Easy Predictions for the Easy-Hard-Easy Transition

Andrew J. Parkes
CIRL
1269 University of Oregon
Eugene OR 97403, USA
http://www.cirl.uoregon.edu/parkes

Abstract

We study the scaling properties of sequential and parallel versions of a local search algorithm, WalkSAT, in the easy regions of the easy-hard-easy phase transition (PT) in Random 3-SAT.

In the underconstrained region, we study scaling of the sequential version of WalkSAT. We find linear scaling at fixed clause/variable ratio. We also study the case in which a parameter inspired by "finite-size scaling" is held constant. The scaling then also appears to be a simple power law. Combining these results gives a simple prediction for the performance of WalkSAT over most of the easy region. The experimental results suggest that WalkSAT is acting as a threshold algorithm, but with threshold below the satisfiability threshold.

Performance of a parallel version of WalkSAT is studied in the over-constrained region. This is more difficult because it is an optimization rather than decision problem. We use the solution quality, the number of unsatisfied clauses, obtained by the sequential algorithm to set a target for its parallel version. We find that qualities obtained by the sequential search with O(n) steps, are achievable by the parallel version in O($log(n)$) steps. Thus, the parallelization is efficient for these "easy MAXSAT" problems.

Introduction

Many systems, both physical and combinatorial, exhibit a phase transition (PT): the probability of a large instance having some property, Q, changes rapidly from zero to one as an "order parameter" passes through some critical value. If the decision problem for the property Q is NP-hard then one might also observe what has been dubbed (somewhat inaccurately) an "easy-hard-easy" transition in the average search cost to solve an instance, i.e to determine whether Q holds (Cheeseman, Kanefsky, & Taylor 1991). Solving instances is hardest when they are taken from the phase transition region, but becomes much easier as we move away.

An extensively studied example of an easy-hard-easy phase transition is that occurring in Random 3-SAT (see, for example, (Kirkpatrick & Selman 1994; Crawford & Auton 1996; Friedgut & Kalai 1996)). If instances are characterized by the number of variables n, and clauses c, then the relevant order parameter is $\alpha = c/n$. It is believed that, in the large n limit, the probability, $P[\alpha, n]$, of an instance being satisfiable exhibits a PT. that is, for some critical value α_C then $\alpha < \alpha_C$ gives $\lim_{n\to\infty} P(\alpha, n) = 1$, and $\alpha > \alpha_C$ gives $\lim_{n\to\infty} P(\alpha, n) = 0$. Empirical estimates are that $\alpha \approx 4.26$ (Crawford & Auton 1996).

In this paper, we will avoid the computationally hard region near α_C and instead focus on properties in the easy regions. The essential question we ask is "How is easy is easy". A theoretical motivation is that current theoretical lower-bounds on α_C are obtained by analysis of polynomial algorithms. For example, analysis of a simple iterated greedy algorithm shows $\alpha_C \geq 3.003$ (Frieze & Suen 1996). A better understanding of the easy region might lead to better lower bounds, and maybe even shed light on the PT itself. Practical motivation comes from the broad correspondence between optimization and decision problems; finding optimal solutions corresponds to solving problems at the PT. However, the associated exponential scaling is often unacceptable, instead scaling must be polynomial, and so sub-optimal solutions accepted, corresponding to solving problems in the easier satisfiable region.

We are also motivated by the common use of iterative repair algorithms for large realistic problems. Hence, we study the iterative repair algorithm "WalkSAT" (WSAT) (Selman, Kautz, & Cohen 1996). Although it is incomplete, and so can never prove unsatisfiability, it is usually very effective on satisfiable instances.

The main portion of the paper presents results on the scaling of WSAT in the easy satisfiable region. Previous work (Parkes 2001) showed that scaling of the number of flips is O(n) if α is held constant. We improve upon that work by finding a simple expression for the median flips needed to solve instances. The expression gives a good prediction of performance over a large portion of the easy region.

"Easy" is usually taken to mean polytime. However, this is specific to sequential algorithms. It is also natural to ask how easy the underconstrained regions are for parallel algorithms. The question is non-trivial because it is generally believed not all polytime algorithms can be done efficiently in parallel. "Easy" in the parallel world corresponds to solving in polylog time, O($\log(n)^k$), i.e "Nicks class", **NC** (see (Papadimitriou 1994)). Problems that are **P**-complete, such as HORNSAT, are easy for sequential algorithms but still hard for parallel algorithms. Earlier work studied a parallel ver-

sion of WSAT and found it to be an efficient parallelization on underconstrained problems (Parkes 2001). Here, we consider the over-constrained region, (the other "easy"), treating the problem as MAXSAT. We find that solution qualities achievable in O(n) time by the sequential version are achievable by the parallel version in polylog time; again suggesting that, on average, this form of "easy" is also "easy parallel."

