On the Empirical Time Complexity of Random 3-SAT at the Phase Transition

Zongxu Mu and **Holger H. Hoos**
Department of Computer Science
University of British Columbia
{zongxumu, hoos}@cs.ubc.ca

Abstract

The time complexity of problems and algorithms, *i.e.*, the scaling of the time required for solving a problem instance as a function of instance size, is of key interest in theoretical computer science and practical applications. In this context, propositional satisfiability (SAT) is one of the most intensely studied problems, and it is generally believed that solving SAT requires exponential time in the worst case. For more than two decades, random 3-SAT at the solubility phase transition has played a pivotal role in the theoretical and empirical investigation of SAT solving, and to this day, it is arguably the most prominent model for difficult SAT instances. Here, we study the empirical scaling of the running time of several prominent, high-performance SAT solvers on random 3-SAT instances from the phase transition region. After introducing a refined model for the location of the phase transition point, we show that the median running time of three incomplete, SLS-based solvers – WalkSAT/SKC, BalancedZ and probSAT – scales polynomially with instance size. An analogous analysis of three complete, DPLL-based solvers – kcnfs, march_hi and march_br – clearly indicates exponential scaling of median running time. Moreover, exponential scaling is witnessed for these DPLL-based solvers when solving only satisfiable and only unsatisfiable instances, and the respective scaling models for each solver differ mostly by a constant factor.

1 Introduction

The propositional satisfiability problem (SAT) was the first problem proven to be $\mathcal{NP}$-complete [Cook, 1971], and no known algorithm solves the problem efficiently in the sense of better-than-exponential scaling of running time with instance size in the worst case. SAT also has a broad range of practical applications, and despite its discouraging worst-case time complexity, many SAT solvers have been constructed that perform well on theoretically and practically interesting sets of benchmark instances. The performance of SAT solvers are regularly assessed in SAT competitions [SatCompetition.org, 2014] , which feature benchmark instances from various applications, instances that are hand-crafted to challenge solvers, and randomly generated instances.

Following seminal work by [Cheeseman *et al.*, 1991] and [Mitchell *et al.*, 1992], randomly generated 3-SAT instances at the so-called solubility phase transition, where 50% of the instances generated at a given size are satisfiable, have been widely studied; to this day, they are regarded as a model for the difficulty of the $\mathcal{NP}$-complete 3-SAT problem and represent one of the most prominent benchmarks for SAT solvers. It is conjectured, yet unproven, that the location of the solubility phase transition for uniform random 3-SAT, in terms of the ratio of the number of clauses and variables, m/n, converges towards a limiting value as n approaches infinity. [Crawford and Auton, 1996] studied the location of the phase transition for random 3-SAT as a function of n and provided a widely used formula, based on empirical data for $n = 20 \ldots 300$.

Here, we consider the question how the performance of prominent, high-performance SAT solvers on uniform random 3-SAT at the solubility phase transition scales with n. Prompted in part by earlier, inconclusive results suggesting polynomial scaling of incomplete SAT solvers based on stochastic local search (SLS) [Gent and Walsh, 1993; Parkes and Walser, 1996; Gent *et al.*, 1997], we characterise the empirical scaling of the performance of prominent incomplete, SLS-based and complete, DPLL-based SAT solvers. We are interested in determining whether there are qualitative differences in the scaling behaviour of SLS-based and DPLL-based solvers, in the scaling of DPLL-based solvers on satisfiable and unsatisfiable instances, and in the scaling of different solvers within the two major groups (SLS-based and DPLL-based). Our investigation leverages and extends a recent approach for empirical scaling analysis by [Hoos and Stützle, 2014] that uses resampling techniques to statistically assess the degree to which observed algorithm behaviour is consistent with given scaling models (see Sec. 3).

To avoid 'falling off' the phase transition, which could bias our scaling results towards easier instances, we re-examine the model in [Crawford and Auton, 1996] for the location of the phase transition point, and derive a new model that agrees better with observed data for $n > 300$ and with recent results from statistical physics (see Sec. 4). Using sets of random 3-SAT instances generated according to this model, our main findings are as follows (see Sec. 5):

- The median running times of the three prominent SLS-

based solvers we studied, WalkSAT/SKC [Selman *et al.*, 1994], BalancedZ [Li *et al.*, 2014] and probSAT [Balint and Schöning, 2014; 2012], scale polynomially (with an exponent of about 3), and exponential scaling models are rejected with 95% confidence. Furthermore, we found no evidence that higher percentiles of the distributions of running times over sets of instances for fixed n may scale exponentially.

