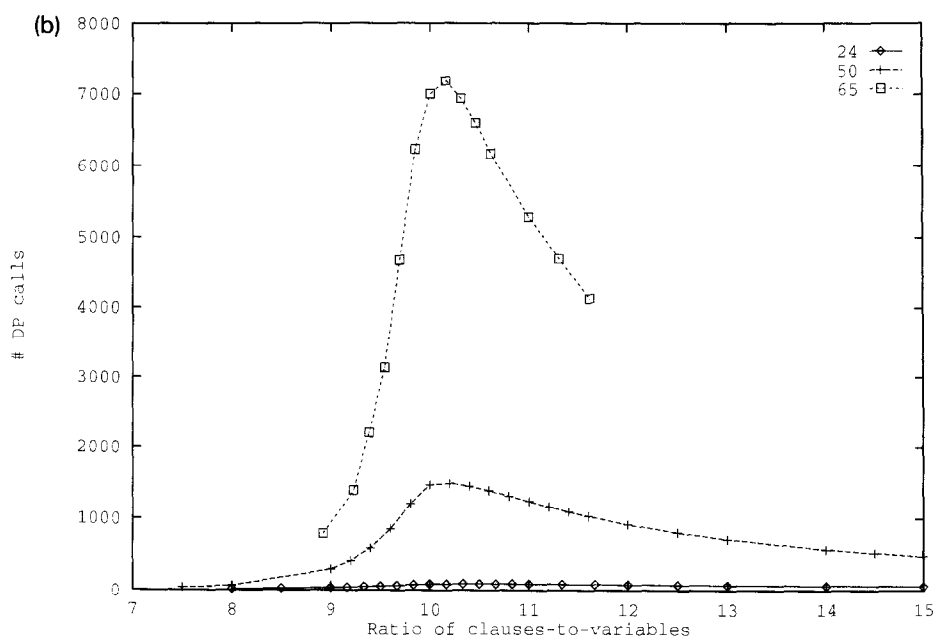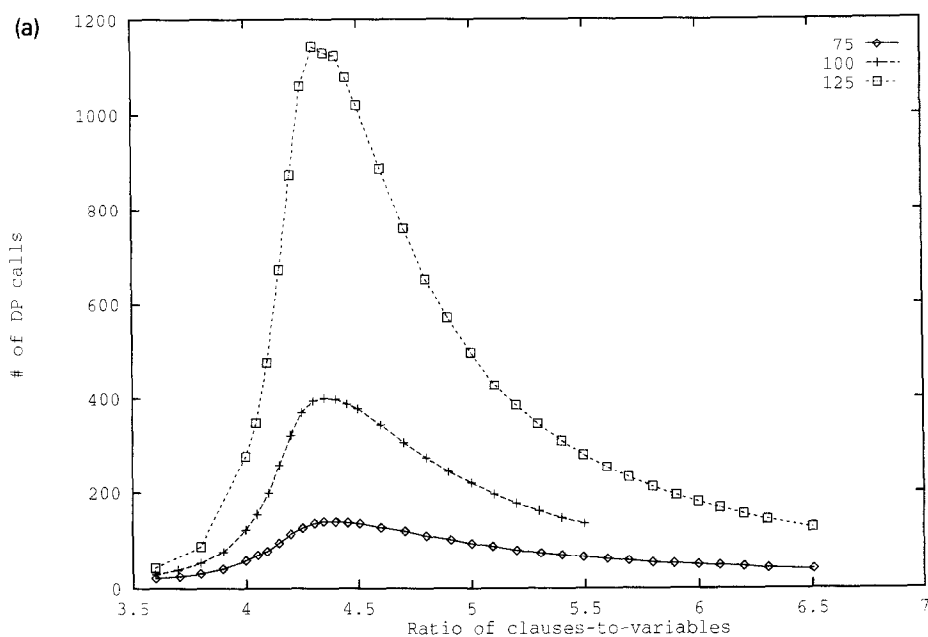
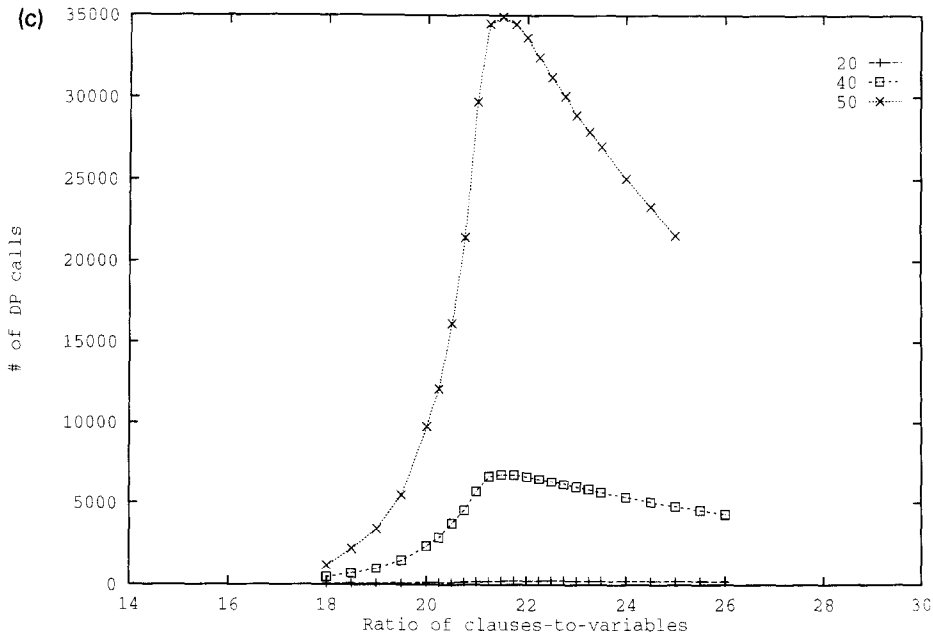Fig. 2. DP data for (a) 3-SAT, (b) 4-SAT, and (c) 5-SAT.
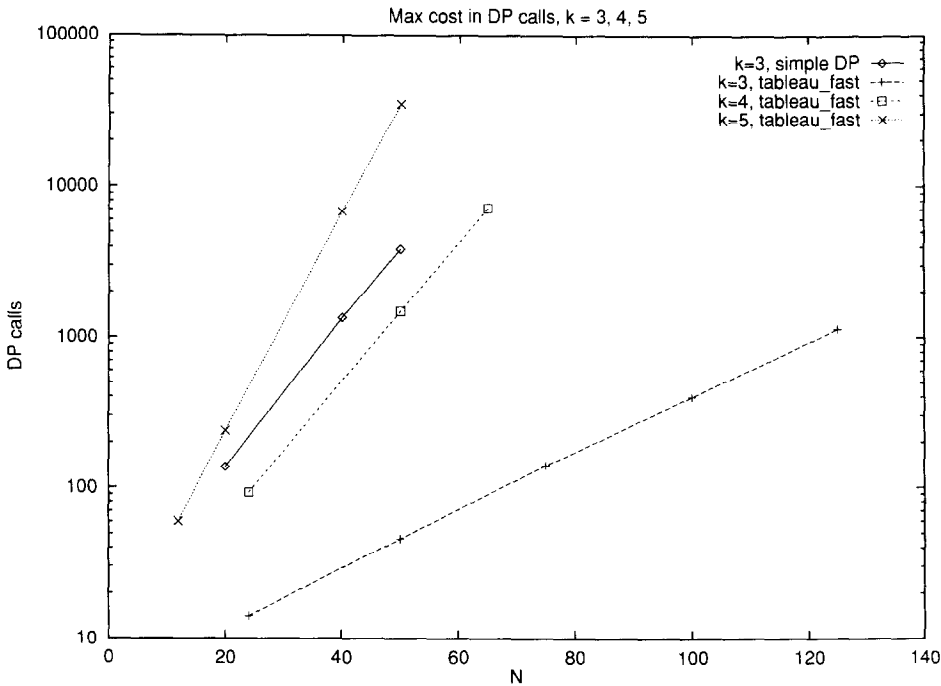
Fig. 2 — continued.