Scaling of WSAT

The version of WSAT that we use is:

for $i := 1$ to MAX-TRIES
  $P :=$ a randomly generated truth assignment
  for $j := 1$ to MAX-FLIPS
    if P is a solution, **then** return P
    else $c :=$ a randomly selected unsatisfied clause
      foreach literal $l \in c$
        $B(l)$ = number of clauses that would break if were to flip l
      m = minimum value of $B(l)$
      $L = \{ l \mid B(l) = m\}$
      if $m = 0$ **then** flip a random literal from L
      else
        with probability p flip a random literal in c
        else flip a random literal from L
return failure

At each iteration an unsatisfied clause is selected randomly and one of its literals is flipped. The heuristic for selection of the literal is to avoid breaking of other clauses. Though sometimes, controlled by a "noise parameter," p, a random choice is made.

To study scaling requires making a choice for α as we increase n, that is, a "slice" through (c, n) space. We first consider the simplest choice: a "fixed-α slice". Previous work (Parkes 2001) showed that the average number of flips increased linearly along such a slice. For completeness, we first repeat the fixed-α experiments of (Parkes 2001). Studying scaling not only requires a selection of problem instances, but also requires decisions about how algorithm parameters, such as p, should be varied. Here we make the simplifying assumption that parameters are constant. In (Parkes 2001), the noise parameter, p was set to 0.3, and although this is close to optimal in the under-constrained region, we will see later that it is significantly sub-optimal as we move closer to the PT. Here, we use $p = 0.5$ which gives better performance.

To obtain the mean, Maxflips was set large enough, 5×10^7, that problems are generally solved on the first try. Unlike (Parkes 2001), we determined not only the mean but also the median numbers of flips. Figure 1 gives the results along fixed-α slices for a representative set of α values. When we are well below the PT then median and mean values essentially coincide and scaling is clearly a simple linear. As α increases then, for smaller values of n the mean is significantly below the mean, but as n increases the two averages converge again.

Since the mean and median are so close, we decided to focus on the median because it requires much less CPU time. To find medians we set MAX-FLIPS large enough to solve over 50% of the instances on the first try. In this case, it is sufficient to set MAX-TRIES=1, keep MAX-FLIPS small and leave many instances unsolved. Also the median tends to be less susceptible to outliers. For small values of n we filter out unsatisfiable instances using a complete solver. For larger n this was impractical, but for the easy regions studied here, this turned out not to be important (very few instances are expected to be unsatisfiable).

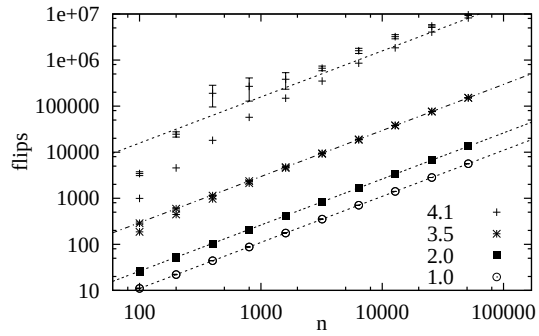

Figure 1: Number of flips needed by WSAT along "fixed-α slices." For each indicated value of α both the mean and median points are plotted; when not overlapping they can be distinguished because the mean is higher and is also given error-bars (corresponding to the usual 95% confidence interval, not the standard deviation).

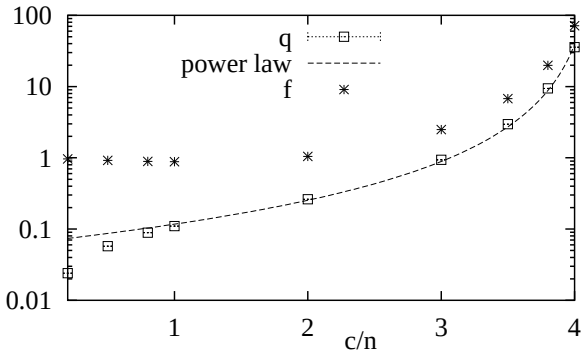

Figure 2: Empirical values for the $q(\alpha)$ of (1), and the flip inefficiency measure $f(\alpha)$ of (2). The "power law" fitted to the $q(\alpha)$ is $1.27/(4.19 - x)^{2.055}$.

