Comparing Beliefs, Surveys and Random Walks

Erik Aurell
SICS, Swedish Institute of Computer Science
P.O. Box 1263, SE-164 29 Kista, Sweden
and Dept. of Physics,
KTH – Royal Institute of Technology
AlbaNova – SCFAB SE-106 91 Stockholm, Sweden
eaurell@sics.se

Uri Gordon and Scott Kirkpatrick
School of Engineering and Computer Science
Hebrew University of Jerusalem
91904 Jerusalem, Israel
{guri,kirk}@cs.huji.ac.il

Abstract

Survey propagation is a powerful technique from statistical physics that has been applied to solve the 3-SAT problem both in principle and in practice. We give, using only probability arguments, a common derivation of survey propagation, belief propagation and several interesting hybrid methods. We then present numerical experiments which use WSAT (a widely used random-walk based SAT solver) to quantify the complexity of the 3-SAT formulae as a function of their parameters, both as randomly generated and after simpli£cation, guided by survey propagation. Some properties of WSAT which have not previously been reported make it an ideal tool for this purpose – its mean cost is proportional to the number of variables in the formula (at a £xed ratio of clauses to variables) in the easy-SAT regime and slightly beyond, and its behavior in the hard-SAT regime appears to re¤ect the underlying structure of the solution space that has been predicted by replica symmetry-breaking arguments. An analysis of the tradeoffs between the various methods of search for satisfying assignments shows WSAT to be far more powerful than has been appreciated, and suggests some interesting new directions for practical algorithm development.

1 Introduction

Random 3-SAT is a classic problem in combinatorics, at the heart of computational complexity studies and a favorite testing ground for both exactly analyzable and heuristic solution methods which are then applied to a wide variety of problems in machine learning and arti£cial intelligence. It consists of a ensemble of randomly generated logical expressions, each depending on N Boolean variables x_i, and constructed by taking the AND of M clauses. Each clause a consists of the OR of 3 "literals" $y_{i,a}$. $y_{i,a}$ is taken to be either x_i

or $\neg x_i$ at random with equal probability, and the three values of the index i in each clause are distinct. Conversely, the neighborhood of a variable x_i is V_i, the set of all clauses in which x_i or $\neg x_i$ appear. For each such random formula, one asks whether there is some set of x_i values for which the formula evaluates to be TRUE. The ratio $\alpha = M/N$ controls the dif£culty of this decision problem, and predicts the answer with high accuracy, at least as both N and M tend to in£nity, with their ratio held constant. At small α, solutions are easily found, while for suf£ciently large α there are almost certainly no satisfying con£gurations of the x_i, and compact proofs of this fact can be constructed. Between these limits lies a complex, spin-glass-like phase transition, at which the cost of analyzing the problem with either exact or heuristic methods explodes.

A recent series of papers drawing upon the statistical mechanics of disordered materials has not only clari£ed the nature of this transition, but also lead to a thousand-fold increase in the size of the concrete problems that can be solved [1, 2, 3] This paper provides a derivation of the new methods using nothing more complex than probabilities, suggests some generalizations, and reports numerical experiments that disentangle the contributions of the several component heuristics employed. For two related discussions, see [4, 5].

An iterative ”belief propagation” [6] (BP) algorithm for K-SAT can be derived to evaluate the probability, or ”belief,” that a variable will take the value TRUE in variable con£gurations that satisfy the formula considered. To calculate this, we £rst de£ne a message (”transport”) sent from a variable to a clause:

- $t_{i\to a}$ is *the probability that variable x_i satis£es clause a*

In the other direction, we de£ne a message (”in¤uence”) sent from a clause to a variable:

- $i_{a\to i}$ is *the probability that clause a is satis£ed by another variable than x_i*

In 3-SAT, where clause a depends on variables x_i, x_j and x_k, BP gives the following iterative update equation for its in¤uence.

$$i^{(l)}_{a\to i} \;=\; t^{(l)}_{j\to a} + t^{(l)}_{k\to a} - t^{(l)}_{j\to a} t^{(l)}_{k\to a} \tag{1}$$

The BP update equations for the transport $t_{i\to a}$ involve the products of in¤uences acting on a variable from the clauses which surround x_i, forming its ”cavity,” V_i, sorted by which literal (x_i or $\neg x_i$) appears in the clause:

$$A^0_i = \prod_{b\in V_i,\, y_{i,b}=\neg x_i} i_{b\to i} \qquad \text{and} \qquad A^1_i = \prod_{b\in V_i,\, y_{i,b}=x_i} i_{b\to i} \tag{2}$$

