A Second Order Parameter for 3SAT

Tuomas W. Sandholm
Department of Computer Science
University of Massachusetts
Amherst, Massachusetts 01003
sandholm@cs.umass.edu

Abstract

The 3-satisfiability problem (3SAT) has had a central role in the study of complexity. It was recently found that 3SAT instances transition sharply from satisfiable to nonsatisfiable as the ratio of clauses to variables increases. Because this phase transition is so sharp, the ratio – an order parameter – can be used to predict satisfiability. This paper describes a second order parameter for 3SAT. Like the classical order parameter, it can be computed in linear time, but it analyzes the structure of the problem instance more deeply. We present an analytical method for using this new order parameter in conjunction with the classical one to enhance satisfiability prediction accuracy. The assumptions of the method are verified by rigorous statistical testing. The method significantly increases the satisfiability prediction accuracy over using the classical order parameter alone. Hardness – i.e. how long it takes to determine satisfiability – results for one complete and one incomplete algorithm from the literature are also presented as a function of the two order parameters. The importance of new order parameters lies in the fact that they refine the locating of satisfiable vs. nonsatisfiable and hard vs. easy formulas in the space of all problem instances by adding a new dimension in the analysis.[1]

1 Introduction

The 3-satisfiability problem (3SAT) is the decision problem of whether a satisfying truth assignment exists for variables in a 3CNF formula. A 3CNF formula is a conjunct of clauses that are disjuncts of three literals. A literal is a negated or non-negated variable. The formula is satisfiable (SAT), if the variables can be assigned Boolean values so that at least one literal in each clause is true. For example, the formula $(x_1 \vee x_2 \vee \sim x_3) \wedge (x_1 \vee x_3 \vee x_4) \wedge (\sim x_1 \vee \sim x_2 \vee \sim x_4)$ is satisfiable by the truth assignment x_1 = true, x_2 = true, x_3 = true, x_4 = false – among others. We will call the number of variables v and the number of clauses c. In the formula above, $v = 4$ and $c = 3$.

The importance of 3SAT lies in the fact that it is structurally simple, yet *NP*-complete, i.e. any problem in *NP* can be reduced to 3SAT in polynomial time. A decision algorithm for 3SAT can therefore (in theory) be used to solve any problem in *NP*, and the complexity of solving an arbitrary problem in *NP* is at most a polynomial factor greater than the complexity of the algorithm for 3SAT. Every exact algorithm for 3SAT requires exponential time in general, unless $P = NP$. Nevertheless, most 3SAT instances can be solved quickly, because the hard instances – i.e. those that require a long time to determine satisfiability – are rare (Mitchell et al, 1992; Crawford and Auton, 1993). Given a problem instance, it would be useful to know how long determining its satisfiability will take. This is especially important because even with problem instances of the same size, the running time of a satisfiability determining algorithm can vary several orders of magnitude (Crawford and Auton, 1993). This paper presents a method to combine a new order parameter with a classical one to estimate hardness. If the problem instance is predicted to be hard, the user may wish to avoid running the satisfiability determining algorithm, and use a satisfiability probability estimate instead. We present a statistically backed analytical method that uses the two order parameters to estimate this probability.

In Section 2 we review the classical order parameter for 3SAT along with central results related to it. Section 3 presents the new order parameter and empirical data about its mean and variance. In Section 4 we describe an analytical method for predicting satisfiability based on the two order parameters. We also present statistical evidence that supports the validity of the assumptions that underlie this method. Section 5 compares the accuracy of the new satisfiability estimation method to one that only uses the classical order parameter. In Section 6 we analyze the effect of the new order parameter on hardness. We do this with respect to one complete algorithm – i.e. one that is guaranteed to halt and return the right answer – and one incomplete algorithm from the literature. Section 7 concludes and presents future applications and research directions.

2 The classical order parameter β

Cheeseman et al. (1991) and Mitchell et al. (1992) present the $\beta = c/v$ order parameter as a hardness estimator for 3SAT. Usually, the formulas with low β are satisfiable, but the ones with high β are not. On average the hard instances occur in the critical region ($\beta \approx 4.3$), where the instances undergo a phase transition from satisfiable to nonsatisfiable. Crawford and Auton (1993) empirically refine the location of this phase transition by the observation that 50% of the formulas are satisfiable on the line $c = 4.24v + 6.21$. As the number of variables increases, the phase transition becomes more pronounced, and the

[1] This work was supported the National Science Foundation under Grant No. IRI-9523419. The content does not necessarily reflect the position or the policy of the Government and no official endorsement should be inferred. An early version of this paper appeared in (Sandholm and Lesser, 1994a).

satisfiability becomes sharper. This hardness peak has been mostly studied for variants of the complete Davis-Putnam algorithm (Davis and Putnam, 1960; Mitchell et al, 1992; Crawford and Auton, 1993; Yugami, 1995).