The (asymptotic) linearity in Figure 1 suggests the median number of flips F is given by

$$F = q(\alpha)n \tag{1}$$

Note, F is the number of flips that will solve 50% of the (satisfiable) instances. In WSAT the initial truth assignment is random and so, for Random 3-SAT, can be expected to leave $c/8$ of the clauses unsatisfied. At fixed α this is also linear in n, and so it is useful to define

$$f(\alpha) = \frac{F}{c/8} = \frac{8q(\alpha)}{\alpha} \tag{2}$$

which can be regarded as an "inefficiency measure". Fits to the data gave values for the $q(\alpha)$ and $f(\alpha)$ as plotted in Figure 2. We will return to the "power law" later. Observe that for small α the repairs are efficient, approximately one flip is needed per initial unsatisfied clause. This is consistent with the expectation that clauses are loosely coupled and so repairs will show little interference.

The results at fixed-α slices show linear scaling but have two drawbacks. They give no insight into the nature of $q(\alpha)$. Also, the phase transition (PT) region covers an ever narrower region of α as n increases, and so there is a reasonable sense in which a fixed α moves away from the PT as n increases. This movement could make the fixed-α slice artificially easy. Both deficiencies are overcome if we also study slices that cause α to increase as n increases. A natural candidate would be a slice at a fixed value of "probability an instance is unsatisfiable." However, in the underconstrained region such probabilities are too small to measure directly; to sidestep this difficulty we use methods inspired by "finite-size rescaling".

For any finite value of n the transition is not a step function, but instead become sharper as n is increased. The behavior of the satisfiability probability, $P[\alpha, n]$, was illuminated by ideas from physics (Kirkpatrick & Selman 1994) which suggested defining a new parameter

$$u(\alpha, n) = (\alpha - \alpha_C)n^{1/\nu} \tag{3}$$

The expectation was that for well-chosen ν the probability would be close to some function $P(u)$ depending on u only. Keeping u constant as n increases drives $\alpha \to \alpha_C$ in such a way that the "probability of satisfiability" stays constant, just as desired for our slice. Experimentally, this was found to happen for $\nu = 3/2$ (Kirkpatrick & Selman 1994) . However, these early results had $\alpha_C = 4.17$ whereas more extensive studies later suggested $\alpha_C = 4.24$ or $\alpha_C = 4.26$ depending on exactly how the data was analyzed (Crawford & Auton 1996). Given this uncertainty we decided to relax the definition for u to

$$u_{\alpha_0}(\alpha, n) = (\alpha_0 - \alpha)n^{2/3} \tag{4}$$

where α_0 was another parameter, (initially) expected to come out to be the same as α_C. (Also, for convenience, we have flipped signs so that $u > 0$ corresponds to the satisfiable side of the PT).

Hence, we define a "fixed-u slice" as a sequence of problems with

$$\alpha(n) = \alpha_0 - u_{\alpha_0} n^{-2/3} \tag{5}$$

Note that $\alpha \to \alpha_0$ as n increases.

Firstly, we took slices took slices at $u = 100$ and with $\alpha_0 = 4.24$ and $\alpha_0 = 4.17$. Results are shown in Figure 3. In all cases, "flips" refers to median number of flips over a set of (satisfiable) instances. (Usually 1000 instances were used per point, except for the vary largest slowest ones. However, away from the PT, the variances are often small and so not many instances are needed to obtain a reliable estimate.) Both cases are close to straight lines on the log-log scale and so close to power laws, however the slopes change

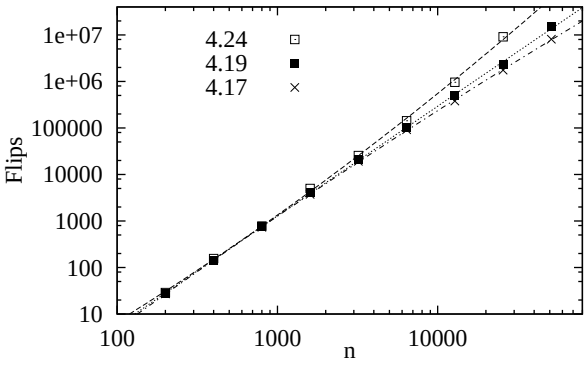

Figure 3: Scaling at $u = 100$, but with different values of α_0. The lines are best-fits of (6).