- The median running times of the three DPLL-based solvers we considered, kcnfs [Dequen and Dubois, 2004], march_hi [Heule and van Maaren, 2009] and march_br [Heule, 2013], exhibit exponential scaling (with a base of about 1.03), and polynomial models are rejected with 95% confidence.
- For all three DPLL-based solvers, the median running times when solving only satisfiable and only unsatisfiable instances, respectively, clearly exhibit exponential running time scaling, and the respective scaling models differ mainly by a constant factor.
- While the scaling models for the SLS-based solvers are very similar to each other, the two march-variants scale significantly better than kcnfs.

As we will discuss in the final part of this paper (Sec. 6), these results shed new light on random 3-SAT at the phase transition and on the fundamental differences in the behaviour of SLS- and DPLL-based solvers; they also invite thorough empirical scaling studies for other problems and solvers.

2 Background and Related Work

SAT is one of the most intensely studied problems in the computer science literature and beyond, and 3-SAT is arguably the most prominent $\mathcal{NP}$-complete decision problem [Cook, 1971]. Interest in phase transition phenomena in combinatorial problems and in uniform random 3-SAT specifically rose sharply when [Cheeseman *et al.*, 1991] demonstrated that the hardest instances are found around a critical value of an order parameter, where a transition from predominantly soluble to mostly insoluble instances occurs. Uniform random k-SAT instances are generated by constructing uniformly and independently at random m clauses, each of which is obtained by sampling, again uniformly and independently at random, 3 of n variables, and negating each of them with probability 1/2 [Mitchell *et al.*, 1992] (duplicate clauses are eliminated); the order parameter is the clauses/variables ratio, m/n. It is believed, but not yet proven, that for $k \geq 3$, the location of the phase transition point of uniform random k-SAT converges to a fixed threshold value as n approaches infinity (it is known, however, that the phase transition is provably sharp for $n \to \infty$ [Friedgut and Bourgain, 1999]). Assuming threshold values exist for small k, an accurate theoretical upper bound was proven by [Franco and Paull, 1983] and was later improved to $2^k \log 2 - (1 + \log 2)/2 + o_k(1)$ by [Kirousis *et al.*, 1998]. [Achlioptas and Peres, 2004] achieved a major breakthrough in proving a lower bound, which was recently improved to $2^k \log 2 - (1 + \log 2)/2 - o_k(1)$ [Coja-Oghlan, 2014].

In a prominent study on the empirical difficulty of SAT instances, [Mitchell *et al.*, 1992] demonstrated that instances drawn from the phase transition region of uniform random 3-SAT instances tend to be the most difficult for a simple DPLL solver. Similar results were shown by [Yokoo, 1997] for an SLS-based solver on the satisfiable phase of uniform random 3-SAT. [Crawford and Auton, 1996] studied the phase transition region of uniform random 3-SAT empirically, developed a model for the location of the phase transition point and presented additional evidence for the difficulty of the SAT instances found there.

[Gent and Walsh, 1993] studied the empirical behaviour of GSAT, one of the earliest and most prominent SLS-based SAT solvers [Selman *et al.*, 1992], and its two variants, DSAT and HSAT. They noted that the scaling of the average number of variable flips required by these solvers for solving phase transition random 3-SAT instances was consistent with less-than-linear exponential scaling with n and did not rule out a polynomial scaling model with degree ≈ 3. Later, [Parkes and Walser, 1996] presented empirical evidence that the scaling of the average number of flips required by a more recent, prominent SLS-based SAT solver, WalkSAT/SKC [Selman *et al.*, 1994], on the same class of instances might scale either as a power function with a slowly growing exponent, or as a sub-exponential function. Furthermore, [Gent *et al.*, 1997] found that the 90th percentile of the number of flips of GSAT for random 3-SAT appears to grow no faster than n^4. However, in all cases, performance was measured in variable flips (which become more expensive as n increases) and the results are based on limited curve fitting only, with a vaguely defined notion of 'good fit'. [Coarfa *et al.*, 2003] used simple curve fitting to study the empirical scaling of median running time for three complete solvers on random 3-SAT and observed exponential scaling above certain solver-dependent density thresholds. In contrast, in this work, we use a more advanced and statistically sound approach for assessing scaling models. Unlike these earlier studies, we also challenge our scaling models by assessing their predictions on larger instances sizes than those used for fitting the models.

A significant advance in the methodology for studying the empirical scaling of algorithm performance with input size was achieved by [Hoos, 2009]. Their method uses standard numerical optimisation approaches to automatically fit scaling models, which are then challenged by extrapolation to larger input sizes. Most importantly, it uses a resampling approach [Efron and Tibshirani, 1993] to assess the models and their predictions in a statistically meaningful way. [Hoos and Stützle, 2014] used this approach to characterise the scaling behaviour of Concorde, a state-of-the-art complete algorithm for the travelling salesperson problem (TSP), demonstrating that the scaling of Concorde's performance on 2D Euclidean TSP instances with n cities agrees well with a model of the form $a \cdot b^{\sqrt{n}}$. In this work, we apply the same empirical methodology to SAT, and extends it in two useful ways.