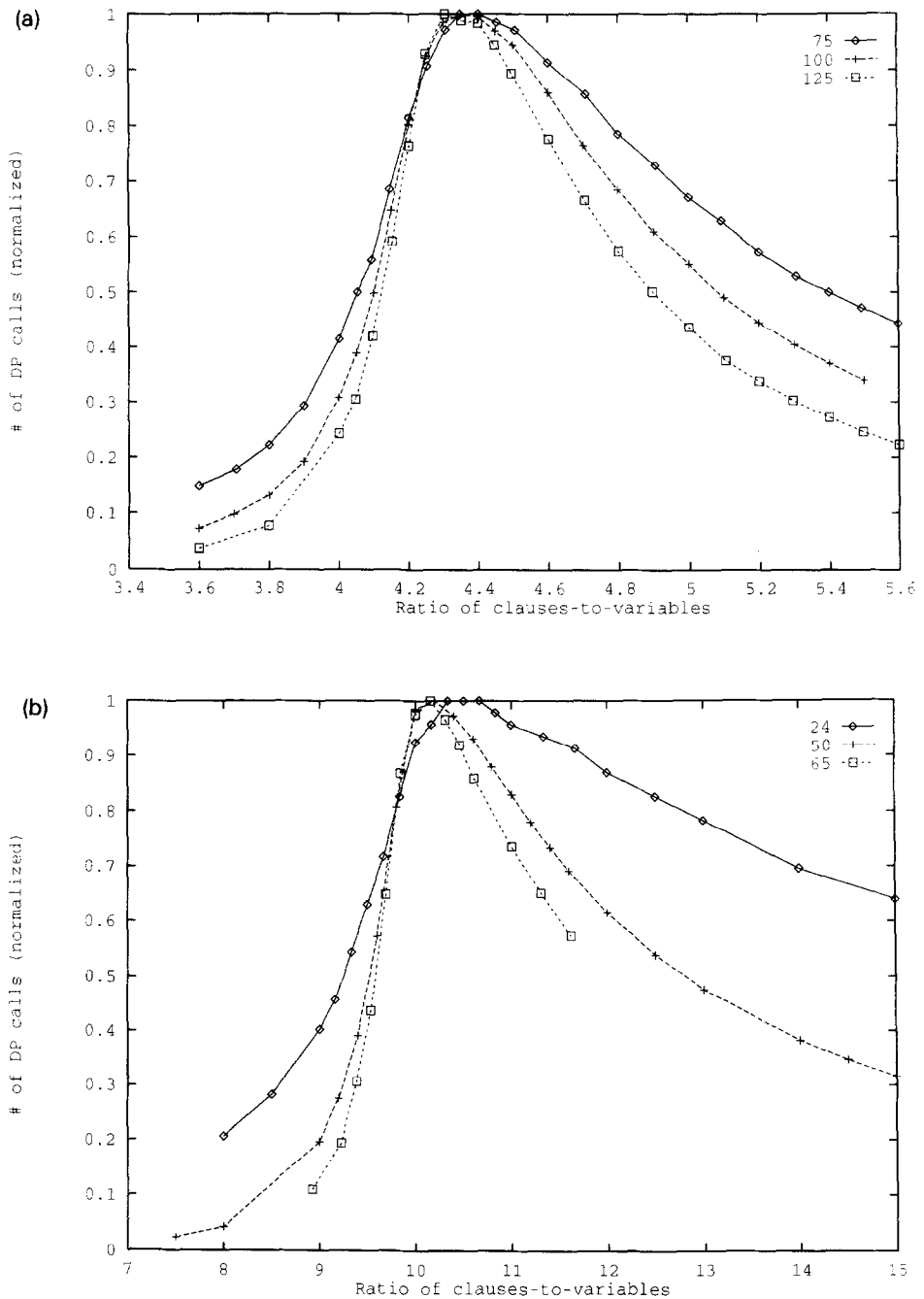Fig. 3. Maximum cost of DP search, $k = 3, 4, 5$.

Fig. 4. Normalized DP data for (a) 3-SAT, (b) 4-SAT, and (c) 5-SAT.
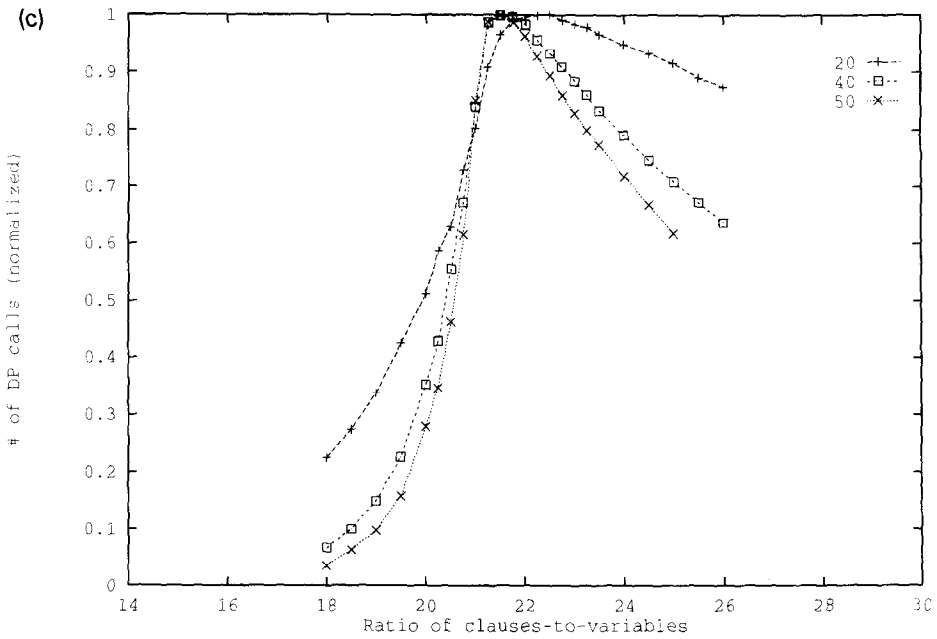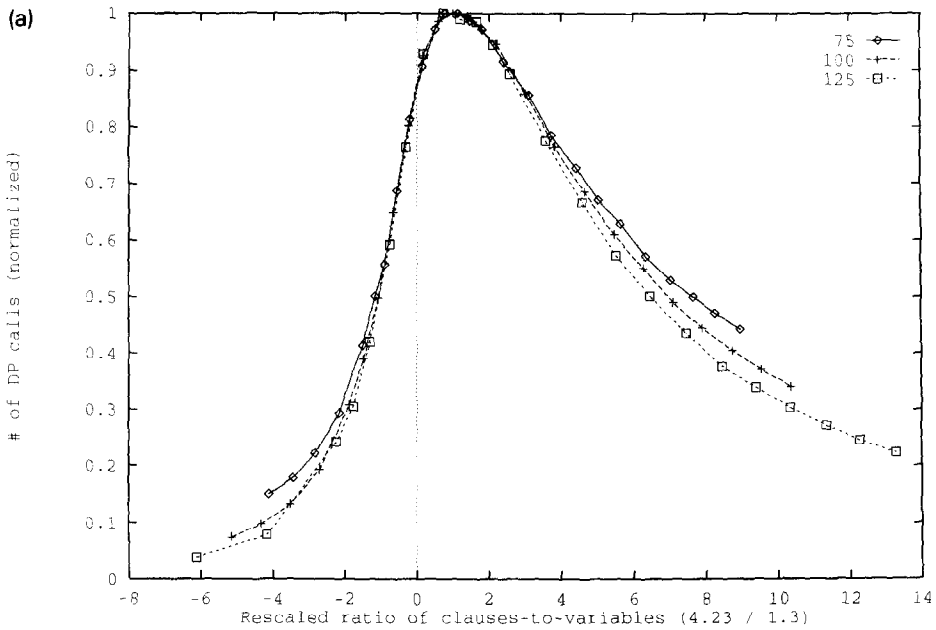
Fig. 4 — continued.