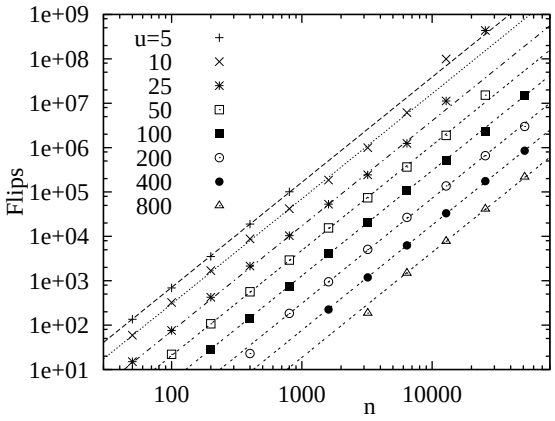

Figure 4: Scaling along lines of fixed u for various values of u with $\alpha_0 = 4.19$. Associated lines are fits to (7) with $p_b = 2.37$ and only p_a allowed to vary with u.

slowly with n. This suggested considering a modified power law:

$$v_3(n) = p_a n^{(p_b + p_c \log(n))} \tag{6}$$

where p_a, p_b, and p_c are adjustable constants, and a positive value of p_c corresponds to a power-law with a slowly increasing exponent. This gave gave reasonable fits to both the $\alpha_0 = 4.24$ and $\alpha_0 = 4.17$ lines. However, as is clear from the trends in the slopes, $p_c > 0$ for $\alpha_0 = 4.24$ but $p_c < 0$ for $\alpha_0 = 4.17$. Given these opposing trends it was natural to ask whether some intermediate value of α_0 was closer to a power law. Trial and error suggested $\alpha_0 = 4.19$, and the associated data is also shown in Figure 3. (In this section our aim is to find a simple functional approximation to the data and so the informal use "trial and error" is adequate.) Fitting to (6) gave a value for p_c that was consistent with zero, that is, within the confidence limits reported by the fit facility in gnuplot that we used for all function fitting.

Having identified $\alpha_0 = 4.19$ as giving a simple power law along the $u = 100$ slice we then moved to try other values of u. Results are shown in Figure 4. Except for deviations at the end of the range that we will discuss later, each value of u also seems to be a straight-line. More surprisingly the

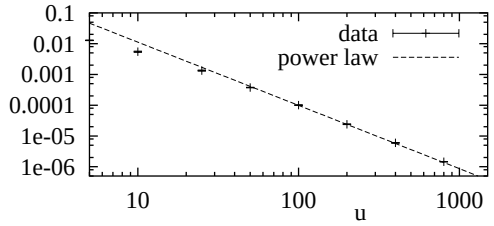

Figure 5: Results for $p_a(u)$ as obtained from Figure 4 and (7) with $p_b = 2.37$. The associated "power law" is $1.27/u^{2.055}$.

lines are parallel on the log-log plot suggesting that they all corresponds to power law scaling with the same exponent. This suggested fitting the data to a 2-parameter restriction of (6)

$$F = p_a(u)n^{p_b} \tag{7}$$

Trial and error gave $p_b = 2.37$. The coefficients $p_a(u)$ were then extracted using each slice; the results are plotted in Figure 5.

We now have two descriptions of the scaling. For constant α, F is O(n), and given by (1). For constant $u_{4.19}$, it is O(n^{p_b}), and given by (7). Consistency requires

$$q(\alpha)n \equiv p_a(u)n^{2.37} \tag{8}$$

Eliminating n using (4) gives

$$q(\alpha)(\alpha_0 - \alpha)^{2.055} = p(u)u^{2.055} \tag{9}$$

The sides of this equation then depend on independent variables and so by "separation of variables" must both be some constant, g_0, whence

$$q(\alpha) = \frac{g_0}{(\alpha_0 - \alpha)^{2.055}} \tag{10}$$

$$p(u) = \frac{g_0}{u^{2.055}} \tag{11}$$

We find that $g_0 = 1.27$ gives a good fit to both $q(\alpha)$ and $p(u)$. The corresponding "power laws" are illustrated along with the empirical coefficients in Figure 2 and Figure 5.

Hence, we find $F \approx G$ where

$$G = \frac{g_0 n}{(\alpha_0 - \alpha)^{g_1}} \tag{12}$$

with $g_0 \approx 1.27$, $\alpha_0 \approx 4.19$, and $g_1 \approx 2.055$.

Refinements of the Scaling Results

The functional form of G suggests that we can generalize the definition of u and a fixed-u slice to

$$\alpha(n) = \alpha_0 - u_{\alpha_0} n^{-\beta} \tag{13}$$

for which

$$G = \frac{g_0}{u} n^{1+\beta g_1} \tag{14}$$

That is, for any value $\beta > 0$, we obtain a slice for which $\alpha \to \alpha_0$ as $n \to \infty$, and for which G is a simple power-law.