3 Experimental Setup and Methodology

For our study, we selected three SLS-based solvers, WalkSAT/SKC [Selman *et al.*, 1994] (UBCSAT version 1.2

[Tompkins and Hoos, 2005]), BalancedZ [Li *et al.*, 2014] (V2014.05.07) and probSAT [Balint and Schöning, 2014] (version sc14), and three DPLL-based solvers, kcnfs [Dequen and Dubois, 2004] (V2004), march_hi [Heule and van Maaren, 2009] and march_br [Heule, 2013] (version sat+unsat). We chose these, because WalkSAT/SKC and kcnfs are two classical solvers that are widely known in the community, and the others showed excellent performance in recent SAT competitions.

All sets of uniform random 3-SAT instances used in our study were obtained using the generator which produced SAT competition instances [Balint *et al.*, 2012]. To separate satisfiable from unsatisfiable instances with $n \leq 500$, we used CSHCrandMC [Malitsky *et al.*, 2013], the winner of Random SAT+UNSAT Track in the 2013 SAT Competition. For larger instances, for which the use of complete solvers is impractical, we performed long runs (43 200 CPU sec) of BalancedZ and treated those instances not solved in those runs as unsatisfiable. Since BalancedZ never failed to solve any of our instances known to be satisfiable, and because the cut-off time we used for these long runs was at least 20 times of the maximum running time observed on any satisfiable instance of the same size, we have high confidence that we did not miss any satisfiable instances. We note that, even if we had misclassified some satisfiable instances, the quantile performance measures would not be much affected (if at all).

For collecting running time data for our solvers, we used the Compute Canada / Westgrid cluster orcinus (DDR), each equipped with two Intel Xeon E5450 quad-core CPUs at 3.0 GHz with 16GB of RAM running 64-bit Red Hat Enterprise Linux Server 5.3. To avoid measurement noise caused by memory bottlenecks, we used only one core per CPU to run our solvers and imposed a memory limit of 1GB per run.

After observing floor effects for WalkSAT/SKC due to small CPU times required for solving small instances, we calculated running times based on the number of variable flips performed, and on estimates of CPU time per variable flip for each problem size obtained from long runs on unsatisfiable instances. The CPU time estimates thus obtained closely agreed with measured CPU times for short and long runs, but are considerably more accurate. For all other solvers, which were less affected by such floor effects, we used runsolver [Roussel, 2011] to record CPU times (and to enforce CPU time limits).

For our scaling analysis, we considered two parametric models:

- $Exp\,[a, b]\,(n) := a \cdot b^n$ (2-parameter exponential);
- $Poly\,[a, b]\,(n) := a \cdot n^b$ (2-parameter polynomial).

Our approach could be easily extended to other scaling models, but, as we will show in the following, these two models jointly characterise the scaling observed in all our experiments, and thus we saw no need to consider different or more complex models. For fitting parametric scaling models to observed data, we used the non-linear least-squares Levenberg-Marquardt algorithm.

Models were fitted to performance observations in the form of quantiles, chiefly the median of the distributions of running times over sets of instances for given n. Compared to the mean, the median has two advantages: it is statistically more stable and immune to the presence of a certain amount of timed-out runs. Considering the stochastic nature of the SLS-based solvers, we performed 5 independent runs per instance and used the median over those 5 running times as the running time for the respective instance. To assess the fit of a given scaling model to observed data, we used root-mean-square error (RMSE).

Closely following [Hoos, 2009; Hoos and Stützle, 2014], we computed 95% bootstrap confidence intervals for the performance predictions obtained from our scaling models, based on 1000 bootstrap samples per instance set and 1000 automatically fitted variants of each scaling model. We extended their approach in two ways. Firstly, noting that observed running time statistics are also based on measurements on sets of instances, we calculate bootstrap percentile intervals for those, in addition to the point estimates used in the original approach. This way, we capture dependency of these statistics on the underlying sample of instances. Secondly, we determine to which extent a solver $\mathcal{A}_1$ shows scaling behaviour different from another solver $\mathcal{A}_2$, by comparing the observed running time statistics of $\mathcal{A}_1$ to the bootstrap confidence intervals obtained from the scaling model of $\mathcal{A}_2$. If the latter do not contain the former, we can reject the hypothesis that the performance of $\mathcal{A}_1$ is consistent with the scaling model of $\mathcal{A}_2$. Both extensions can obviously be used in the scaling analysis of solvers for problems other than SAT.