We repeated some of these tests to determine the satisfiability probability as a function of β. We ran tests for $v \in \{50, 100, 150\}$, $\beta \in \{0.1, 0.2, ..., 9.0\}$. At each one of these 270 experiment points, we generated 500 3CNF formulas and determined their satisfiability with a complete algorithm similar to that of Crawford and Auton (1993). The results, displayed in Figure 1, agree with the literature.

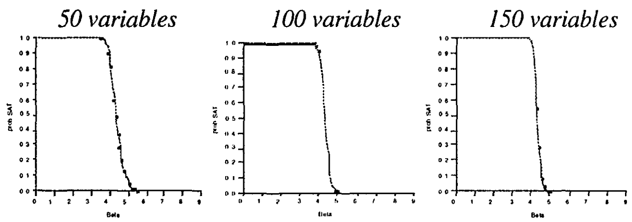

Figure 1. Satisfiability probability as a function of the β-parameter: $p^{SAT}(\beta)$.

Throughout this paper, we use formulas that we generated by the standard method (Mitchell et al, 1992) for constructing hard 3SAT instances: for every clause, pick three variables randomly disallowing duplicates, and then negate each variable independently with probability 0.5. We use the same set of instances in all experiments, except in Section 5, where we use a separate test set to evaluate our satisfiability prediction method.

3 The new order parameter Δ

In our search for new 3SAT order parameters we examined a set of intuitively plausible predictors that were computable in linear time. During exploratory data analysis, just one of these turned out to predict satisfiability. We call this one Δ, and define it as follows.

$$\Delta = \sum_{x \in Variables} |pos_x - neg_x|,$$

where pos_x is the number of positive (non-negated) occurrences of variable x, and neg_x is the number of negated occurrences of variable x in a formula. Given a 3SAT problem instance, the value of Δ can be computed in $O(c + v)$ time.

The intuition behind Δ is that if the difference of positive and negative occurrences is large, assigning the variable the value with more occurrences satisfies a large number of clauses. On the other hand, if the numbers of positive and negative occurrences are close to each other, assigning the variable one of the values leaves a large number of clauses that will have to be satisfied in the future using a reduced number of variables.

Figure 2 illustrates the fact that satisfiable formulas tend to have a larger Δ than nonsatisfiable ones at a given value of β. This difference is statistically significant. For example, in the regions $4 \le \beta \le 5$ in Figure 2, all of the sample means for satisfiable formulas are greater than those for nonsatisfiable ones. Let us look at any one of the three panes in Figure 2. Assuming that the true means are equal, there would be a 0.5 chance of each sample mean for satisfiable formulas being greater than that for nonsatisfiable ones. So the chance of that happening at all eleven points (4.0, 4.1,...,5.0) is $1/2^{11} < 0.0005$. Therefore, the difference is statistically significant even at the 0.0005 level in those regions.

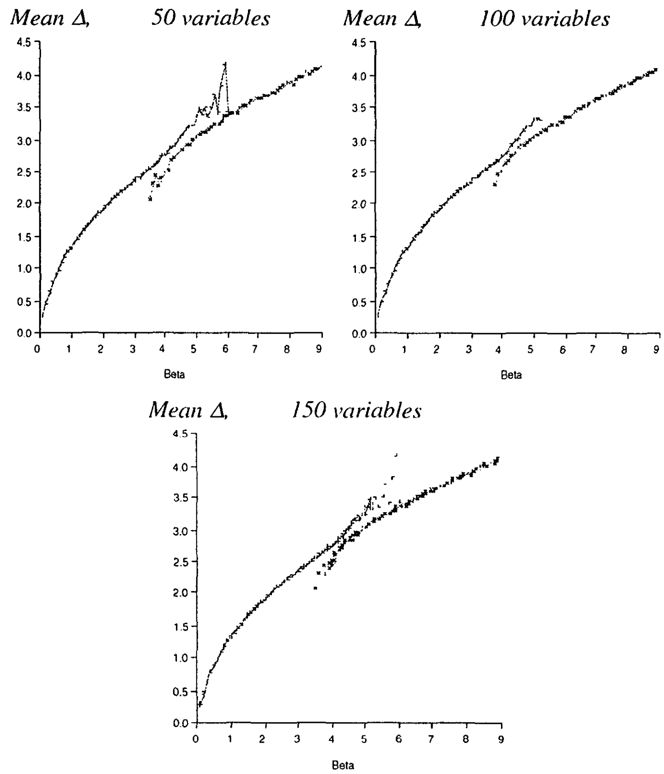

Figure 2. Mean Δ as a function of β. The upper curve is for satisfiable formulas (μ^{SAT}) and the lower is for nonsatisfiable ones (μ^{NON}). The high variability at the ends of the curves is due to noise caused by the small number of formulas with extreme β values for their class (i.e. SAT or NON).

For a given β, the difference $\mu^{SAT} - \mu^{NON}$ becomes smaller as the number of variables increases. One might conclude that Δ's discriminatory power drops as the number of variables increases, because this difference becomes insignificant. However, as Figure 3 shows, the variance of Δ for satisfiable formulas ($Var[\Delta^{SAT}]$) and the variance of Δ for nonsatisfiable formulas ($Var[\Delta^{NON}]$) *decrease* drastically with increasing v. Therefore, the difference remains significant, and Δ's discriminatory power remains intact.