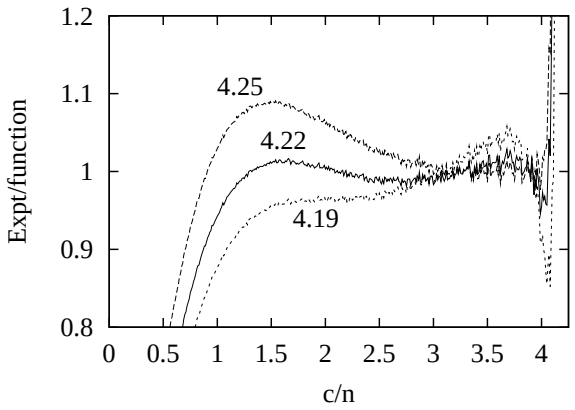

Figure 6: Ratio of experimental F to "best-fit" G for various α_0. (At $n = 51200$ with $p = 0.5$)

Conversely, it suggests imposing polynomial growth on F also forces us below α_0, that is, α_0 would be a threshold between polynomial and super polynomial (though not necessarily exponential) growth. It is important to note that the finite-size scaling value, $\beta = 2/3$, turns out not to be preferred in any way. Hence, although the results were inspired by finite-size scaling, they are ultimately independent. This also answers objections to using finite-size scaling parameters far outside the PT region.

The value $\alpha_0 \approx 4.19$ is somewhat surprising since the PT itself is believed to be at $\alpha_C \approx 4.24$ which is distinctly above this value. Since previous experiments to determine α_C were limited to $n \leq 400$ whereas we have worked up to $n = 51,200$, it is possible that the PT has shifted slightly for these values of n. However, we do not know any reason to disbelieve $\alpha_C \approx 4.24$. Also, Figure 4 show the data deviates from the fitted lines for both small and large α. Hence, in this section, we make a closer investigation of the scaling.

Experiments are performed at fixed values of n and at intervals of $\Delta\alpha = 0.01$ and with 101 instances per point, and with a single run of WSAT per instance. This small sample size leaves some noise in the data, as will be evident from the graphs, but was essential in order to keep the total cpu time practical for the large values of n used.

Firstly, Figure 6 gives the ratio of experimental F to various "best fits" for G. The differences are systematic rather than just random. Over most of the range the difference is less than 10%, and so the function G still provides an good predictor of F within the easy region. However, the systematic discrepancies do mean that fitting G to the data will give slightly different values for the parameters α_0, g_0 and g_1 depending on the range used for the fit. Furthermore, the tendencies in the deviations are such as to favor the small value of α_0 obtained in the experiments along constant-u slices.

The failure of G for small values of α is not unreasonable. For typical instances many variables will not occur at all, and the clauses are highly likely to be very decoupled. They could easily have a qualitatively different behavior. Hopefully, the small deviations of the flip inefficiency from 1.0 could be derivable analytically, and G modified accordingly.

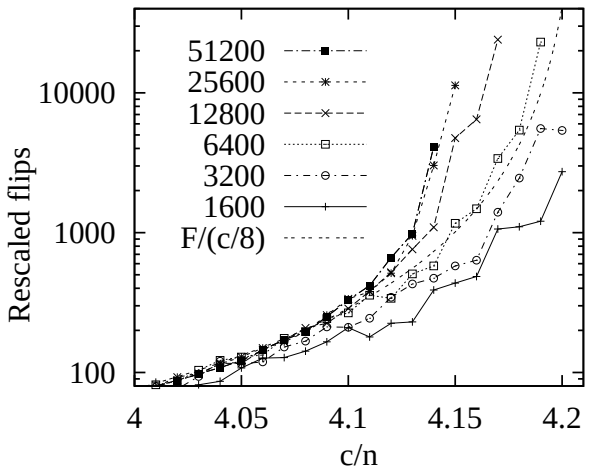

Figure 7: Rescaled flips, $f = F/(c/8)$. For noise $p = 0.5$. G is selected by fitting to $n = 25600$ over the range [3.0,4.1] giving $\alpha_0 = 4.21$, $g_1 = 2.10$, and $g_0 = 1.43$.

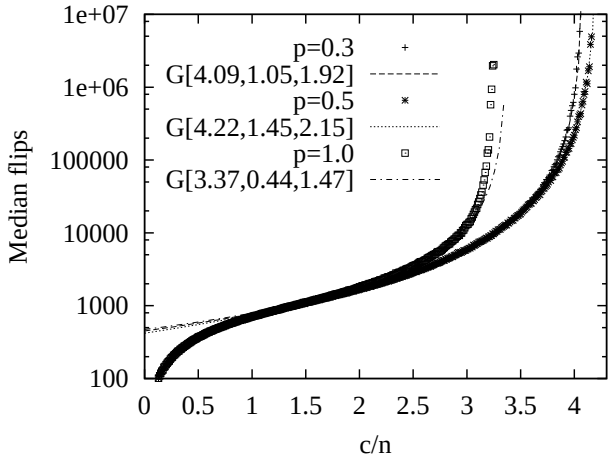

Figure 8: Median flips for indicated values of the noise parameter p. The lines are those of G, using the indicated "best-fit-values" for the parameter lists $[\alpha_0, g_0, g_1]$.