4 Location of Phase Transition

[Crawford and Auton, 1996] modelled the location of the phase transition point based on extensive experiments on uniform random 3-SAT instances with up to 300 variables as:

$$m_c = 4.258 \cdot n + 58.26 \cdot n^{-2/3}, \tag{1}$$

where n is the number of variables and m_c the critical number of clauses, at which about 50% satisfiable instances are obtained. The parametric form of this model was derived by [Kirkpatrick and Selman, 1994], using finite-size scaling. While this model does provide a good fit for $n \leq 300$, in preliminary experiments, we found that for $n > 300$, it increasingly underestimates m_c. Furthermore, the model in Eq. 1 predicts that, as n grows, m_c/n approaches 4.258, which is in contradiction with a more recent result by [Mertens *et al.*, 2006], who used the heuristic one-step replica symmetry breaking cavity method from statistical physics to estimate the value of m_c/n as 4.26675 ± 0.00015.

Because in the empirical scaling study that follows, we wanted to be sure to not drift off the phase transition point (which could bias the scaling models, especially, as the phase transition, in terms of m_c/n, is known to become increasingly steep as n grows), we decided to revisit and improve the Crawford & Auton's model.

We first chose several m/n ratios for each $n \in \{300, 400, \ldots, 1400\}$, around the predictions from Eq. 1, loosely adjusted based on results from preliminary experiments. Next, we generated sets of 600 uniform random 3-SAT instances for each m/n ratio. We separated out the satisfiable instances using a hybrid complete solver, CSHCrandMC [Malitsky *et al.*, 2013], with a cutoff of 3600 CPU

n	Lower bound	Upper bound	Prediction	
			Eq. 1	Eq. 3
300	4.2600 (3600)	4.2667 (3600)	4.2623	4.2638
400	4.2575 (4800)	4.2650 (600)	4.2607	4.2626
500	4.2580 (2400)	4.2660 (600)	4.2598	4.2622
600	4.2567 (600)	4.2667 (4800)	4.2594	4.2622
700	4.2600 (1200)	4.2657 (4800)	4.2591*	4.2623
800	4.2575 (2400)	4.2638 (2400)	4.2588	4.2624
900	4.2600 (2400)	4.2667 (3600)	4.2587*	4.2626
1000	4.2600 (600)	4.2670 (2400)	4.2586*	4.2627
1100	4.2609 (1200)	4.2655 (600)	4.2585*	4.2629
1200	4.2600 (2400)	4.2650 (3600)	4.2584*	4.2630
1300	4.2608 (600)	4.2646 (600)	4.2584*	4.2631
1400	4.2600 (1200)	4.2664 (2400)	4.2583*	4.2633

Table 1: Lower and upper bounds on the location of the phase transition (m_c/n) determined with 95% confidence, and predictions from the two models discussed in the text. In parentheses: number of instances used for determining the bound. Model predictions inconsistent with our lower bounds are marked with asterisks (*).

seconds per run. Up to $n = 500$, we solved all instances within that cutoff. Beyond, it would have been impractical to run any complete solver sufficiently long to prove unsatisfiability, and we therefore heuristically assumed that the instances not solved by CSHCrandMC are unsatisfiable. As mentioned earlier, we later used much longer runs of BalancedZ to challenge these putative unsatisfiable instances further, and among the thousands of instances for which this was done, only one was found to be satisfiable. Nevertheless, in what follows, we are well aware of the fact that we may underestimate the fraction of satisfiable instances in our sets.

We note that, even at large numbers of instances sampled at the correct (unknown) value of m_c/n, we should expect to get sets with a fraction of satisfiable instances varying according to a binomial distribution. Based on this observation, we determined 95% confidence intervals for the fraction of satisfiable instances observed. We then rejected the m/n values for which we empirically observed fractions of satisfiable instances outside of the respective 95% confidence interval as valid estimates of m_c/n. In this way, we obtained bounds on the location of the phase transition point, m_c/n. In many cases, the confidence intervals for our initial sets of 600 instances were too wide to yield bounds, and we subsequently increased the number of instances in those sets until for every n, we obtained an upper and lower bound on m_c/n. Those bounds and the respective set sizes are shown in Table 1.

These results provide further evidence for our earlier observation that for larger n, Crawford & Auton's model becomes inaccurate, as the respective predictions are below the lower bounds from our analysis. We note that our lower bounds are valid, as they could only increase if some of our putatively unsatisfiable instances were in fact satisfiable. For the same reason, the upper bounds may be inaccurate.