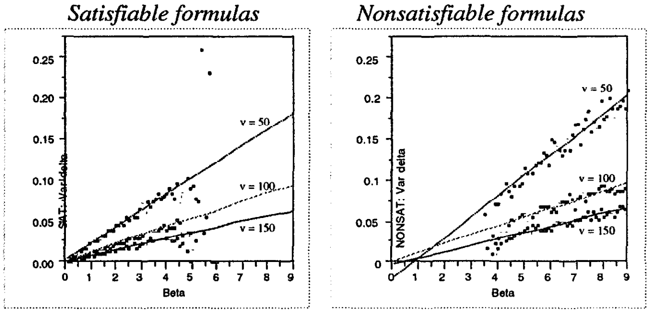

Figure 3. Variance of Δ as a function of β. The top line is a regression line for the 50 variable case, the middle line is for 100 variables and the bottom line is for 150.

The empirical regression line of the variance for the satisfiable formulas with 50 variables is Var[Δ] = 0.020 β + 0.0019; r^2 = 0.95. For 100 variables it is Var[Δ] = 0.010 β + 0.00026; r^2 = 0.97, and for 150 variables it is Var[Δ] = 0.0068 β + 0.00057; r^2 = 0.97. For the nonsatisfiable formulas with 50 variables the regression line is Var[Δ] = 0.025 β - 0.019; r^2 = 0.93. For 100 variables it is Var[Δ] = 0.010 β + 0.0026; r^2 = 0.88, and for 150 variables it is Var[Δ] = 0.0076 β - 0.0018; r^2 = 0.90. Based on these observations, we use the following formula for approximation for both satisfiable and nonsatisfiable formulas:

$$\mathrm{Var}[\Delta^{SAT}] \approx \mathrm{Var}[\Delta^{NON}] \approx \beta / v.$$

4 Predicting satisfiability with β and Δ

This section develops a new satisfiability prediction method based on β and Δ. The method relies on the assumption that Δ^{SAT} and Δ^{NON} are roughly normally distributed at a given v and β. These assumptions are verified via rigorous statistical testing in Appendix A. Here we present the new method for predicting satisfiability. Let $n(x, \mu, \sigma^2)$ be a normal probability density function (pdf) with mean μ and variance σ^2. Let the pdf corresponding to Δ at a certain v and a certain β be $f_{v,\beta}(\Delta)$ for satisfiable and nonsatisfiable formulas combined. Similarly, let the pdf for satisfiable formulas be $f_{v,\beta}^{SAT}(\Delta)$, and let the pdf for nonsatisfiable formulas be $f_{v,\beta}^{NON}(\Delta)$. Now

$$f_{v,\beta}(\Delta) = c_1 f_{v,\beta}^{SAT}(\Delta) + c_2 f_{v,\beta}^{NON}(\Delta),$$

where $c_1 + c_2 = 1$, $c_1 \geq 0$, and $c_2 \geq 0$. Another constraint on c_1 is based on the β estimator (Fig. 1.):

$$p^{SAT}(\beta) = \frac{\int_{-\infty}^{\infty} c_1 f_{v,\beta}^{SAT}(\Delta) d\Delta}{\int_{-\infty}^{\infty} f_{v,\beta}(\Delta) d\Delta} = \frac{c_1 \int_{-\infty}^{\infty} f_{v,\beta}^{SAT}(\Delta) d\Delta}{\int_{-\infty}^{\infty} f_{v,\beta}(\Delta) d\Delta} = \frac{c_1 \cdot 1}{1} = c_1$$

Thus $c_2 = 1 - p^{SAT}(\beta)$. We use the normal approximation – that we statistically validate in Appendix A – for $f_{v,\beta}^{SAT}(\Delta)$ and for $f_{v,\beta}^{NON}(\Delta)$:

$$f_{v,\beta}^{SAT}(\Delta) \approx n(\Delta, \mu^{SAT}, Var[\Delta^{SAT}]) \approx n(\Delta, \mu^{SAT}, \beta/v),$$
$$f_{v,\beta}^{NON}(\Delta) \approx n(\Delta, \mu^{NON}, Var[\Delta^{NON}]) \approx n(\Delta, \mu^{NON}, \beta/v)$$

Now, the satisfiability probability

$$p_{v,\beta}^{SAT}(\Delta) = \frac{c_1 f_{v,\beta}^{SAT}(\Delta)}{f_{v,\beta}(\Delta)} = \frac{c_1 f_{v,\beta}^{SAT}(\Delta)}{c_1 f_{v,\beta}^{SAT}(\Delta) + c_2 f_{v,\beta}^{NON}(\Delta)} =$$
$$\frac{p^{SAT}(\beta) \cdot f_{v,\beta}^{SAT}(\Delta)}{p^{SAT}(\beta) \cdot f_{v,\beta}^{SAT}(\Delta) + [1 - p^{SAT}(\beta)] \cdot f_{v,\beta}^{NON}(\Delta)} =$$
$$\frac{p^{SAT}(\beta) \cdot n(\Delta, \mu^{SAT}, \beta/v)}{p^{SAT}(\beta) \cdot n(\Delta, \mu^{SAT}, \beta/v) + [1 - p^{SAT}(\beta)] \cdot n(\Delta, \mu^{NON}, \beta/v)}$$