Fig. 5. Normalized DP data for (a) 3-SAT, rescaled with $\alpha = 4.23$ and $\nu = 1.3$; (b) 4-SAT, rescaled with $\alpha = 9.7$ and $\nu = 1.25$; (c) 5-SAT, rescaled with $\alpha = 20.9$ and $\nu = 1.1$.
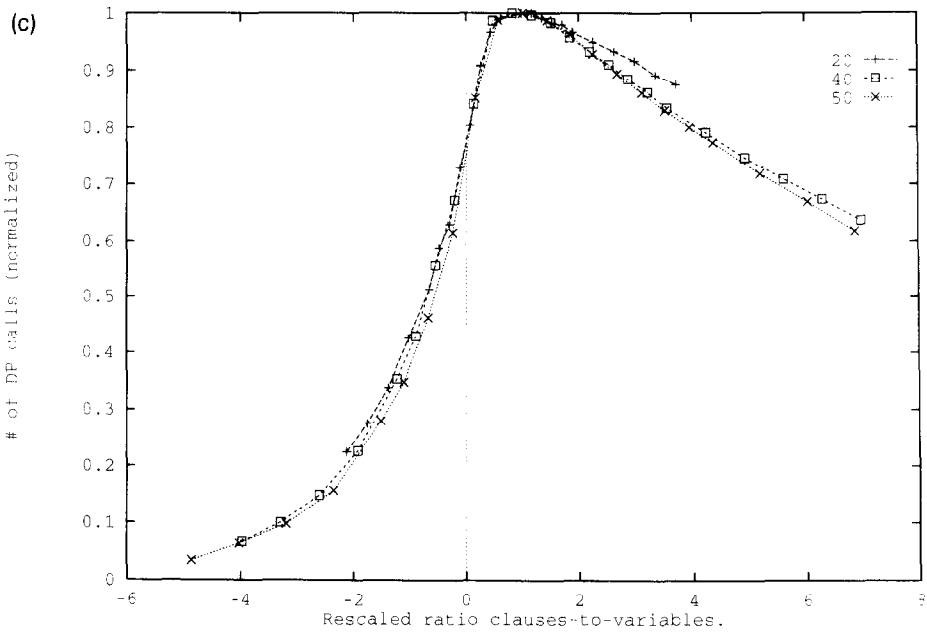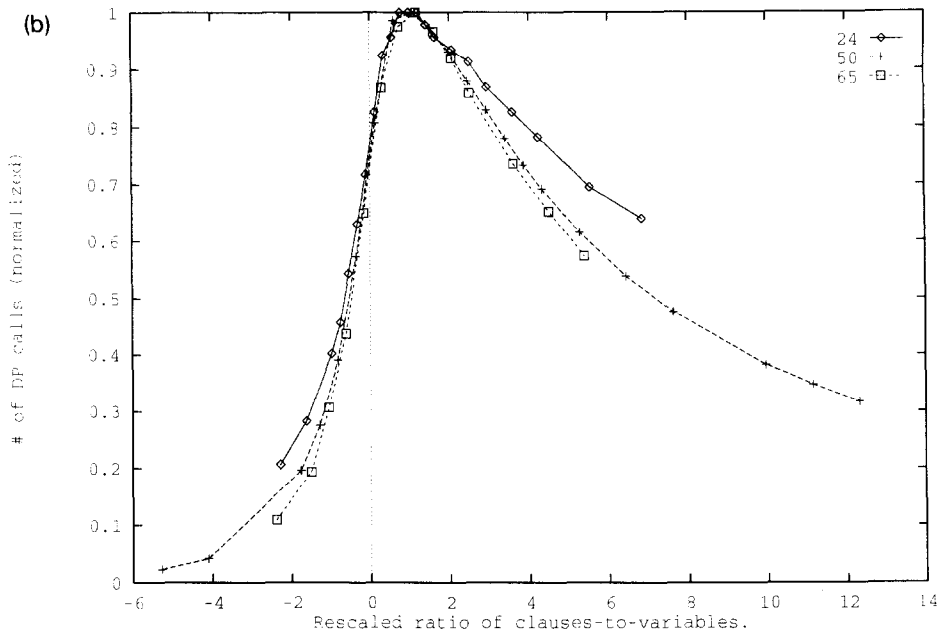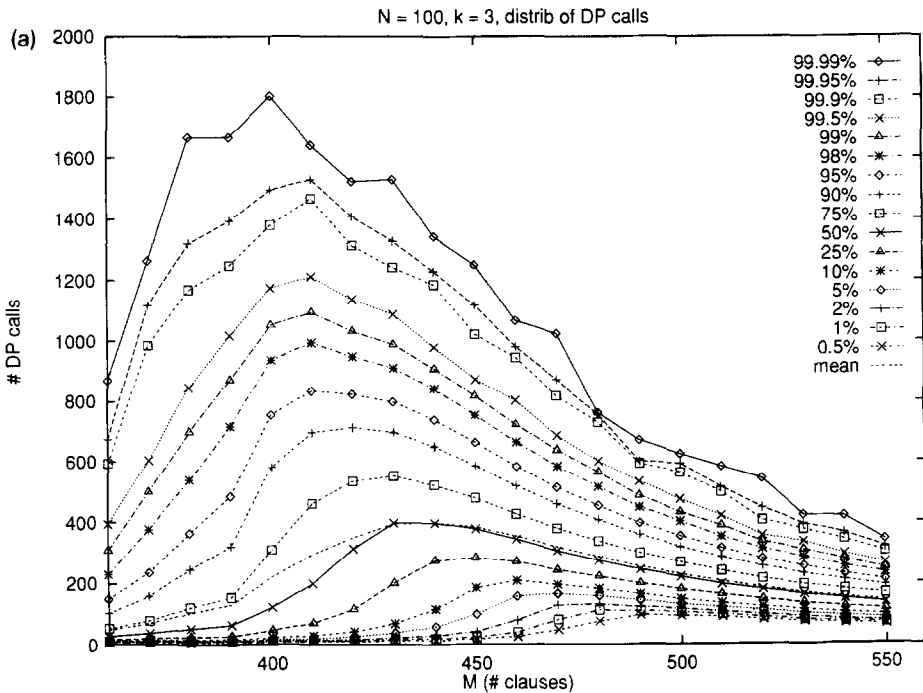
Fig. 5 — continued.

Fig. 6. DP data for 3-SAT, $N = 100$, detailed distribution, 10,000 expressions per point.

that the critical points $(y = 0)$ for $k = 3, 4, 5$ occur when about 0.3 of the expressions are unsatisfiable, and that $y = 0.9$ corresponds to approximately 0.55 of the expressions being satisfied.

The behavior seen in Figs. 2, 4 and 5 has some similarities with critical slowing down at conventional phase transitions, in which relaxation times diverge at the critical point and finite-size scaling can be used to determine the power law with which the divergence occurs. Here the cost of a solution diverges even more strongly, as e^N, so we have simply normalized each set of curves to their maxima before rescaling.