Interestingly, and notably different from Crawford & Auton's model, our results suggest that m_c/n as a function of n is not monotonic, but first drops, before slowly increasing again, possibly towards a threshold value. This observation, in combination with the limiting value found by [Mertens *et al.*, 2006] led us to choose a model of the form

$$m_c = 4.26675 \cdot n + a \cdot n^b + c \cdot n^d, \qquad (2)$$

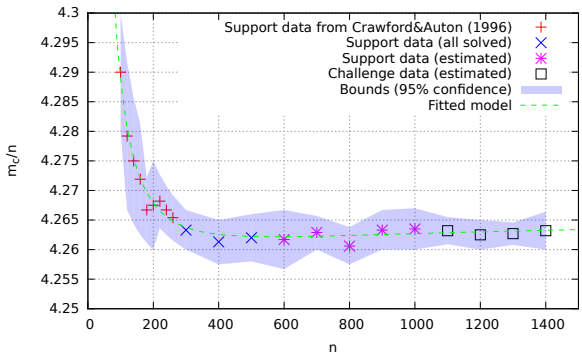

Figure 1: Empirical bounds on the location of the phase transition points, m_c/n for different n, data used for fitting our model and model predictions. Data for $n > 500$ is based on heuristic estimates of the fraction of satisfiable instances and may underestimate the true m_c/n.

where $a \cdot c < 0$, $b < 0$ and $d < 0$.[1]

Finally, taking the data points from [Crawford and Auton, 1996] up to $n = 300$ (which we believe to be of high quality) and the mid-points between our bounds from Table 1, we fitted this 4-parameter model, resulting in:

$$m_c = 4.26675 \cdot n + 447.884 \cdot n^{-0.0350967} - 430.232 \cdot n^{-0.0276188} \qquad (3)$$

Figure 1 shows the predictions obtained from this model, along with the bounds and mid-points between them from Table 1. This model was subsequently used to generate the instance sets used in the scaling analysis described in the following. While it is possible that our model is biased by the fact that we may have missed satisfiable instances for larger n, it provides a better basis than previously available for generating instances at or very near the solubility phase transition of uniform random 3-SAT.

5 Empirical Scaling of Solver Performance

We first fitted our two parametric scaling models to the median running times of the six solvers we considered, as described in Section 3. For each SLS-based solver, both models were fitted using median running times for $n = 200, 250, \ldots 500$ (support) and later challenged with median running times for $n = 600, 700, \ldots 1000$. For each DPLL-based solver, we used support data for $n = 200, 250, \ldots 400$ and challenge data for $n = 450, \ldots 550$, as for even larger n we could no longer complete sufficiently many runs to estimate even median running times. This resulted in the models, shown along with RMSEs on support and challenge data, shown in Table 2.

We note that the RMSEs on support data, *i.e.*, the data used for fitting the models, often, but not always provide a good indication of their predictive power. Based on the latter, in the form of RMSEs on challenge data, we see clear indications that the median running times of all three SLS-based

[1] This type of model is consistent with current theoretical thinking that, for small n, second-order terms may cause non-monotonic behaviour of the function $m_c(n)$. (Dimitris Achlioptas, personal communication.)

			Model	RMSE (support)	RMSE (challenge)
WalkSAT/SKC	Sat.	Exp. Model	$6.89157 \times 10^{-4} \times 1.00798^{n}$	0.0008564	0.7598
		Poly. Model	$\mathbf{8.83962 \times 10^{-11} \times n^{3.18915}}$	**0.0007433**	**0.03225**
BalancedZ	Sat.	Exp. Model	$1.32730 \times 10^{-3} \times 1.00759^{n}$	0.001759	1.081
		Poly. Model	$\mathbf{5.14258 \times 10^{-10} \times n^{2.97890}}$	**0.002870**	**0.05039**
probSAT	Sat.	Exp. Model	$8.35877 \times 10^{-4} \times 1.00763^{n}$	0.0013867	0.6487
		Poly. Model	$\mathbf{2.92275 \times 10^{-10} \times n^{2.99877}}$	**0.002285**	**0.03301**
kcnfs	All	**Exp. Model**	$\mathbf{4.30400 \times 10^{-5} \times 1.03411^{n}}$	**0.05408**	**143.3**
		Poly. Model	$9.40745 \times 10^{-31} \times n^{12.1005}$	0.06822	1516
	Sat.	**Exp. Model**	$\mathbf{2.41708 \times 10^{-5} \times 1.03205^{n}}$	**0.02496**	**83.86**
		Poly. Model	$2.41048 \times 10^{-30} \times n^{11.7142}$	0.05600	225.8
	Unsat.	**Exp. Model**	$\mathbf{6.38367 \times 10^{-5} \times 1.03386^{n}}$	**0.001466**	**52.19**
		Poly. Model	$9.70804 \times 10^{-31} \times n^{12.1448}$	0.1813	2291
march_hi	All	**Exp. Model**	$\mathbf{4.93309 \times 10^{-5} \times 1.03295^{n}}$	**0.05449**	**460.0**
		Poly. Model	$1.05593 \times 10^{-30} \times n^{12.0296}$	0.05971	1266
	Sat.	**Exp. Model**	$\mathbf{8.33113 \times 10^{-6} \times 1.03119^{n}}$	**0.03035**	**3.087**
		Poly. Model	$2.44435 \times 10^{-30} \times n^{11.4789}$	0.03879	61.77
	Unsat.	**Exp. Model**	$\mathbf{7.86081 \times 10^{-5} \times 1.03281^{n}}$	**0.03336**	**399.7**
		Poly. Model	$2.10794 \times 10^{-30} \times n^{11.9828}$	0.16703	1912
march_br	All	**Exp. Model**	$\mathbf{6.17600 \times 10^{-5} \times 1.03220^{n}}$	**0.05401**	**402.4**
		Poly. Model	$5.56146 \times 10^{-30} \times n^{11.7408}$	0.05199	1253
	Sat.	**Exp. Model**	$\mathbf{1.02788 \times 10^{-5} \times 1.03048^{n}}$	**0.02497**	**13.72**
		Poly. Model	$1.25502 \times 10^{-29} \times n^{11.1944}$	0.03341	67.85
	Unsat.	**Exp. Model**	$\mathbf{6.10959 \times 10^{-5} \times 1.03344^{n}}$	**0.03230**	**262.8**
		Poly. Model	$5.18600 \times 10^{-31} \times n^{12.2154}$	0.1586	1920