Figure 5 shows the contours of this satisfiability probability based on the empirical μ^{SAT} and μ^{NON} of Figure 2. The use of Δ refines satisfiability probability prediction compared to the use of β alone, which would imply vertical contours, i.e. Δ would be assumed to have no effect. The gain of using Δ is largest in the phase transition region. The contours are irregular due to the use of empirical μ^{SAT} and μ^{NON}. This is especially apparent in the edge regions of the 50 variable case due to small sample sizes of satisfiable formulas for β > 5, and nonsatisfiable formulas for β < 4.

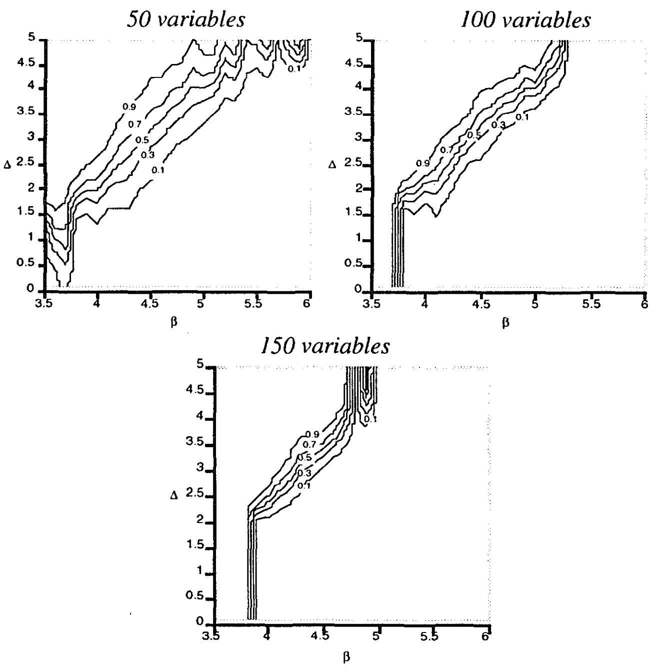

Figure 5. Satisfiability probability contours $p_{v,\beta}^{SAT}(\Delta)$.

5 Evaluation of satisfiability prediction

We compared the prediction accuracy of β alone to that of β and Δ together using a test set consisting of 100 formulas at each $\beta \in \{0.1, 0.2, ..., 9.0\}$ for each $v \in \{50, 100, 150\}$. These formulas were generated separately from the ones that we used to collect statistics for the prediction methods, but with the same algorithm. We determined the true satisfiability (1 or 0) of each formula using a complete algorithm. The satisfiability prediction error metric was the average absolute value of the difference of the satisfiability prediction ($p^{SAT}(\beta)$ or $p^{SAT}_{v,\beta}(\Delta)$) and the true satisfiability. The prediction error of both predictors was largest in the phase transition region of β. At some point in this region the $p^{SAT}(\beta)$ predictor has a prediction error of 0.5, i.e. it is no more accurate than random guessing. The new predictor $p^{SAT}_{v,\beta}(\Delta)$ had statistically significantly smaller prediction errors than the classical predictor $p^{SAT}(\beta)$, and the advantage of $p^{SAT}_{v,\beta}(\Delta)$ over $p^{SAT}(\beta)$ was largest in the hard region, Fig. 6. When measured at the worst β, the prediction error of $p^{SAT}_{v,\beta}(\Delta)$ was 0.424 for $v = 50$ ($c = 220$), 0.407 for $v = 100$ ($c = 430$), and 0.453 for $v = 150$ ($c = 645$). The corresponding errors for $p^{SAT}(\beta)$ were 0.500, 0.494 and 0.496.

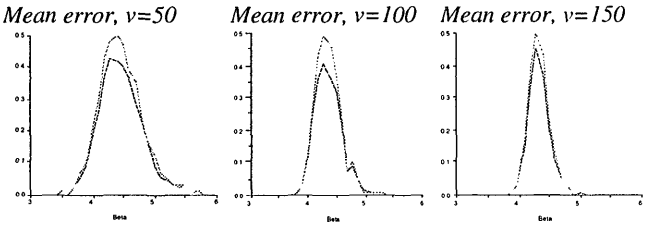

Figure 6. Prediction error of the two predictors. The higher curve is for $p^{SAT}(\beta)$ and the lower for $p^{SAT}_{v,\beta}(\Delta)$.

To make sure that the advantage of using the Δ estimator does not diminish with increasing numbers of variables, we ran tests for 200 and 250 variables at the hardest – as suggested by Crawford and Auton's (1993) formula $c = 4.24v + 6.21$ – region of β. The prediction errors for $p^{SAT}_{v,\beta}(\Delta)$ were 0.417 ($v = 200$, $c = 854$) and 0.413 ($v = 250$, $c = 1066$). The corresponding errors for $p^{SAT}(\beta)$ were 0.494 and 0.474. These results lead us to conclude that the advantage of using the Δ estimator does not diminish as the number of variables increases.