4. A more detailed look at the cost distribution

In order to understand the sources of the exponentially diverging computational cost, we now examine in detail the cost of determining satisfiability for 10,000 3-CNF expressions each with $N = 100$, at values of M from 360 (almost completely out of the critical regime into the SAT phase—there were only 9 UNSAT expressions encountered) to 560 (no SAT expressions were found at $M = 550$ or 560). In Figs. 6(a)–6(c), we connect points which are at the same percentile in the resulting distributions of all 10,000 cases at each M. The figure shows that the median cost is largest for $M = 430$ and 440, slightly above the position, $M = 420$, of the critical point. For values of M below 430,
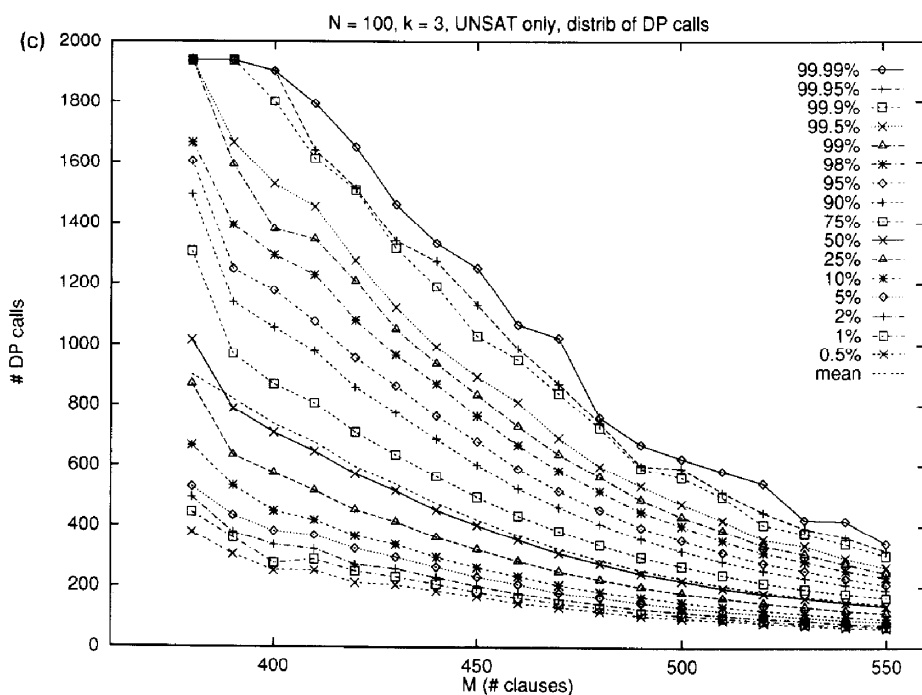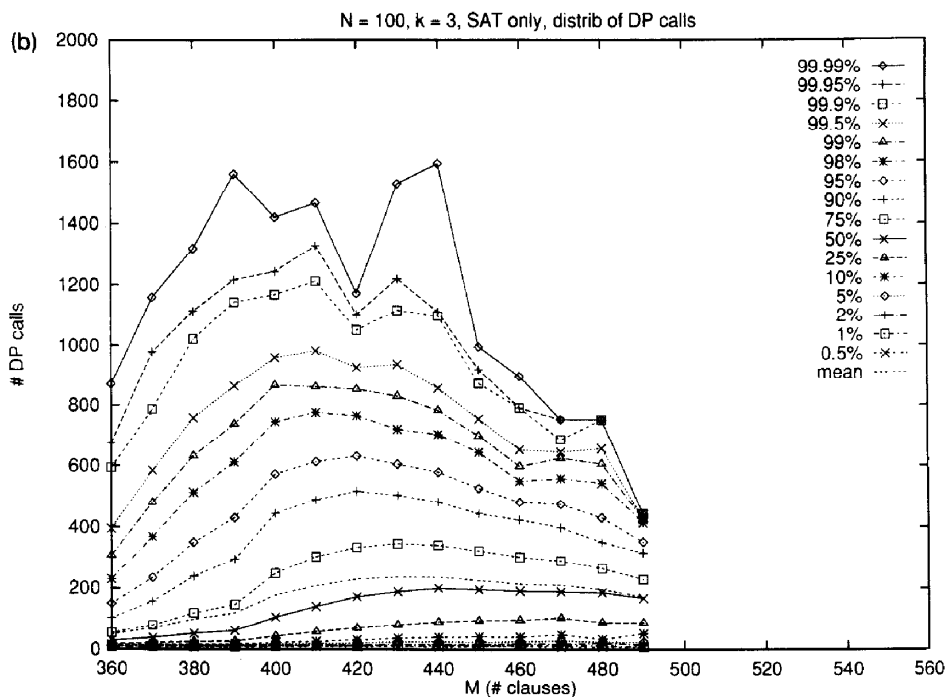
Fig. 6 —continued.

note that the mean cost is noticeably greater than the median The rare states contributing to the increased mean are seen as an erratic peak in the extreme tails (for example, the 99.95 and 99.99 percentile lines) occurring roughly in the range $M = 380$–400, below the critical point. Similar observations of very rare, time-consuming runs occurring in the mostly "easy" phase of a combinatorial problem have been described recently by Hogg and Williams [11] and by Gent and Walsh [9].

When separating the SAT (Fig. 6(b)) from the UNSAT (Fig. 6(c)) cases in our sample, we found that both the SAT and UNSAT expressions give rise to the longest computations, but that only the SAT states account for the shortest running times, which can be attributed to the fact that DP halts after finding the first assignment of a satisfiable instance.

To consider the variability of SAT instances further, we give in Figs. 7(a) (linear scale) and 7(b) (log scale) the N-dependence of the cost distribution in the SAT phase, at $\alpha = 3$. 10,000 expressions were examined at each point, except for $N = 800$, which is based on 4400 expressions. The cost is linear in N up to the 95th percentile, over the range plotted. Fig. 7(b) shows that the points in the high tail of the distribution appear to increase exponentially with N, but with extremely weak coefficients, perhaps as $e^{N/300}$. In contrast, Fig. 8(a) (for the cases with $\alpha = 4.2$ and N ranging from 12 to 200) shows the *entire* distribution shifting up and broadening as $e^{N/c}$ as N increases. The critical region is thus distinctly harder to solve than the SAT phase, as had been observed previously by considering only the median cost [22]. At $\alpha = 3.0$, we find a cost of 100 DP calls at $N = 1200$ and the subsequent increase is linear in N; at $\alpha = 4.2$, solving $N = 200$ requires over 100 DP calls, and the subsequent increase is exponential in n.

When separating the SAT from the UNSAT expressions (Figs. 8(b) and 8(c)), we found that the increased variability in the critical region is associated with the SAT states. The UNSAT expressions remain spread over about a factor 8 at each N. In both cases, the mean and median costs increase exponentially with N.

5. Role of DP versus expressions

To sort out the various factors in the range of results obtained for satisfiable expressions, we distinguish the variability due to the expressions themselves from that due to the performance of DP search on an individual expression. We generated 20 distinct random expressions, each with $N = 400$, $M = 1200$, and ran DP 5000 times on each, relabelling the variables within the expression for each run. We also generated 5000 random expressions with $N = 400$, $M = 1200$, and ran DP once on each. The resulting cumulative distributions are plotted in Figs. 9(a) (linear scale) and 9(b) (log scale).[3] We see that running DP many times on any of the expressions accounts for all the observed variability. This is not to say that every expression behaves exactly the same.