Table 2: Fitted models of median running times and RMSE values (in CPU sec). Model parameters are shown with 6 significant digits, and RMSEs with 4 significant digits; the models yielding more accurate predictions (as per RMSEs on challenge data) are shown in boldface.

solvers are overall more consistent with our polynomial scaling model, while those of the DPLL-based solvers are more consistent with our exponential scaling model – even when only considering performance on satisfiable instances. We also studied Lingeling [Biere, 2014] (version ayv) and found that its median running time for $n = 400$ is more than 1000 times larger than that of kcnfs, and that this performance gap increases with n.

But how much confidence should we have in these models? Are the RMSEs small enough that we should accept them? [2] To answer this question, we assessed the fitted models using the bootstrap approach outlined in Section 3. The results of this analysis, shown in Table 3, clearly show that observed median running times for the SLS-based solvers are consistent with our polynomial scaling model and inconsistent with the exponential model (as illustrated in Figure 2 for WalkSAT/SKC), while the opposite holds for the DPLL-based solvers. This is especially striking for the larger challenge instances sizes. Limited experiments for even larger instances sizes (up to $n = 2000$) produced further evidence consistent with the polynomial scaling models for all three SLS-based solvers (data not shown). Although the running time of the SLS-based solvers are still quite moderate even at these instances sizes, the much longer runs required to filter out satisfiable instances with high confidence make it difficult to further increase n.

Next, we used the same bootstrap confidence intervals to investigate the significance of differences observed between the scaling models for different solvers. Using this approach, we cannot reject the hypothesis that the differences reflected in the constants for the polynomial models of our three SLS-based solvers seen in Table 2 are insignificant. Furthermore,

[2] We note that RMSE values are sensitive to the absolute magnitude of the differences in the given data. It is therefore difficult to judge different models solely based on absolute RMSE.

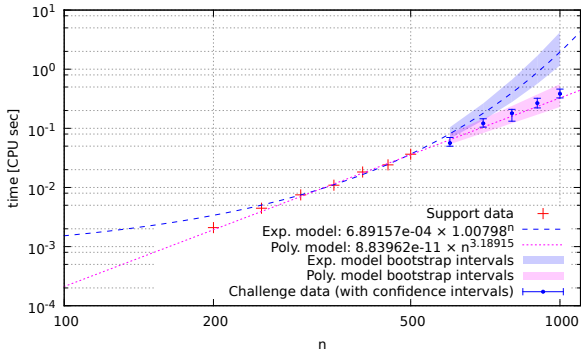

Figure 2: Fitted models of the median running times of WalkSAT/SKC. Both models are fitted with the median running times of walksat solving the SAT instances from the set of 1200 random instances of 200, 250, ..., 500 variables, and are challenged by median running times of 600, 700, ..., 1000 variables.

we observed that the median running times of CCASat [Cai and Su, 2012] (V2014), another prominent SLS-based solver, are consistent with the same polynomial models. Likewise, the bootstrap confidence intervals for march_hi and march_br largely overlap with each other, and we cannot detect sufficient evidence for statistically significant differences in scaling behaviour. On the other hand, the observed running times for kcnfs are inconsistent with the scaling models for march_hi and march_br, confirming that the performance of the march solvers indeed scales significantly better than that of kcnfs. Examining these results in detail, we believe that by substantially increasing the number of instances for each value of n, the bootstrap confidence intervals can likely be narrowed to also demonstrate significant scaling differences between all pairs of SLS-based solvers.