6 Relationships between Δ and hardness

The average hardness of formulas is highest in the phase transition region of β (Cheeseman et al., 1991). This led us to test whether the Δ parameter also impacts hardness. At each $v \in \{50, 100, 150\}$ and each $\beta \in \{0.1, 0.2, ..., 9,0\}$ we divided the formulas into 5 equal size buckets according to Δ. We did this division for each of the following experiments separately. The average hardness of the formulas in each bucket gave one data point for Figures 7-10.

First, we analyzed the effect of Δ on the number of steps required by a complete algorithm that uses the two primary heuristics of Crawford and Auton (1993). To begin with, we divided the 500 formulas into two categories: satisfiable and nonsatisfiable ones. On average, satisfiable formulas become harder as β increases - as expected from the literature (Mitchell et al., 1992). This is because with more clauses – i.e. constraints – more of the variable assignments are nonsatisfiable and thus more backtracking is required. The contours are slanted to the right implying that, on average, satisfiable formulas become easier as Δ increases, Fig. 7. Due to reasons that were discussed in Section 3, increasing Δ promotes satisfiability. A larger number of variable assignments will be satisfiable and thus less backtracking is required.

Figure 8 shows that, on average, nonsatisfiable clauses become harder as β decreases, as expected (Mitchell et al., 1992). With fewer constraints, early pruning of branches becomes more difficult and the search proceeds to deep levels. Because no variable assignments satisfy the formula, deep backtracking occurs leading to large time costs. On average, the nonsatisfiable low-β formulas are harder than the formulas in the phase transition region. Yet, these formulas are so rare that the average hardness over satisfiable and nonsatisfiable formulas together is highest in the phase transition region.

The contours are tilted to the right implying that, on average, nonsatisfiable formulas become harder as Δ increases. This can be explained by the size of the search trees when the algorithm branches on an arbitrary variable x. For low Δ, x occurs evenly ($pos_x \approx neg_x$), and there are two subtrees with roughly equal numbers of unsatisfied clauses: moderately deep backtracking is required. For high Δ, x occurs unevenly. For the more frequent value of x, the remaining subtree has very few unsatisfied clauses (constraints) and will require deep backtracking. For the less frequent value of x, the remaining subtree will have very many unsatisfied clauses and a contradiction will be found with shallow backtracking. Because no variable assignments satisfy the formula, both trees will have to be explored in either case. Time is exponential in search depth; hence the time cost incurred by deep backtracking in one subtree offsets any time saved by shallow backtracking in the other subtree. Therefore, search time tends to be higher with high Δ.

When the satisfiable and nonsatisfiable formulas are merged, the average hardness is highest around the threshold $\beta = 4.3$ as expected, Fig. 9. The contours have a slight tilt to the right implying that below the threshold, on average, formulas become easier as Δ increases, and above the threshold, formulas become harder as Δ increases. These results are intuitive, because the characteristics of satisfiable formulas prevail for low β where they are dominant, and for high β, the characteristics of nonsatisfiable formulas prevail.

The above results are specific to our complete algorithm. In order to get an indication of whether the results hold in general, we also analyzed hardness using the BREAKOUT (Morris 1993) satisfiability determining algorithm. BREAKOUT is incomplete, i.e. if the formula is not satisfiable, the algorithm never halts, but if the formula is satisfiable, the algorithm may halt, proving satisfiability, or it might not halt. BREAKOUT was run on the satisfiable formulas, and its number of steps was recorded. If it had not found a solution by 20,000 steps, the run was aborted and 20,000 was recorded as the required number of steps. As expected, the average hardness of satisfiable formulas increases as β increases, Fig. 10. This is because fewer variable assignments satisfy the formula and thus more search is needed. As with the complete algorithm, on average, problem instances become easier as Δ increases for a given β, because more variable assignments satisfy the formula and thus less search is required.

7 Conclusions and future research

We presented a new order parameter, Δ, for 3SAT. Like the classical order parameter, it can be computed in linear time, and it can be used to predict satisfiability and hardness. To estimate the satisfiability probability, we modeled the distribution of Δ for satisfiable formulas, and the distribution of Δ for nonsatisfiable formulas, with normal distributions. Based on these models we derived an analytical formula for the satisfiability probability. The new estimator is significantly more accurate than the classical estimator that uses β alone. The difference is greatest in the phase transition region of β.

For both a complete and an incomplete algorithm, on average, satisfiable formulas become easier as Δ increases. On the other hand, for the complete algorithm, nonsatisfiable formulas become harder as Δ increases. The incomplete algorithm never halts on nonsatisfiable formulas. For satisfiable and nonsatisfiable formulas combined, average hardness using the complete algorithm decreases with increasing Δ under the threshold of β, and increases with increasing Δ above the threshold of β.