The update equations are then

$$t^{(l)}_{i\to a} = \begin{cases} \frac{i^{(l-1)}_{a\to i} A^1_i}{i^{(l-1)}_{a\to i} A^1_i + A^0_i} & \text{if } y_{i,a} = \neg x_i \\ \\ \frac{i^{(l-1)}_{a\to i} A^0_i}{i^{(l-1)}_{a\to i} A^0_i + A^1_i} & \text{if } y_{i,a} = x_i \end{cases} \tag{3}$$

The superscripts (l) and $(l-1)$ denote iteration. The probabilistic interpretation is the following: suppose we have $i^{(l)}_{b\to i}$ for all clauses b connected to variable i. Each of these clauses can either be satis£ed by another variable (with probability $i^{(l)}_{b\to i}$), or not be satis£ed by another variable (with probability $\left(1 - i^{(l)}_{b\to i}\right)$), and also be satis£ed by variable i itself. If we set variable x_i to 0, then some clauses are satis£ed by x_i, and some have to be satis£ed by other variables. The probability that they are all *satis£ed* is $\prod_{b\neq a, y_{i,b}=x_i} i^{(l)}_{b\to i}$. Similarly, if x_i is set to 1 then all these clauses b are satis£ed with probability $\prod_{b\neq a, y_{i,b}=\neg x_i} i_{b\to i}^{(l)}$. The products in (3) can therefore be interpreted as joint probabilities of independent events. Variable x_i can be 0 or 1 in a solution if the clauses in which x_i appears are either satis£ed directly by x_i itself, or by other variables. Hence

$$\text{Prob}(x_i) = \frac{A_i^0}{A_i^0 + A_i^1} \qquad \text{and} \qquad \text{Prob}(\neg x_i) = \frac{A_i^1}{A_i^0 + A_i^1} \tag{4}$$

A BP-based decimation scheme results from £xing the variables with largest probability to be either true or false. We then recalculate the beliefs for the reduced formula, and repeat.

To arrive at SP we introduce a modi£ed system of beliefs: every variable falls into one of three classes: TRUE in all solutions (1); FALSE in all solutions(0); and TRUE in some and FALSE in other solutions ($free$). The message from a clause to a variable (an in¤uence) is then the same as in BP above. Although we will again only need to keep track of one message from a variable to a clause (a transport), it is convenient to £rst introduce three ancillary messages:

- $\hat{T}_{i\to a}(1)$ is *the probability that variable x_i is true in clause a in all solutions*
- $\hat{T}_{i\to a}(0)$ is *the probability that variable x_i is false in clause a in all solutions*
- $\hat{T}_{i\to a}(free)$ is *the probability that variable x_i is true in clause a in some solutions and false in others*.

Note that there are here three transports for each directed link $i \to a$, from a variable to a clause, in the graph. As in BP, these numbers will be functions of the in¤uences from clauses to variables in the preceeding update step. Taking again the incoming in¤uences independent, we have

$$\begin{array}{lll} \hat{T}_{i\to a}^{(l)}(free) & \propto & \prod_{b\in V_i\setminus a} \quad i_{b\to i}^{(l-1)} \\ \hat{T}_{i\to a}^{(l)}(0) + \hat{T}_{i\to a}^{(l)}(free) & \propto & \prod_{b\in V_i\setminus a, y_{i,b}=x_i} \quad i_{b\to i}^{(l-1)} \\ \hat{T}_{i\to a}^{(l)}(1) + \hat{T}_{i\to a}^{(l)}(free) & \propto & \prod_{b\in V_i\setminus a, y_{i,b}=\neg x_i} \quad i_{b\to i}^{(l-1)} \end{array} \tag{5}$$

The proportionality indicates that the probabilities are to be normalized. We see that the structure is quite similar to that in BP. But we can make it closer still by introducing $t_{i\to a}$ with the same meaning as in BP. In SP it will then, as the case might be, be equal to to $T_{i\to a}(free) + T_{i\to a}(0)$ or $T_{i\to a}(free) + T_{i\to a}(1)$. That gives (compare (3)):

$$t_{i\to a}^{(l)} = \begin{cases} \frac{i_{a\to i}^{(l-1)} A_i^1}{i_{a\to i}^{(l-1)} A_i^1 + A_i^0 - A_i^1 A_i^0} & \text{if } y_{i,a} = \neg x_i \\ \\ \frac{i_{a\to i}^{(l-1)} A_i^0}{i_{a\to i}^{(l-1)} A_i^0 + A_i^1 - A_i^1 A_i^0} & \text{if } y_{i,a} = x_i \end{cases} \tag{6}$$