Using the same approach, we investigated the significance of the differences in scaling for each DPLL-based solver when applied to satisfiable and unsatisfiable instances only, respectively. Intuitively, we would expect unsatisfiable instances to be harder, and the differences between the respective scaling models are significant, in that the observed running times for solving unsatisfiable instances are generally inconsistent with the scaling model for the same solver on only satisfiable instances (as illustrated in Figure 3 for march_hi). To further investigate whether the performance difference can be attributed to the constant factor a in the exponential models, we fitted a model of the form $a \cdot b_{sat}^{n}$ of the median running times for solving unsatisfiable instances, where a is a free parameter and b_{sat} is the value of parameter b from the scaling model for satisfiable instances. For all three DPLL-based solvers, the observed running times for solving unsatisfiable instances is consistent with these constrained models, suggesting that, indeed, the scaling on satisfiable and unsatisfiable instances differs only by a constant factor.

Finally, we investigated the question whether our observations regarding the qualitative differences in the scaling of median running times of SLS- and DPLL-based solvers also hold when considering higher quantiles of the distribution of running times. For the DPLL-based solvers, we found no evidence for a substantial change in the ratios between the

Solver	Model	n	Predicted confidence intervals of median running time (sec)		Observed median running time (sec)	
			Poly. model	Exp. model	Point estimates	Confidence intervals
WalkSAT/SKC	Poly$[a, b]$ $a \in [2.58600 \times 10^{-12}, 8.63869 \times 10^{-10}]$ $b \in [2.80816, 3.76751]$	600	$[\mathbf{0.054}, \mathbf{0.081}]$	$[0.067, 0.104]$	0.056	$[0.050, 0.070]$
		700	$[\mathbf{0.083}, \mathbf{0.146}]$*	$[0.137, 0.264]$	0.121	$[0.105, 0.145]$
		800	$[\mathbf{0.122}, \mathbf{0.238}]$*	$[0.277, 0.664]$	0.180	$[0.132, 0.209]$
		900	$[\mathbf{0.170}, \mathbf{0.373}]$*	$[0.565, 1.676]$	0.267	$[0.222, 0.323]$
		1000	$[\mathbf{0.229}, \mathbf{0.557}]$*	$[1.151, 4.200]$	0.385	$[0.327, 0.461]$
BalancedZ	Poly$[a, b]$ $a \in [3.65984 \times 10^{-11}, 4.53094 \times 10^{-9}]$ $b \in [2.60985, 3.41689]$	600	$[\mathbf{0.082}, \mathbf{0.116}]$*	$[0.103, 0.149]$	0.095	$[0.085, 0.102]$
		700	$[\mathbf{0.124}, \mathbf{0.195}]$*	$[0.204, 0.348]$	0.142	$[0.131, 0.154]$
		800	$[\mathbf{0.177}, \mathbf{0.308}]$*	$[0.400, 0.816]$	0.194	$[0.177, 0.212]$
		900	$[\mathbf{0.240}, \mathbf{0.462}]$	$[0.782, 1.915]$	0.270	$[0.231, 0.324]$
		1000	$[\mathbf{0.316}, \mathbf{0.663}]$	$[1.531, 4.493]$	0.353	$[0.307, 0.398]$
probSAT	Poly$[a, b]$ $a \in [5.00843 \times 10^{-12}, 1.02411 \times 10^{-8}]$ $b \in [2.40567, 3.66266]$	600	$[\mathbf{0.050}, \mathbf{0.085}]$	$[0.063, 0.110]$	0.050	$[0.043, 0.059]$
		700	$[\mathbf{0.073}, \mathbf{0.149}]$*	$[0.119, 0.271]$	0.089	$[0.077, 0.105]$
		800	$[\mathbf{0.101}, \mathbf{0.245}]$*	$[0.222, 0.664]$	0.151	$[0.133, 0.197]$
		900	$[\mathbf{0.135}, \mathbf{0.379}]$*	$[0.413, 1.640]$	0.237	$[0.209, 0.295]$
		1000	$[\mathbf{0.174}, \mathbf{0.559}]$*	$[0.771, 4.050]$	0.357	$[0.304, 0.438]$
kcnfs	Exp$[a, b]$ $a \in [3.33378 \times 10^{-5}, 1.07425 \times 10^{-4}]$ $b \in [1.03136, 1.03476]$	450	$[98.326, 122.115]$	$[\mathbf{120.078}, \mathbf{161.444}]$	156.480	$[143.340, 166.770]$
		500	$[327.997, 439.089]$	$[\mathbf{561.976}, \mathbf{889.428}]$*	750.510	$[708.290, 806.130]$
		550	$[971.862, 1402.255]$	$[\mathbf{2622.488}, \mathbf{4901.661}]$*	3896.450	$[3633.630, 4130.915]$
march_hi	Exp$[a, b]$ $a \in [2.90480 \times 10^{-5}, 1.72479 \times 10^{-4}]$ $b \in [1.02928, 1.03433]$	450	$[62.021, 91.787]$	$[\mathbf{74.982}, \mathbf{116.729}]$	112.553	$[101.957, 121.167]$
		500	$[190.395, 333.963]$	$[\mathbf{317.398}, \mathbf{628.498}]$*	564.821	$[508.433, 614.105]$
		550	$[523.034, 1074.244]$	$[\mathbf{1342.438}, \mathbf{3375.460}]$*	2971.450	$[2660.430, 3152.570]$
march_br	Exp$[a, b]$ $a \in [2.61030 \times 10^{-5}, 1.08165 \times 10^{-4}]$ $b \in [1.03064, 1.03466]$	450	$[69.649, 91.290]$	$[\mathbf{84.743}, \mathbf{117.937}]$	112.812	$[101.108, 121, 469]$
		500	$[226.874, 332.773]$	$[\mathbf{385.943}, \mathbf{640.179}]$*	594.095	$[542.564, 620.963]$
		550	$[659.553, 1070.478]$	$[\mathbf{1754.277}, \mathbf{3492.830}]$*	2975.580	$[2544.450, 3179.950]$