---

[3] The collection of 20 instances included the easiest and hardest expression found in our sample of 5000 instances. The cost distributions of random relabellings of those two instances do not appear to be any different from the distributions obtained for the other instances.
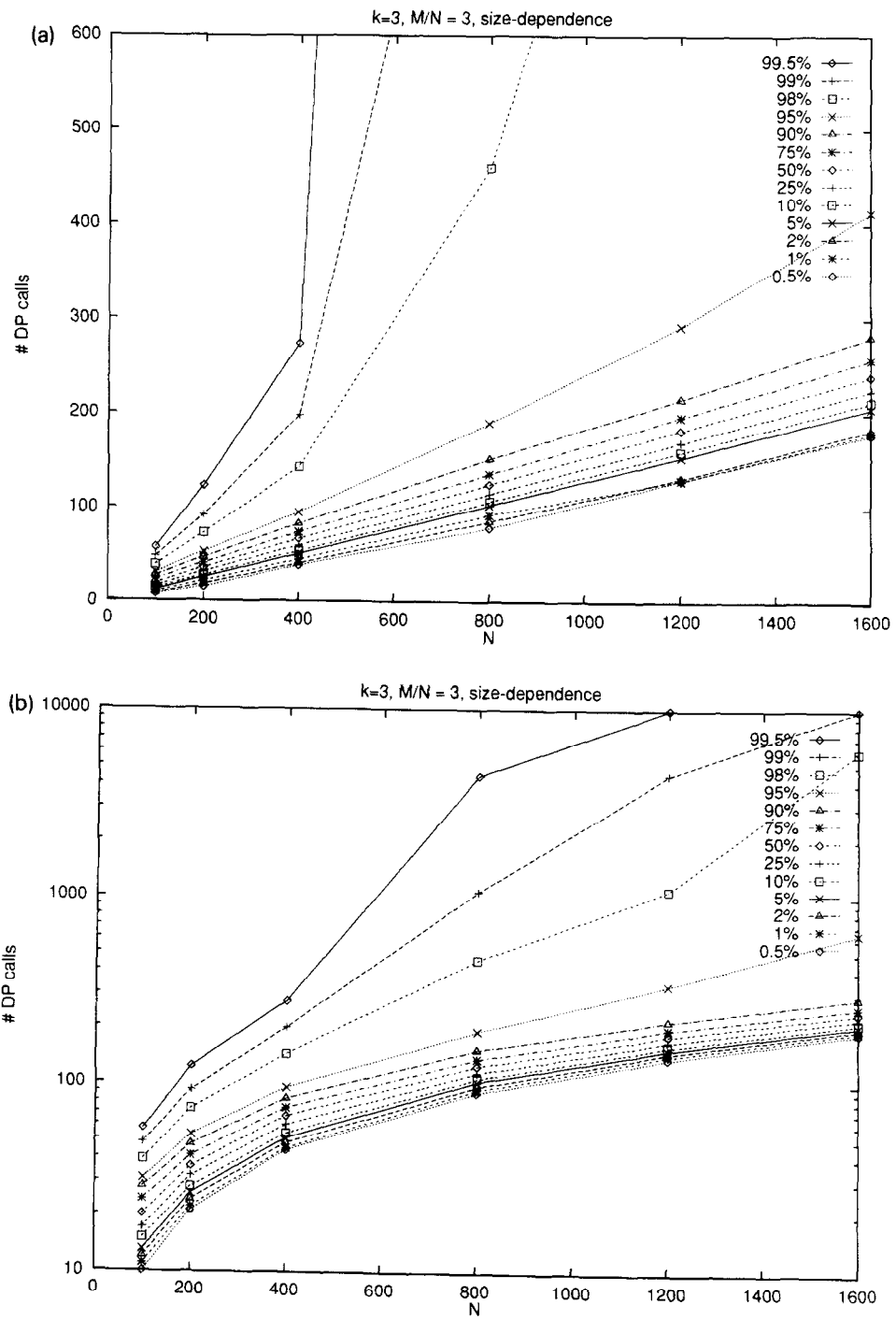
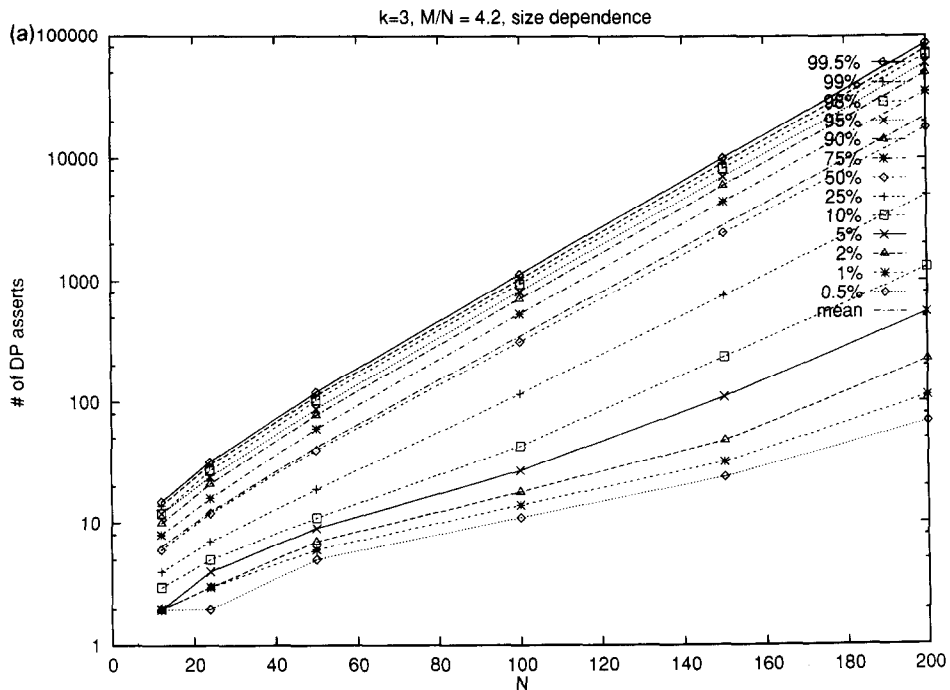Fig. 7. DP data in "easy" SAT area. 3-SAT, $N = 400$, and $M/N = 3.0$.

Fig. 8. DP data in "hard" area. 3-SAT, $N = 200$, and $M/N = 4.2$. Note logarithmic vertical scale.

From Fig. 9(a), we see that the distributions of costs resulting from different expressions indeed differ, although they bracket the distribution obtained by sampling each expression for one run only. Using the Kolmogorov–Smirnov criterion (see [19, pp. 54–56] for distinguishability of cumulative distributions, we estimate that the likelihood that the same process gives rise to any two of these distributions, with the differences resulting from only sampling error, is roughly e^{-100}.

The practical consequence of Fig. 9(a) is that none of these expressions is inherently hard. In fact none of them has even a 10 per cent probability of requiring more than 100 DP calls for solution. (Gent and Walsh have made a similar observation about the behavior of DP on P-SAT, an easier problem. See also [1].) We made use of this fact to obtain data for $N = 1200$ and 1600 at $\alpha = 3$, by stopping any runs exceeding 10,000 DP calls, relabelling, and restarting DP. At $N = 1200$, 76 out of 10,000 samples required an additional run to obtain a solution, one required two additional runs. At $N = 1600$, 167 expressions required a second run, and four took two additional runs. This trick gave a correction to the 99.5 and 99.9 percentile data in Fig. 7; the expressions requiring additional runs were treated as having a cost of 10,000 DP calls.