The update equations for $t_{i\to a}$ are the same in SP as in BP, ´.e. one uses (1) in SP as well. Similarly to (4), decimation now removes the most £xed variable, i.e. the one with the largest absolute value of $(A_i^0 - A_i^1)/(A_i^0 + A_i^1 - A_i^1 A_i^0)$. Given the complexity of the original derivation of SP [1, 2], it is remarkable that the SP scheme can be interpreted as a type of belief propagation in another belief system. And even more remarkable that the £nal iteration formulae differ so little.

A modi£cation of SP which we will consider in the following is to interpolate between BP

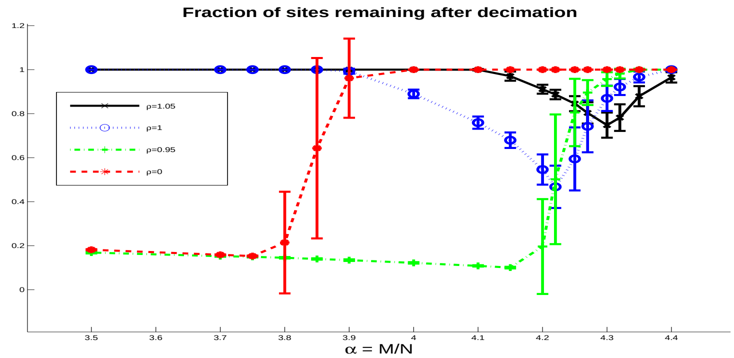


Figure 1: Dependence of decimation depth on the interpolation parameter ρ.

$(\rho = 0)$ and SP $(\rho = 1)$ [1] by considering equations

$$t_{i\to a}^{(l)} \begin{cases} \frac{i_{a\to i}^{(l-1)} A_i^1}{i_{a\to i}^{(l-1)} A_i^1 + A_i^0 - \rho A_i^1 A_i^0} & \text{if } y_{i,a} = \neg x_i \\ \\ \frac{i_{a\to i}^{(l-1)} A_i^0}{i_{a\to i}^{(l-1)} A_i^0 + A_i^1 - \rho A_i^1 A_i^0} & \text{if } y_{i,a} = x_i \end{cases} \tag{7}$$

We do not have an interpretation of the intermediate cases of ρ as belief systems.

2 The Phase Diagram of 3-SAT

Early work on developing 3-SAT heuristics discovered that as α is increased, the problem changes from being easy to solve to extremely hard, then again relatively easy when the formulae are almost certainly UNSAT. It was natural to expect that a sharp phase boundary between SAT and UNSAT phases in the limit of large N accompanies this "easy-hard-easy" observed transition, and the £nite-size scaling results of [7] con£rmed this. Their work placed the transition at about $\alpha = 4.2$. Monasson and Zecchina [8] soon showed, using the replica method from statistical mechanics, that the phase transition to be expected had unusual characteristics, including "frozen variables" and a highly nonuniform distribution of solutions, making search dif£cult. Recent technical advances have made it possible to use simpler cavity mean £eld methods to pinpoint the SAT/UNSAT boundary at $\alpha = 4.267$ and suggest that the "hard-SAT" region in which the solution space becomes inhomogeneous begins at about $\alpha = 3.92$. These calculations also predicted a speci£c solution structure (termed 1-RSB for "one step replica symmetry-breaking") [1, 2] in which the satis£able con£gurations occur in large clusters, maximally separated from each other. Two types of frozen variables are predicted, one set which take the same value in all clusters and a second set whose value is £xed within a particular cluster. The remaining variables are "paramagnetic" and can take either value in some of the states of a given cluster. A careful analysis of the 1-RSB solution has subsequently shown that this extreme structure is only stable above $\alpha = 4.15$. Between 3.92 and 4.15 a wider range of cluster sizes, and wide range of inter-cluster Hamming distances are expected.[9] As a result, we expect the values $\alpha = 3.9$, 4.15 and 4.267 to separate regions in which the nature of the 3-SAT decision problem is distinctly different.

[1] This interpolation has also been considered and implemented by R. Zecchina and co-workers.