The results provide a deeper understanding of 3SAT, but they also have other uses. If a satisfiability result is required under real-time constraints, giving the estimate generated in linear time may be more appropriate than trying to run a satisfiability determining algorithm. Using linear time estimates and running complete satisfiability determining algorithms are two ends of a spectrum. A continuum of algorithms can be developed such that allowing more time enhances satisfiability prediction accuracy: the rigid distinction between order-parameter-based prediction and satisfiability determining algorithms is somewhat artificial.

The satisfiability estimate can also be used as a mandatory setup phase for an anytime algorithm for determining satisfiability (Sandholm and Lesser, 1994b). The new order parameter can be used to parameterize the performance profile of such an algorithm.

The hardness analysis of this paper can be used to predict the termination time of a satisfiability determining algorithm. The distribution (parameterized by v, β and Δ) of termination time of each alternative algorithm can also be used in algorithm selection. Furthermore, order parameters can be used in search heuristics – for example in variable and value ordering, in pruning branches that are unlikely to contain a solution, and to trade solution quality off against computational complexity (Zhang and Pemberton, 1994).

Our satisfiability and hardness prediction methods are based on a statistical analysis of 3SAT instances from a specific hard instance distribution. Therefore, they are not necessarily accurate for instances from a different distribution – e.g. reduced from a different problem. To use the methods with a different instance distribution, the statistical data for calibrating the methods should be collected from instances from the corresponding distribution. Nevertheless, prediction with β and Δ may be more robust against variations in the instance distribution than prediction with β alone because the combined method performs a deeper analysis with an added orthogonal perspective.

The BREAKOUT algorithm is not necessarily efficient for hard 3SAT instance distributions, although Morris (1993) shows very promising results for easier distributions. For the studied problem instances, it was usually slower than a complete algorithm similar to that of Crawford and Auton (1993). Also, it often failed to find a truth assignment for a satisfiable formula, especially in the hard region. In the future, it would be interesting to study the hardness of other incomplete satisfiability determining algorithms such as GSAT (Selman et al., 1992) or its newer variants (Gent and Walsh, 1993; Selman and Kautz, 1993; Cha and Iwama, 1995) as a function of Δ.

The analytical formula for the satisfiability probability based on the mentioned problem instance features (v, β, and Δ) and possibly other such features is an interesting open problem. It seems difficult: even the formula that only accounts for v and β is unknown (Williams and Hogg, 1993; Yugami, 1995).

Acknowledgments

I would like to thank Victor Lesser, Neil Immerman, and Frank Klassner for their helpful comments.

References

Cha, B. and Iwama, K. 1995. Performance Test of Local Search Algorithms Using New Types of Random CNF Formulas. *14th International Joint Conf. on Artificial Intelligence*, pp. 304-310.

Cheeseman, P., Kanefsky, B. and Taylor, W. 1991. Where the Really Hard Problems Are. *12th International Joint Conference on Artificial Intelligence*, pp. 331-337.

Crawford, J. and Auton, L. 1993. Experimental Results on the Crossover Point in Satisfiability Problems. *11th National Conference on Artificial Intelligence*, pp. 21-27.

D'Agostino, R. and Stephens, M. 1986. *Goodness-of-fit techniques*. Marcel Dekker, Inc., New York.

Davis, M. and Putnam, H. 1960. A computing procedure for quantification theory. *Journal of the ACM*, 7: 201-215.

Gent, I. and Walsh, T. 1993. Towards an Understanding of Hill-climbing Procedures for SAT. *11th National Conference on Artificial Intelligence*, pp. 28-33.

Mitchell, D., Selman, B. and Levesque, H. 1992. Hard and Easy Distributions of SAT Problems. *10th National Conference on Artificial Intelligence*, pp. 459-465.

Morris, P. 1993. The Breakout Method For Escaping From Local Minima. *11th National Conf. on Artificial Intelligence*, pp. 40-45.

Sandholm, T. and Lesser, V. 1994a. A New Order Parameter for 3SAT. *AAAI Workshop on Experimental Evaluation of Reasoning and Search Methods*, pp. 57-63.

Sandholm, T. and Lesser, V. 1994b. Utility-Based Termination of Anytime Algorithms. *European Conf. on Artificial Intelligence Workshop on Decision Theory for DAI Applications*, pp. 88-99.

Selman, B. and Kautz, H. 1993. Domain-Independent Extensions to GSAT: Solving Large Structured Satisfiability Problems. *13th International Joint Conf. on Artificial Intelligence*, pp. 290-295.

Selman, B., Levesque, H. and Mitchell, D. 1992. A New Method for Solving Hard Satisfiability Problems. In *Proceedings of the Tenth National Conference on Artificial Intelligence*, pp. 440-446.

Williams, C. and Hogg, T. 1993. Extending Deep Structure. *11th National Conference on Artificial Intelligence*, pp. 152-157.

Yugami, N. 1995. Theoretical Analysis of Davis-Putnam Procedure and Propositional Satisfiability. *14th International Joint Conference on Artificial Intelligence*, pp. 282-288.