Fig. 9(b) shows power-law tails in all of the histograms, similar in form to the distributions reported by Hogg and Williams [11]. The distributions plotted in Figs. 9(a) and 9(b) are cumulative distributions. If we define $C(d)$ as the fraction of solutions whose cost exceeds d, then in terms of the probability distribution, $p(d)$, for a solution to cost d,
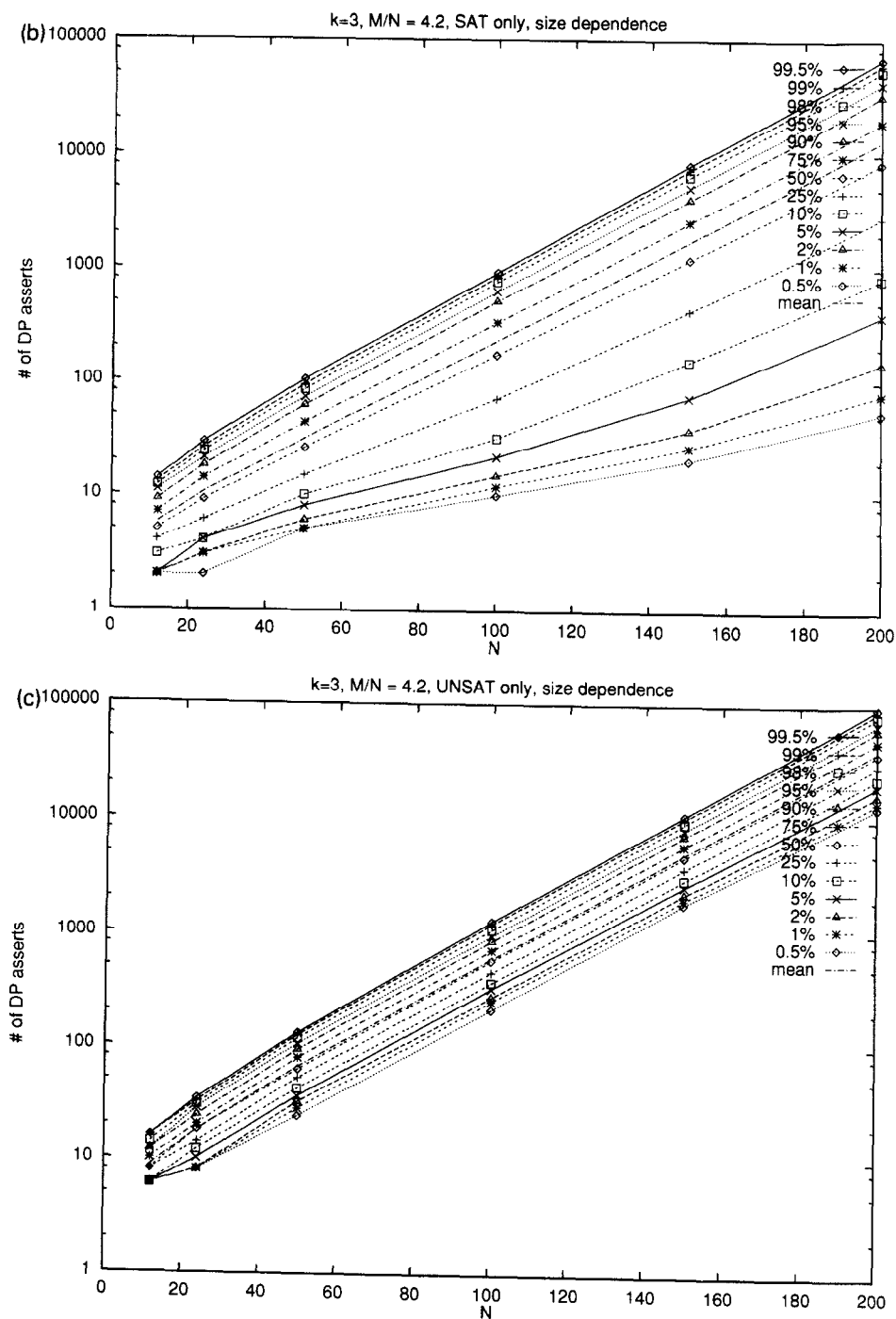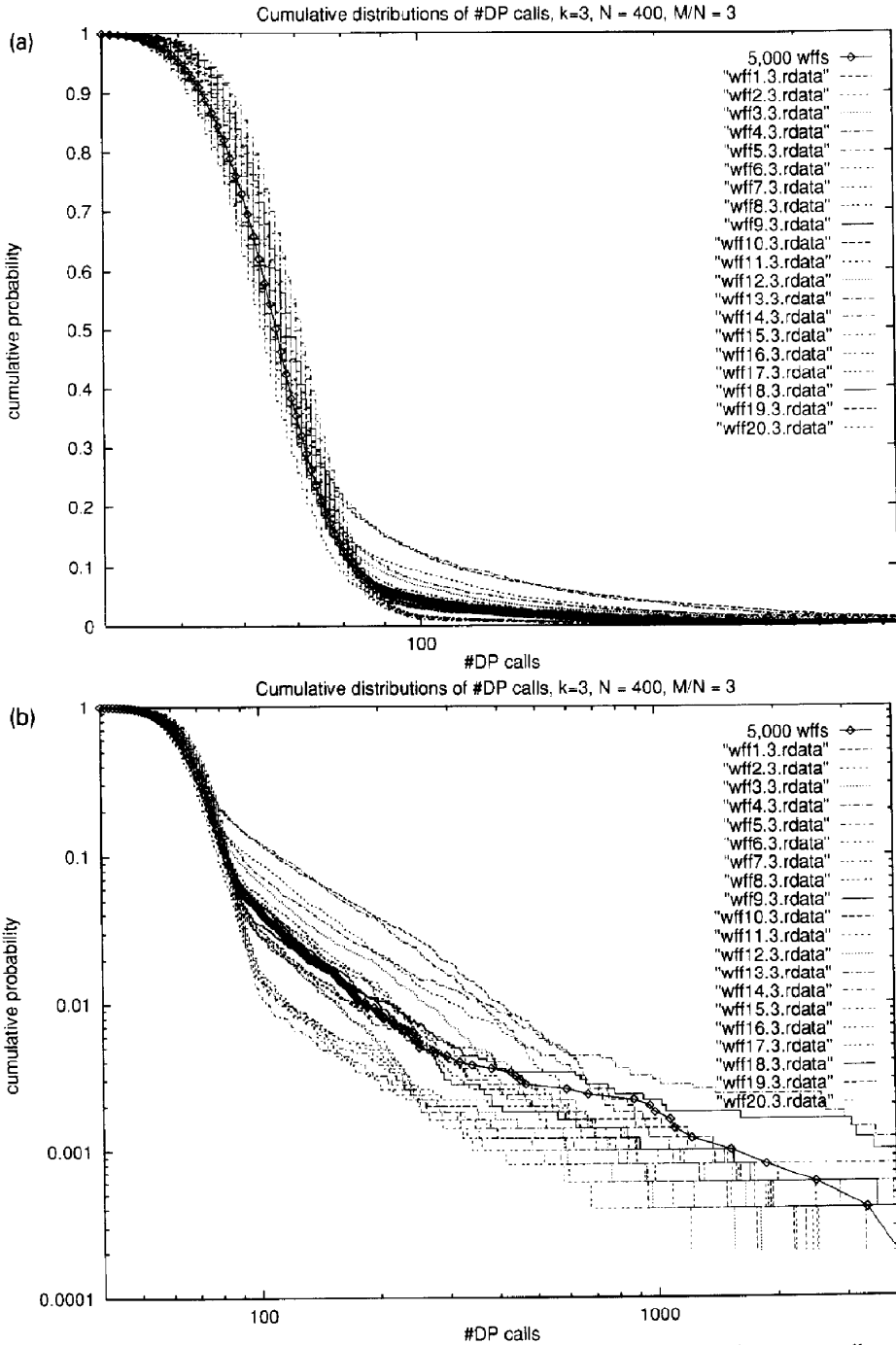
Fig. 8 — continued.