"Survey-induced decimation" consists of using SP to determine the variable most likely to be frozen, then setting that variable to the indicated frozen value, simplifying the formula as a result, updating the SP calculation, and repeating the process. For $\alpha < 3.9$ we expect SP to discover that all spins are free to take on more than one value in some ground state, so no spins will be decimated. Above 3.9, SP ideally should identify frozen spins until all that remain are paramagnetic. The depth of decimation, or fraction of spins reminaing when SP sees only paramagnetic spins, is thus an important characteristic. We show in Fig. 1 the fraction of spins remaining after survey-induced decimation for values of α from 3.85 to 4.35 in hundreds of formulae with $N = 10,000$. The error bars show the standard deviation, which becomes quite large for large values of α. To the left of $\alpha = 4.2$, on the descending part of the curves, SP reaches a paramagnetic state and halts. On the right, or ascending portion of the curves, SP stops by simply failing to converge.

Fig 1 also shows how different the behavior of BP and the hybrids between BP and SP are in their decimation behavior. We studied BP ($\rho = 0$), underrelaxed SP ($\rho = 0.95$), SP, and overrelaxed SP ($\rho = 1.05$). BP and underrelaxed SP do not reach a paramagnetic state, but continue until the formula breaks apart into clauses that have no variables shared between them. We see in Fig. 1 that BP stops working at roughly $\alpha = 3.9$, the point at which SP begins to operate. The underrelaxed SP behaves like BP, but can be used well into the RSB region. On the rising parts of all four curves in Fig 1, the scheme halted as the surveys ceased to converge. Overrelaxed SP in Fig. 1 may give reasonable recommendations for simpli£cation even on formulae which are likely to be UNSAT.

3 Some Background on WSAT

Next we consider WSAT, the random walk-based search routine used to £nish the job of exhibiting a satisfying con£guration after SP (or some other decimation advisor) has simpli£ed the formula. The surprising power exhibited by SP has to some extent obscured the fact that WSAT is itself a very powerful tool for solving constraint satisfaction problems, and has been widely used for this. Its running time, expressed in the number of walk steps required for a successful search is also useful as an informal de£nition of the complexity of a logical formula. Its history goes back to Papadimitriou's [10] observation that a subtly biased random walk would with high probability discover satisfying solutions in the simpler 2-SAT problem after, at worst, $O(N^2)$ steps. His procedure was to start with an arbitary assignment of values to the binary variables, then reverse the sign of one variable at a time using the following random process:

- select an unsatis£ed clause at random
- select at random a variable that appears in the clause
- reverse that variable

This procedure, sometimes called RWalkSAT, works because changing the sign of a variable in an unsatis£ed clause always satis£es that clause and, at £rst, has no net effect on other clauses. It is much more powerful than was proven initially. Two recent papers [12, 13]. have argued analytically and shown experimentally that Rwalksat £nds satisfying con£gurations of the variables after a number of steps that is proportional to N for values of α up to roughly 2.7. after which this cost increases exponentially with N.

The second trick in WSAT was introduced by Kautz and Selman [11]. They also choose an unsatis£ed clause at random, but then reverse one of the "best" variables, selected at random, where "best" is de£ned as causing the fewest satis£ed clauses to become unsatis£ed. For robustness, they mix this greedy move with random moves as used in RWalkSAT, recommending an equal mixture of the two types of moves. Barthel *et al.*[13] used these two moves in numerical experiments, but found little improvement over RWalkSAT.

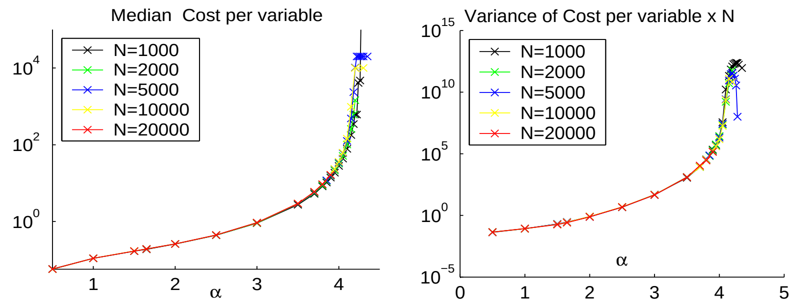
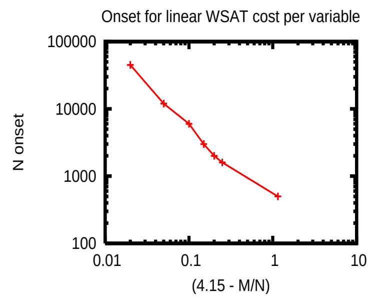


Figure 2: (a) Median of WSAT cost per variable in 3-SAT as a function of α. (b) Variance of WSAT cost, scaled by N.