Generally, instances close to the PT are more important, and so the discrepancies at large values of α are more of a concern. Hence, consider Figure 7 where we show empirical values for, f, "flips/(c/8)," close the the PT. On reading Figure 7 it helps to consider the case that flips were a simple power law, $F = an^b$, and so $f(n) = (8a/\alpha)n^{b-1}$. With fixed α, this gives $log(f(2n)) - log(f(n)) = b - 1$. Each doubling of n leads to the same increment on the y-axis, and linear scaling corresponds to zero increment between lines.

For $\alpha < 4.1$ and large n the points on Figure 7 indicate that the scaling approaches linear scaling, though after initially being super-linear for $n < 3200$. However, at $\alpha = 4.15$ the increment on doubling n seems to be increasing each time, suggesting that growth is faster than a simple power-law. Also, observe that the formula for G, even with $\alpha_0 = 4.21$, gives a line that severely underestimates F for large n once we are in the region $\alpha > 4.15$, and again the underestimation seems to increase with n.

Hence, the data indicates a transition at $\alpha \approx 4.15$ from power-law to super-polynomial scaling. Note that the transition is not associated with unsatisfiable instances (for these values of α and n we are well outside the PT).

Assuming the threshold is real, then it is natural to ask whether it is sensitive to the exact algorithm used, or is possibly tied to some (unknown) change in properties of the instances. Hence, in Figure 8 we give medians flips for $p = 0.3$ and 1.0 as well as the $p = 0.5$ used until this point. Again, G gives good fits to the data over the middle range of α values, but we see that the obtained values for the parameters do change significantly. This suggests the thresholds are as driven more by failings of the algorithm, rather directly by semantic properties of instances.

Interestingly, performance at smaller values of α is insensitive to p, presumably reflecting the lesser need for escape from local minima on such easy instances.

Lastly we look at Figure 9 which is the same type of plot as Figure 7 but for $p = 1.0$ instead of $p = 0.5$. In the region

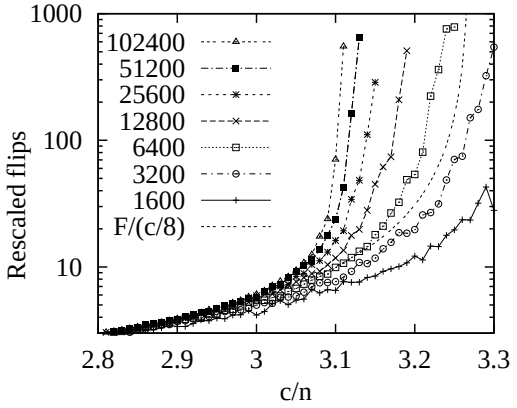

Figure 9: Rescaled flips, $f = F/(c/8)$. For noise $p = 1.0$. G is selected by fitting to $n = 25600$ over the range [2.0,3.0] giving $\alpha_0 = 3.27$, $g_1 = 1.30$, and $g_0 = 0.39$.

$n \approx 10^5$, then for $\alpha_T < 3.0$ scaling approaches linear for the largest n values used. Whereas for $\alpha > 3.1$ it seems to be growing faster than a power law. This time there appears to emerge a transition at $\alpha_T \approx 3.1$ from linear to super-linear (and apparently super-polynomial). At $p = 1.0$, WSAT is particularly simple: a "no-breaks" move is taken if possible, otherwise a literal is flipped at random. Hopefully, the extra simplicity will eventually allow an analytic derivation the non-trivial asymptotic behavior observed.

We should note that in all these apparent thresholds it is hard to eliminate the possibility that the threshold moves slowly as n increases. For example, a slow decrease in the threshold, would correspond to scaling, at fixed-α, being linear until some large value of n and then changing to super-linear or super-polynomial at very large n. We see no evidence for such behavior, but cannot eliminate it.

We also remark that the thresholds only become apparent for large n. Although a few hundred variables are sufficient

to show a clear satisfiability threshold, it seems that the algorithmic thresholds here needs tens of thousands in order to become clear. The ability of WSAT to escape local minima effectively might only diminish when faced with the (presumably) much deeper local minima occurring in large theories. Also the results often only show linear growth once reaching large n. In practice, this means that scaling results obtained at small n (meaning $n < 1000$) might be misleading. Unfortunately, experiments with such large values of n are currently impractical except in the easy region.