Zhang, W. and Pemberton, J. 1994. Epsilon-Transformation: Exploiting Phase Transitions to Solve Combinatorial Optimization Problems - Initial Results. *12th National Conference on Artificial Intelligence*, pp. 895-900.

Appendix A. Verifying the normality of Δ^{SAT} and Δ^{NON}

In this appendix we use a rigorous standard statistical test to show that the assumptions regarding the normality of Δ for satisfiable and nonsatisfiable formulas separately, for a given value of v and β, are valid. These assumptions were used in Section 4 to derive the analytical estimator for the satisfiability probability. We statistically verify the assumptions using the previously presented set of 3SAT instances. At each experiment point – combination of v and β – we divide the 500 formulas into satisfiable and nonsatisfiable ones. This provides one (possibly empty) set of Δ^{SAT}'s and one of Δ^{NON}'s at each experiment point. Let us call the corresponding cumulative distributions $F^{SAT}_{v,\beta}(\Delta)$ and $F^{NON}_{v,\beta}(\Delta)$.

At each point (v, β), the null hypothesis is that the distribution of Δ^{SAT} and that of Δ^{NON} are normal with the same mean and variance as those of Δ^{SAT} and Δ^{NON}. Let us call the normal distributions $n^{SAT}_{v,\beta}(\Delta)$ and $n^{NON}_{v,\beta}(\Delta)$, and their cumulative distributions $N^{SAT}_{v,\beta}(\Delta)$ and $N^{NON}_{v,\beta}(\Delta)$. Next we discuss the testing of the null hypothesis for satisfiable formulas. For nonsatisfiable ones it is analogous. To statistically test whether we can reject the null hypothesis, we use the following procedure at each point (v, β). We generate 300 samples of $n^{SAT}_{v,\beta}(\Delta)$ via Monte Carlo simulation, with each sample having as many Δ-values as there were satisfiable formulas at that point (v, β). Let us call the cumulative distribution of each such sample $\hat{N}^{SAT}_{v,\beta}(\Delta)_i$, $i \in \{1, 2, \ldots, 300\}$. For each such sample, we compute how much it deviates from the actual $N^{SAT}_{v,\beta}(\Delta)$ by using the Kolmogorov-Smirnoff metric (D'Agostino and Stephens, 1986):

$$e_i = \max_{\Delta}\left|N^{SAT}_{v,\beta}(\Delta) - \hat{N}^{SAT}_{v,\beta}(\Delta)_i\right|.$$

This provides a sampling distribution of size 300. Next we calculate the proportion of these 300 e_i's that exceed the actual deviation

$$\max_{\Delta}\left|N^{SAT}_{v,\beta}(\Delta) - F^{SAT}_{v,\beta}(\Delta)\right|.$$

This proportion is the confidence level (p-value) at which we could reject the null hypothesis. In other words, if the actual deviation were greater than almost all of the deviations in the sampling distribution, we could confidently reject the null hypothesis that the Δ^{SAT}'s came from a normal distribution. Figure 4 shows these p-values. They vary greatly, but especially in the phase transition region of β they are large enough to justify not rejecting the null hypothesis that the Δ^{SAT}'s and Δ^{NON}'s are from normal distributions.

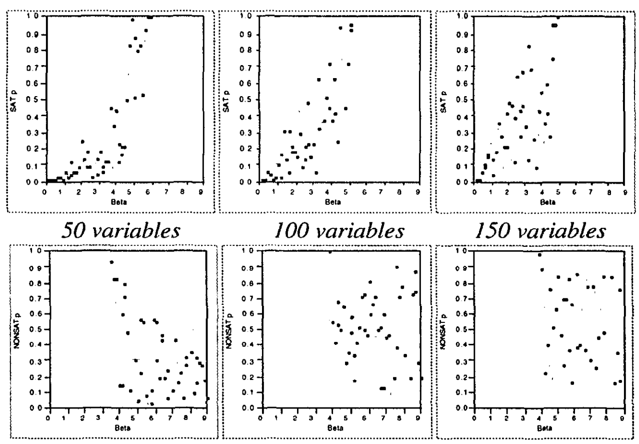

Figure 4. The figures show the p-values for the null hypothesis that the sample (one dot) is from a normal distribution with the mean and variance of the sample. The top row denotes satisfiable formulas; the bottom row nonsatisfiable ones.