There is a third trick in the most often used variant of WSAT, introduced slightly later [14]. If any variable in the selected unsatis£ed clause can be reversed without causing any other clauses to become unsatis£ed, this "free" move is immediately accepted and no further exploration is required. Since we shall show that WSAT works well above $\alpha = 2.7$, this third move apparently gives WSAT its extra power. Although these moves were chosen by the authors of WSAT after considerable experiment, we have no insight into why they should be the best choices.

In Fig. 2a, we show the median number of random walk steps per variable taken by the standard version of WSAT to solve 3-SAT formulas at values of α ranging from 0.5 to 4.3 and for formulae of sizes ranging from $N = 1000$ to $N = 20000$. The cost of WSAT remains linear in N well above $\alpha = 3.9$. WSAT cost distributions were collected on at least 1000 cases at each point. Since the distributions are asymmetric, with strong tails extending to higher cost, it is not obvious that WSAT cost is, in the statistical mechanics language, self-averaging, or concentrated about a well-de£ned mean value which dominates the distribution as $N \to \infty$. To test this, we calculated higher moments of the WSAT cost distribution and found that they scale with simple powers of N. For example, in Fig. 2b, we show that the variance of the WSAT cost per variable, scaled up by N, is a well-de£ned function of α up to almost 4.2. The third and fourth moments of the distribution (not shown) also are constant when multiplied by N and by N^2, respectively. The WSAT cost per variable is thus given by a distribution which concentrates with increasing N in exactly the way that a process governed by the usual laws of large numbers is expected to behave, even though the typical cost increases by six orders of magnitude as we move from the trivial cases to the critical regime.

A detailed analysis of the cost distributions which we observed will be published elsewhere but we conclude that the median cost of solving 3-SAT using the WSAT random walk search, as well as the mean cost if that is well-de£ned, remains linear in N up to $\alpha = 4.15$, coincidentally the onset of 1-RSB. In the 1-RSB regime, the WSAT cost per variable distributions shift to higher values as N increases, and an exponential increase in cost with N is likely. Is 4.15 really the endpoint for WSAT's linearity, or will the search cost per variable converge at still larger values of N which we could not study? We de£ne a rough estimate of $N_{onset}(\alpha)$ by study of the cumulative distributions of WSAT cost as the value of N for a given α above which the distributions cross at a £xed percentile. Plotting $log(N_{onset})$ against $log(4.15 - \alpha)$ in Fig. 3, we £nd strong indication that 4.15 is indeed an asymptote for WSAT.



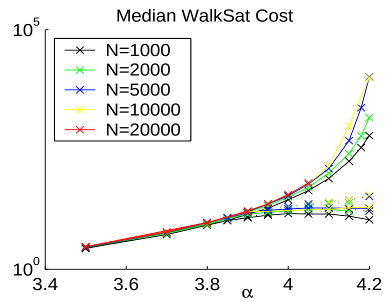


Figure 3: Size N at which WSAT cost is linear in N as function of $4.15 - \alpha$.
Figure 4: WSAT cost, before and after SP-guided decimation.

4 Practical Aspects of SP + WSAT

The power of SP comes from its use to guide decimation by identifying spins which can be frozen while minimally reducing the number of solutions that can be constructed. To assess the complexity of the reduced formulae that decimation guided in this way produces we compare, in Fig. 4, the median number of WSAT steps required to £nd a satisfying con£guration of the variables before and after decimation. To a rough approximation, we can say that SP caps the cost of £nding a solution to what it would be at the entry to the critical regime. There are two factors, the reduction in the number of variables that have to be searched, and the reduction of the distance the random walk must traverse when it is restricted to a single cluster of solutions. In Fig. 2c the solid lines show the WSAT costs divided by N, the original number of variables in each formula. If we instead divide the WSAT cost after decimation by the number of variables remaining, the complexity measure that we obtain is only a factor of two larger, as shown by the dotted lines. The relative cost of running WSAT without bene£t of decimation is 3-4 decades larger.

We measured the actual compute time consumed in survey propagation and in WSAT. For this we used the Zecchina group's version 1.3 survey propagation code, and the copy of WSAT (H. Kautz's release 35, see [15]) that they have also employed. All programs were run on a Pentium IV Xeon 3GHz dual processor server with 4GB of memory, and only one processor busy. We compare timings from runs on the same 100 formulas with $N = 10000$ and $\alpha = 4.1$ and 4.2 (the formulas are simply extended slightly for the second case). In the £rst case, the 100 formulas were solved using WSAT alone in 921 seconds. Using SP to guide decimation one variable at a time, with the survey updates performed locally around each modi£ed variable, the same 100 formulas required 6218 seconds to solve, of which only 31 sec was spent in WSAT.