Previous studies of the scaling of WSAT have generally focussed on the PT region. The behavior of WSAT along the phase transition line was studied in (Singer, Gent, & Smaill 2000)).(Gent *et al.* 1997) study scaling across the PT region; Although they only study much smaller values on n it might well be valuable to relate their results at the edge of the PT region to ours at the edge of the easy region. Also, see (Coarfa *et al.* 2000) for recent studies of scaling of systematic algorithms in the underconstrained region, and a useful overview of other scaling results.

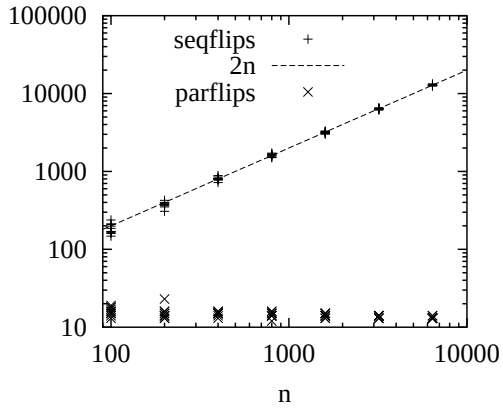

Figure 10: Instances are taken from Random 3-SAT with $\alpha = 6$, and an associated target quality is fixed to that achieved by WSAT(SEQ) in $2n$ flips. For each instance, we then plot both the average sequential flips and parallel flips needed to achieve the target quality.

Parallel WSAT and Empirical Scaling Results

A direct parallelization of WSAT is simply to repair all the unsatisfied clauses in parallel, rather than a randomly selected one (Parkes 2001). For WSAT(PAR) the central loop of WSAT(SEQ) is replaced with a "parallel flip" or "parflip":

parallel: foreach clause c
  if c is unsatisfied
    foreach literal $l \in c$
      $B(l)$ = number of clauses that would break if were to flip l
    m = minimum value of $B(l)$
    $L = \{\ l \mid B(l) = m\}$
    if $m = 0$ **then** flip a random literal from L
    else
      with probability p flip a random literal in c
      else flip a random literal from L

Note that O(n) flips are done simultaneously, and so it is reasonable to hope for a linear speedup. If the sequential version takes O(n) then it is also reasonable to hope that the parallel version would run in (poly-)logarithmic number of flips. Observing that constant-α slices are O(n) for sequential WSAT, Previous studies (Parkes 2001) of the above parallel version found that the hoped for speedup did indeed occur. Empirically, constant-α slices were O($\log(n)^2$).

One might study the behavior of WSAT(PAR) along constant-u slices however this would probably not be interesting. The sequential is polynomial, O($n^{2.37}$) rather than linear, and so (presumably) the most one can hope for is to reduce this to O($n^{1.37}$).

Instead, we are motivated from the fact that many problems are over-constrained and are treated as optimization rather than decision problems. We take this to correspond to taking $\alpha > \alpha_C$ and treating the problems as MAXSAT rather than SAT. In this case one goal might be to satisfy as many clauses as possible; however, this is NP-hard and can be expected to take exponential time. In practice, suboptimal solutions are sought, and local repair methods such as WSAT are still effective.

Hence, we ask whether the WSAT(PAR) gives an effective parallelization on over-constrained optimization problems. Unfortunately, doing optimization rather than simple satisfiability means that there is no natural meaning of "success" for a search attempt. To overcome this we took a simple method to evaluate WSAT(PAR) with respect to WSAT(SEQ).

1. select a value of Maxflips m(n)
2. for each instance we determine a target quality T by
   (a) for each instance we run WSAT(SEQ) for m(n) flips and determine the smallest number of unsatisfied clauses that occurred
   (b) this is averaged over multiple runs of WSAT(SEQ)
3. this target quality is then used as "success criterion." That is, WSAT is considered to succeed as soon as the number of unsatisfied clauses is reduced to T or better.
4. both WSAT(SEQ) and WSAT(PAR) are then evaluated by this success criterion.

Not surprisingly, WSAT(SEQ) then takes O($m(n)$) flips. If $m(n)$ is selected to be linear then it is reasonable to hope that WSAT(PAR) will reduce this to polylog flips. This appears to be confirmed by experiments, of which we report just a typical representative in Figure 10: Qualities achievable by WSAT(SEQ) in $2n$ flips, are, on average, achievable in O($\log(n)$) flips by WSAT(PAR).