Fig. 9. Cumulative distribution of DP calls for repeated relabellings of 20 SAT expressions, (a) semilog plot, (b) log-log plot. Distribution of results from running DP once each on 5000 different expressions is shown as data points. 3-SAT, $N = 400$, and $M/N = 3$.
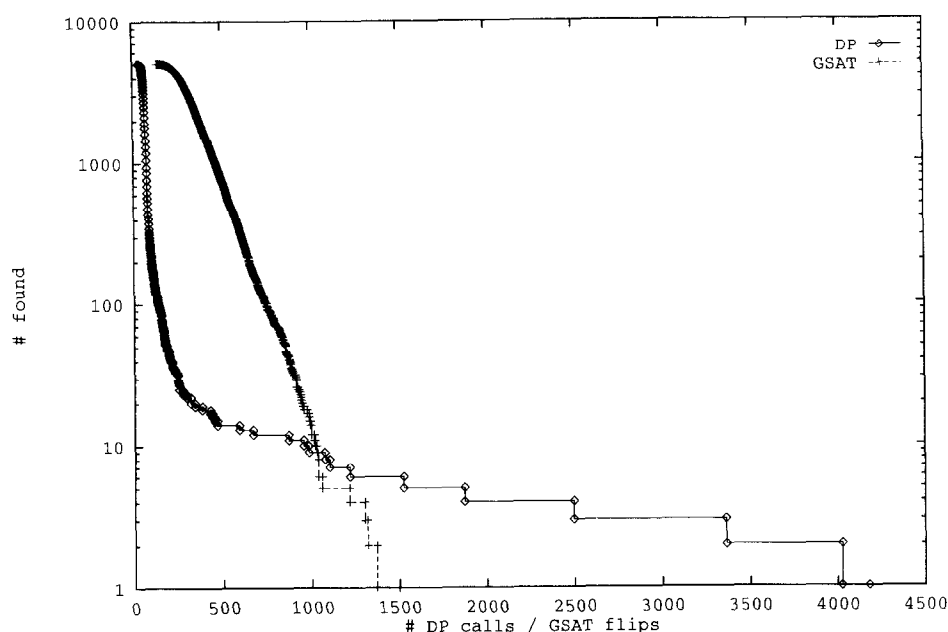
Fig. 10. Cumulative distribution of GSAT flips and DP calls for sample of 5000 expressions. 3-SAT, $N = 400$, and $M/N = 3$.

$$C(d) = \int\limits_{d}^{\infty} \rho(d).$$

In Fig. 9(b), the log-log plot, it appears that $C(d)$ decreases at first as d^{-2}, and later with some smaller slope, perhaps tending toward d^{-1}. This implies that $\rho(d)$ is proportional to d^{-3} or perhaps d^{-2}. Except when $\rho(d)$ is strictly equal to d^{-2} the average of d is well behaved. Although the mean can exceed the median with these lopsided distributions, it does not appear likely that we are observing divergent mean costs. However it is still hard to treat such tails with Monte Carlo sampling. This accounts for the high scatter in the results for high percentiles in Fig. 6.

We also ran experiments with a stochastic satisfiability procedure (GSAT, Selman et al. [23]). Fig. 10 shows the cumulative distribution of running GSAT on a sample of 5000 expressions below the threshold (3-SAT, $N = 400$, and $\alpha = 3$). We also included the data for DP on the same sample. Even though there is still a spread in the GSAT data, there is a clear qualitative difference between the GSAT and DP curve. In particular, the GSAT data does not show the long tail we observe in the DP data. This is further evidence that the high costs observed on some instances in the "easy" SAT region are most likely a result of the use of a our particular deterministic search procedure. [4]

---

[4] Apparently, on certain labelling, DP follows a ill-chosen branching sequence, leading to an exponentially large search tree.
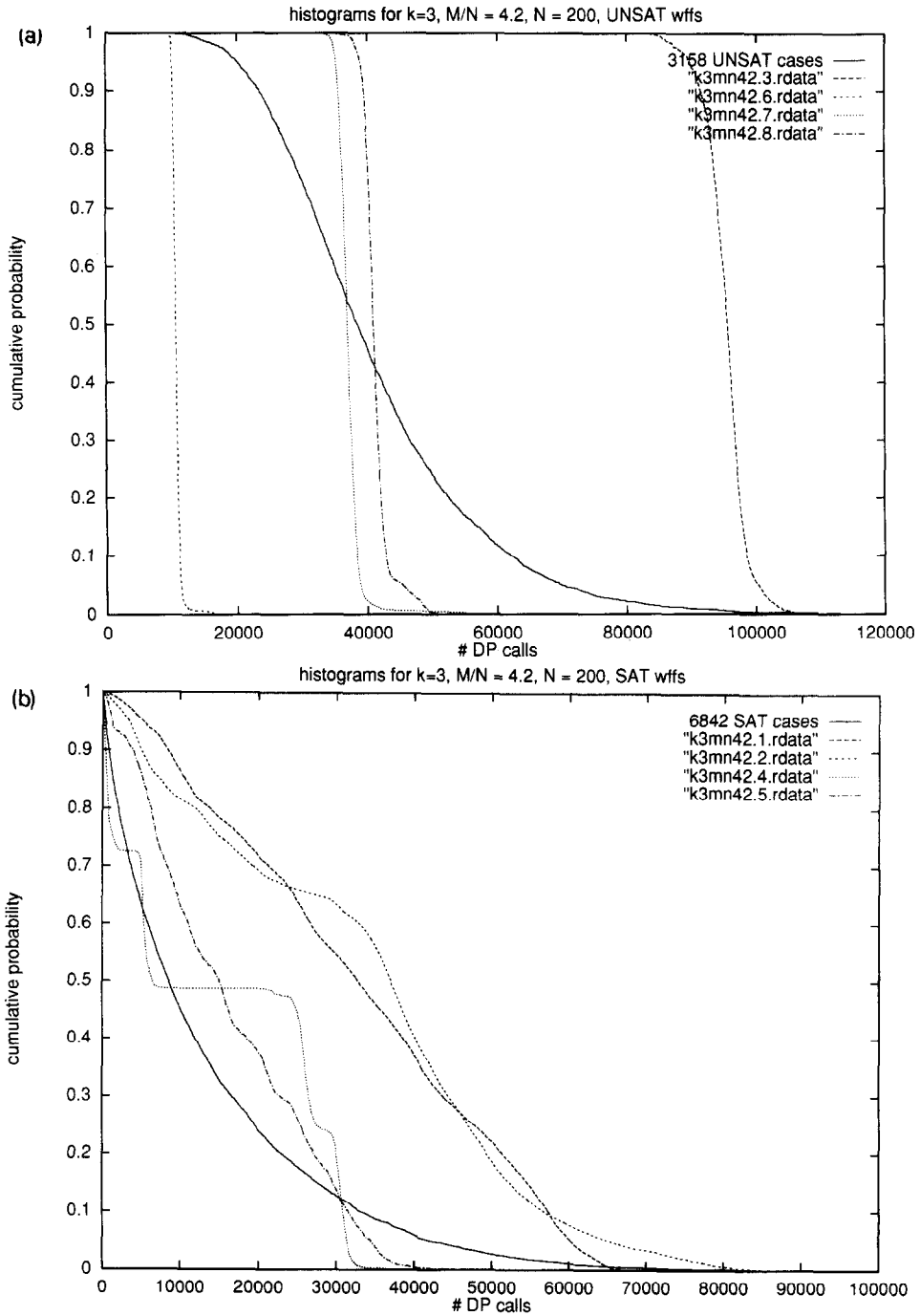
Fig. 11. Cumulative distributions of DP calls. 3-SAT, $N = 200$ and $M/N = 4.2$. (a) shows four UNSAT cases, and (b) four SAT expressions. Results of running many expressions, with DP run only once are shown as solid lines.

Finally, we consider the situation in the critical regime. In Fig. 11, we give a summary of results obtained by repeatedly applying DP to SAT and to UNSAT expressions at $\alpha = 4.2$. In Figs. 11(a) and 11(b) we show results of repeatedly applying DP to four SAT expressions and to four UNSAT expressions (relabelled each time), and compare these distributions to the distributions obtained by applying DP once to each of 3158 UNSAT expressions (in Fig. 11(a)) and 6842 SAT expressions (in Fig. 11(b)). Note that in Fig. 11(a), the overall range of costs in the distribution of 3158 UNSAT expressions is about a factor of 8, but each individual expression's cost of solution ranges over only a factor of 30–50 per cent. Therefore, in the critical regime on UNSAT expression, the variability due to DP is only a limited factor in the observed variability between expressions. Finally, the distributions obtained by multiple solutions of a single SAT expression at the critical point differ in form from the overall distribution obtained by generating 5000 SAT expressions (Fig. 11(b)) and solving each one once. The SAT expressions shown in Fig. 11(b) individually have cumulative distributions that look roughly linear, indicating approximately a uniform distribution of costs. A semilog plot of the data in Fig. 11(b) suggests that at criticality, $C(d) \approx e^{-d}$, quite different in form from the distributions found in the easy SAT phase as shown in Fig. 9.