Appendix B. Hardness figures

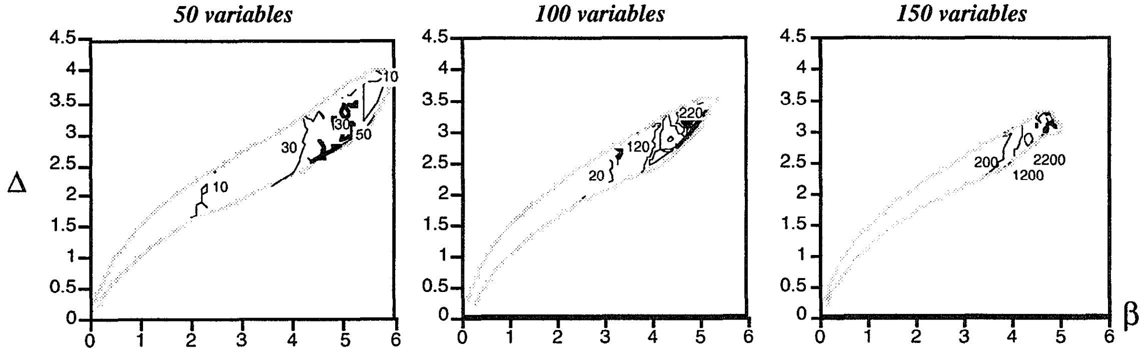

Figure 7. Contours of the steps required to determine satisfiability using a complete algorithm. Results are for satisfiable clauses. The gray lines represent the envelope in which the mean Δ's within each bucket at each v fall.

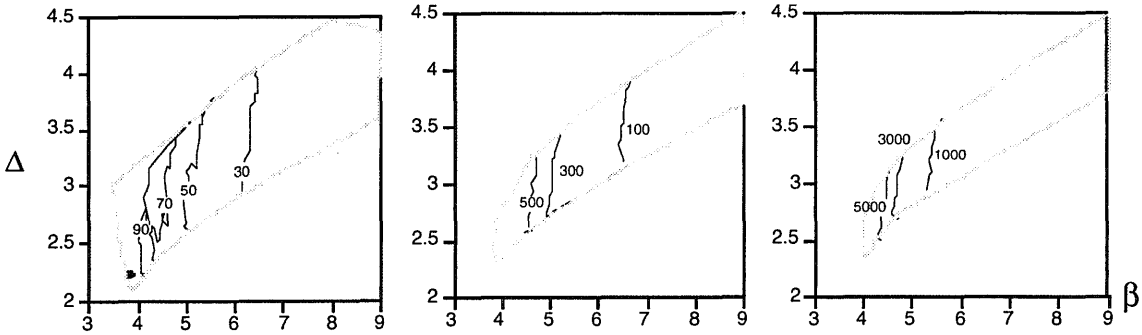

Figure 8. Contours of the steps required to determine satisfiability using a complete algorithm. Results are for nonsatisfiable clauses.

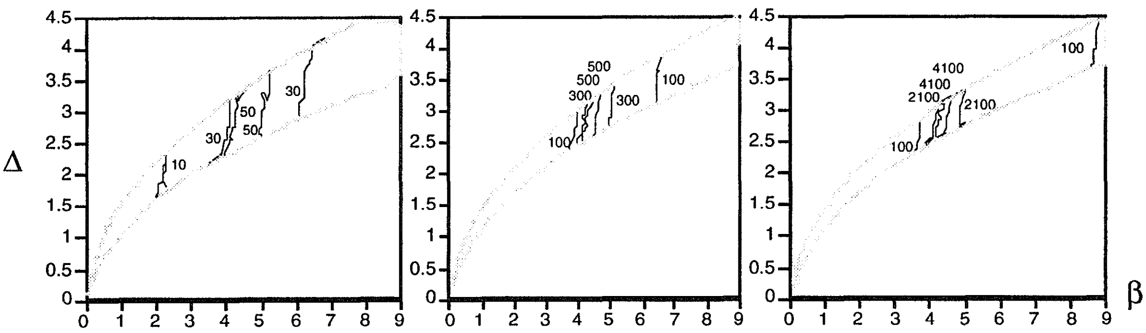

Figure 9. Contours of the steps required to determine satisfiability using a complete algorithm. Results are for all clauses (satisfiable and nonsatisfiable).

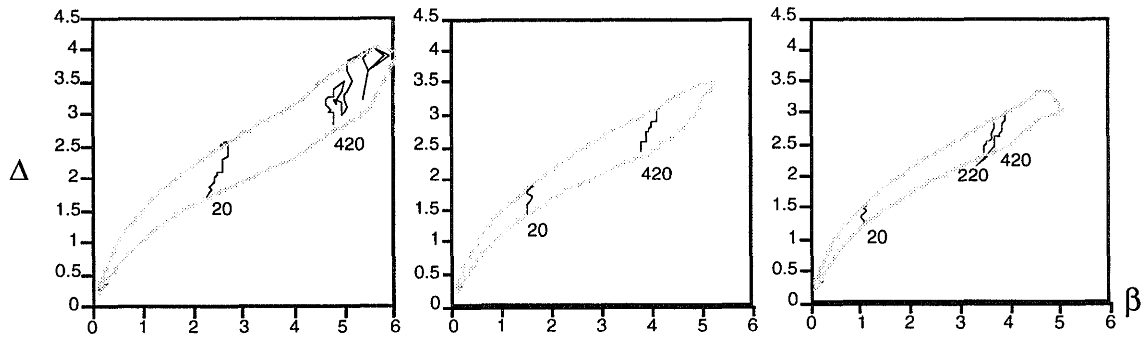

Figure 10. Contours of the steps required to determine satisfiability with the incomplete BREAKOUT algorithm. Results are for satisfiable clauses; the algorithm never halts on nonsatisfiable ones.