More challenging for WSAT(PAR) is to give O($n \log(n)$) flips to WSAT(SEQ). A typical result is given in Figure 11. In this case WSAT(PAR) fails to achieve a linear speedup. However, we noted that WSAT(PAR) can make sets of flips that are likely to be too drastic. In particular, it can cause clauses that satisfied by two or more literals to become unsatisfied. Hence, we implemented a second

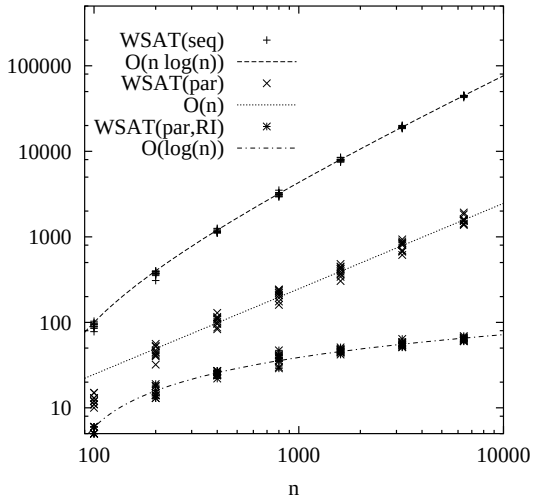

Figure 11: Points are determined as for Figure 10 except that the target quality is determined using $n(1 + \log_2(n/100))$ flips of WSAT(SEQ). We also plot points giving the performance of the modified parallel WSAT.

version WSAT(par,NI), that only allows one of the literals to be flipped in such cases. In this case, Figure 11 a linear speedup, from O($n \log(n)$) to O($\log(n)$) was again observed.

We take this as evidence that the "easy MAXSAT" problems we have defined using the over-constrained region of Random 3-SAT are also easy for parallel algorithms.

Closing Comments and Future Work

We have found a remarkably simple expression, (12), that is a good predictor of the median number of flips used by WSAT over a large portion of the easy region below the phase transition in Random 3-SAT.

The expression (12) along with other results presented, suggest that WSAT has a threshold value of α beyond which the median flips scaling is super-polynomial, but below which scaling is linear. The position of the threshold depends on the noise parameter. It would be interesting to see whether other versions of local repair show similar behavior.

The greatest deficiency of this work is that it is purely empirical, and so cannot prove anything about true asymptotic behavior. Hopefully, some of the results can eventually be supported by analytic studies, if only for small values of α, or for simpler versions of WSAT such as with the noise parameter $p = 1$.

Acknowledgments

This work was sponsored in part by grants from Defense Advanced Research Projects Agency (DARPA), number F30602-98-2-0181, and DARPA and Air Force Research Laboratory, Rome, NY, under agreement numbered F30602-00-2-0534. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright annotation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of DARPA, Rome Laboratory, or the U.S. Government.

References

Cheeseman, P.; Kanefsky, B.; and Taylor, W. M. 1991. Where the really hard problems are. In *Proceedings of the Ninth National Conference on Artificial Intelligence (AAAI–91)*, 331–337.

Coarfa, C.; Demopoulos, D. D.; Aguirre, A. S. M.; Subramanian, D.; and Vardi, M. Y. 2000. Random 3-SAT: The plot thickens. In *Principles and Practice of Constraint Programming*, 143–159.

Crawford, J. M., and Auton, L. D. 1996. Experimental results on the crossover point in random 3-SAT. *Artificial Intelligence* 81:31–57.

Friedgut, E., and Kalai, G. 1996. Every monotone graph property has a sharp threshold. *Proc. Amer. Mathl. Soc.* 124:2993–3002.

Frieze, A., and Suen, S. 1996. Analysis of two simple heuristics on a random instance of k-SAT. *Journal of Algorithms* 20:312–355.

Gent, I.; MacIntyre, E.; Prosser, P.; and Walsh, T. 1997. The scaling of search cost. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence (AAAI–97)*, 315–320.

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

Papadimitriou, C. 1994. *Computational Complexity*. Addison-Wesley.

Parkes, A. J. 2001. Distributed local search, phase transitions, and polylog time. In *Proceedings of the workshop on "Stochastic Search Algorithms", held at "Seventeenth International Joint Conference on Artificial Intelligence (IJCAI-01)"*.

Selman, B.; Kautz, H.; and Cohen, B. 1996. Local search strategies for satisfiability testing. In Johnson, D. S., and Trick, M. A., eds., *Cliques, Coloring and Satisfiability*, 521–531. American Mathematical Society. Proceedings of the second DIMACS Implementation Challenge, October 1993.

Singer, J.; Gent, I. P.; and Smaill, A. 2000. Local search on random 2+p-sat. In *Proceedings of ECAI-2000*, 113–117.