6. Conclusions and comments

We have shown that finite-size scaling can be used to characterize the dependency on N of the cost of DP search in k-SAT. Rescaling gives us a a single, scale-invariant function in the vicinity of α_c, the critical ratio of M/N. This function allows us to predict the cost of search as a function of N through the threshold, where the inherent cost of this search is highest. Our results further illustrate the applicability of finite-size scaling methods from statistical mechanics in the analysis of properties of combinatorial structures near a phase transition. The situation is different far below the threshold, in the SAT phase, where we find very rare, long searches which make it more difficult to measure the true average cost of search, a result similar to that of Hogg and Williams [16], and of Gent and Walsh [9]. The high computational costs do not appear to be inherent in the formulas but rather the result of running a systematic search procedure on a random instance. Much of the large variation in search cost can be observed by running DP many times on a single formula which is repeatedly randomly relabelled. Understanding the practical impact of these rare searches also requires knowing how the effect scales with N beyond the sizes of problems which today's computers and search heuristics can handle. Our results suggest that only a fixed fraction of runs on a given expression, or a fixed fraction of expressions each searched once, contribute costs which diverge exponentially in N, but this cannot be determined with high confidence from the present results.

We can instead compare the behavior of DP search on large k-SAT problems to two random systems for which the large-N behavior is better known. These are glasses (like the SiO_2 found in windows) and spin glasses, magnetic systems which exhibit many of the properties of structural glasses. Both have been extensively modelled and analyzed.

For an overview and references on spin glasses, see the books by Mezard and Parisi [21], and by Fischer and Hertz [8]. Spin glass methods described in [21] have recently been employed by Monasson and Zecchina [5] to calculate the mean number of satisfying configurations for a k-SAT formula in the SAT phase. They find good agreement with numerical experiments such as those reported in [15].

SiO_2 can form either crystalline quartz or random glass, but when macroscopic amounts of the substance are assembled by either adding atoms individually to a growing nucleus (a common technique in integrated circuit technology) or by cooling molten SiO_2 rapidly, the result is always random glass, a structure with higher energy than quartz. Nevertheless, the lower energy structure does exist, and crystals of quartz can be grown with great care and very slow cooling. Spin glasses are incapable of settling into a regular structure which is optimal in its energy, but they take on magnetically rigid arrangements, just as glass is a structurally rigid arrangement of atoms. Finding the ground state of simple models of a spin glass is an NP-complete problem, and evidence from computer experiments [24] suggests that for a search for low energy arrangements to approach within $1/N$ of the true ground state of a spin glass requires e^N time or computational work.

k-SAT near the critical point seems to us to be behaving as a spin glass, with no evidence that a means of easy search can be found when $N \to \infty$. The open question is whether the apparently easy search for satisfying solutions in the SAT phase becomes like a structural glass, in which the solutions, though known to be present, are extremely difficult to find. If the fraction of cases with exponentially large cost remains constant as $N \to \infty$, then a solution can be found with acceptable probability by merely running DP several times, always stopping if the solution is not encountered in the initial part of the search. If the fraction of costly searches increases with N, then even in the SAT phase, the solution which must be present may become extremely hard to construct.

There is one additional curious characteristic of structural glasses that is shared by k-SAT. Kirkpatrick, Györgyi, Tishby, and Troyansky [15] showed that the entropy measured in k-SAT extrapolates to zero at values of α just above the observed α_c. This evidence that a simple transition might be concealed behind the effects of randomness is one of the oldest speculations about structural glass (see book by Goldstein and Simha [10] or, for a more modern view, T. Kirkpatrick et al. [18]).

Acknowledgement

---

[5] *Phys. Rev. Lett.* (to appear).

References

[1] A. Baker, Intelligent backtracking on the hardest constraint problems, *J. Artif. Intell. Res.* (to appear).

[2] B. Bollobás, *Random Graphs* (Academic Press, London, 1985).

[3] M. Buro and H. Kleine-Büning, Report on a SAT competition, Technical Report #110, Department of Mathematics and Informatics, University of Paderborn, Germany (1992).

[4] P. Cheeseman, B. Kanefsky and W.M. Taylor, Where the really hard problems are, in: *Proceedings IJCAI-91*, Sydney, Australia (1991) 163-169.

[5] S.H. Clearwater, B.A. Huberman and T. Hogg, Cooperative solution of constraint satisfaction problems, *Science* 254 (5035) (1991) 1181-1183.

[6] J.M. Crawford and L.D. Auton, Experimental results on the crossover point in satisfiability problems, in: *Proceedings AAAI-93*, Washington, DC (1993).

[7] M. Davis and H. Putnam, A computing procedure for quantification theory, *J. ACM* 7 (1960) 201-215.

[8] K.H. Fischer and J.A. Hertz, *Spin Glasses* (Cambridge University Press, Cambridge, UK, 1991).

[9] I.P. Gent and T. Walsh, Easy problems are sometimes hard, The DIMACS Challenge on Satisfiability Testing, Piscataway, NJ (1993).

[10] M. Goldstein and R. Simha, *The Glass Transition and the Nature of the Glassy State*, Annals of the NY Academy of Sciences 279 (1976).

[11] T. Hogg and C.P. Williams, The hardest constraint problems: a double phase transition, *Artif. Intell.* 69 (1994) 359-377.

[12] T. Hogg and C.P. Williams, Expected gains from parallelizing constraint solving for hard problems, Xerox PARC preprint (1994).

[13] S. Janson, D.E. Knuth, T. Luczak and B. Pittel, The birth of the giant component, *Random Structures Algorithms* 4 (1993) 231-358.

[14] D. Johnson and M. Trick, The DIMACS Challenge on Satisfiability Testing, Piscataway, NJ (1993).

[15] S. Kirkpatrick, G. Györgyi, N. Tishby and L. Troyansky, in: J.D. Cowan, G. Tesauro and J. Alspector, eds., Advances in Neural Information Processing Systems 6 (Morgan-Kaufmann, San Francisco, CA, 1993) 439-447.

[16] S. Kirkpatrick and B. Selman, Critical behavior in the satisfiability of random Boolean expressions, *Science* 264 (1994) 1297-1301.

[17] S. Kirkpatrick and R.H. Swendsen, Statistical mechanics and disordered systems, *Commun. ACM* 28 (1985) 363-373.

[18] T. Kirkpatrick, D. Thirumalai and P. Wolynes, Scaling Concepts in Glasses, *Phys. Rev. A* 40 (1989) 1045-5589.

[19] D.E. Knuth, *The Art of Computer Programming, Vol. 2* (Addison-Wesley, Boston, MA, 2nd ed., 1981).

[20] T. Larrabee and Y. Tsuji, Evidence for a satisfiability threshold for random 3CNF formulas, in: *Proceedings AAAI Spring Symposium on AI and NP-Hard Problems*, Palo Alto, CA (1993).

[21] M. Mézard, G. Parisi and M.A. Virasoro, *Spin Glass Theory and Beyond* (World Scientific, Singapore, 1986).

[22] D.G. Mitchell, B. Selman and H.J. Levesque, Hard and easy distributions of SAT problems, in: *Proceedings AAAI-92*, San Jose, CA (1992) 456-465.

[23] B. Selman, H.J. Levesque and D.G. Mitchell, GSAT: a new method for solving hard satisfiability problems, in: *Proceedings AAAI-92*, San Jose, CA (1992) 440-446.

[24] C.M. Soukoulis, K. Levin and G.S. Grest, *Phys. Rev. Lett.* 48 (1982) 1756.

[25] J. Spencer, *Ten Lectures on the Probabilistic Method* (SIAM, Philadelphia, PA, 1987).