When we increase alpha to 4.2, the situation is reversed. Running WSAT on 100 formulas with $N = 10000$ required 27771 seconds on the same servers, and would have taken even longer if about half of the runs had not been stopped by a cutoff without producing a satisfying con£guration. In contrast, the same 100 formulas were solved by SP followed with WSAT in 10,420 sec, of which only 300 seconds were spent in WSAT. The cost of SP does not scale linearly with N, but appears to scale as N^2 in this regime. We solved 100 formulas with $N = 20,000$ using SP followed by WSAT in 39643 seconds, of which 608 sec was spent in WSAT. The cost of running SP to decimate roughly half the spins has quadrupled, while the cost of the £nal WSAT runs remained proportional to N.

Decimation must stop short of the paramagnetic state at the highest values of α, to avoid having SP fail to converge. In those cases we found that WSAT could sometimes £nd satisfying con£gurations if started slightly before this point. We also explored partial decimation as a means of reducing the cost of WSAT just below the 1-RSB regime, but found that decimation of small fractions of the variables caused the WSAT running times to be highly unpredictable, in many cases increasing strongly. As a result, partial decimation does not seem to be a useful approach.

5 Conclusions and future work

The SP and related algorithms are quite new, so programming improvements may modify the practical conclusions of the previous section. However, a more immediate target for future work could be the WSAT algorithms. Further directing its random choices to incorporate the insights gained from BP and SP might make it an effective algorithm even closer to the SAT/UNSAT transition.

Acknowledgments

We have enjoyed discussions of this work with members of the replica and cavity theory community, especially Riccardo Zecchina, Alfredo Braunstein, Marc Mezard, Remi Monasson and Andrea Montanari. This work was performed in the framework of EU/FP6 Integrated Project EVERGROW (www.evergrow.org), and in part during a Thematic Institute supported by the EXYSTENCE EU/FP5 network of excellence. E.A. acknowledges support from the Swedish Science Council. S.K. and U.G. are partially supported by a US-Israeli Binational Science Foundation grant.

References

[1] Mézard M., Parisi G. & Zecchina R.. (2002) Analytic and Algorithmic Solutions of Random Satis£ability Problems. *Science*, **297**:812-815

[2] Mézard M. & Zecchina R. (2002) The random K-satis£ability problem: from an analytic solution to an ef£cient algorithm. *Phys. Rev. E* **66**: 056126.

[3] Braunstein A., Mezard M. & Zecchina R., "Survey propagation: an algorithm for satis£ability", arXiv:cs.CC/0212002 (2002).

[4] Parisi G. (2003), On the probabilistic approach to the random satis£ability problem, Proc. SAT 2003 and arXiv:cs:CC/0308010v1 .

[5] Braunstein A. and Zecchina R., (2004) Survey Propagation as Local Equilibrium Equations. arXiv:cond-mat/0312483 v5.

[6] Pearl J. (1988) *Probabilistic Reasoning in Intelligent Systems*, 2nd Edition, Kauffmann.

[7] Kirkpatrick S. & Selman B. (1994) Critical Behaviour in the Sati£ability of Random Boolean Expressions. *Science* **264**: 1297-1301.

[8] Monasson R. & Zecchina R. (1997) Statistical mechanics of the random K-Sat problem. *Phys. Rev. E* **56**: 1357–1361.

[9] Montanari A., Parisi G. & Ricci-Tersenghi F. (2003) Instability of one-step replica-symmetric-broken phase in satis£ability problems. cond-mat/0308147.

[10] Papadimitriou C.H. (1991). In *FOCS 1991*, p. 163.

[11] Selman B. & Kautz H.A. (1993) In *Proc. AAAI-93* **26**, pp. 46-51.

[12] Semerjian G. & Monasson R. (2003). *Phys Rev E* **67**: 066103.

[13] Barthel W., Hartmann A.K. & Weigt M. (2003). *Phys. Rev. E* **67**: 066104.

[14] Selman B., Kautz K. & Cohen B. (1996) Local Search Strategies for Satis£ability Testing. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science* **26**.

[15] `http://www.cs.washington.edu/homes/kautz/walksat/`