Table 3: 95% bootstrap confidence intervals for median running time predictions and observed running times on random 3SAT instances. The instance sizes shown here are larger than those used for fitting the models. Bootstrap intervals on predictions that agree with the observed point estimates are shown in boldface, and those that fully contain the confidence intervals on observations are marked by asterisks (*).

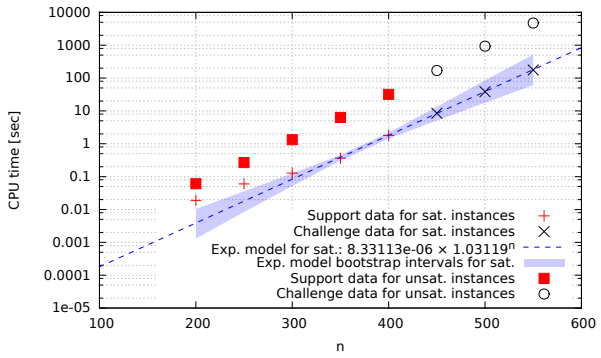

Figure 3: Median running time scaling of march_hi on satisfiable *vs* unsatisfiable instances. We show the scaling model and bootstrap confidence intervals for performance on satisfiable instances, along with observations on support and challenge data for satisfiable and unsatisfiable instances.

median and higher quantiles as n increases; thus, there is no reason to assume any difference in the scaling models of a higher quantile and the median other than a constant factor. For the SLS-based solvers, we noted weak evidence that the ratio between the higher quantiles and the median increases with n, but the observed scaling of the 0.9-quantile is still found to be consistent with a polynomial model and inconsistent with an exponential model (data not shown).

6 Conclusions

In this work, we presented an empirical analysis of the scaling behaviour of several prominent SAT solvers on phase transition uniform random 3-SAT instances. We were surprised to find solid support for low-degree polynomial scaling of the performance of all SLS-based SAT solvers we studied, which stands in stark contrast to the exponential performance scaling of the DPLL-based solvers we considered, even when these were run on satisfiable instances only. As expected, we did find evidence for significant differences in the performance scaling models of DPLL-based solvers on satisfiable and unsatisfiable instances, but these differences could be attributed to a constant factor, suggesting that they cannot be leveraged to obtain reasonably accurate guesses on the unsatisfiability of instances based on long runs of those solvers. However, the qualitative differences between the scaling of SLS- and DPLL-based solvers can be exploited, in that for growing n, it appears to be possible to separate satisfiable from unsatisfiable instances with high and further increasing accuracy based on long runs of SLS-based or hybrid solvers. Furthermore, the polynomial scaling of even high quantiles of the distribution of running times for SLS-based solvers across instance sets suggests that random 3-SAT instances from the solubility phase transition are likely not capturing the difficulty we expect to encounter in the worst case when solving an $\mathcal{NP}$-hard problem. We note that, considering their sizes, these instances are still very hard, compared to almost all types of structured SAT instances, and may therefore still be useful as a benchmark.

To the best of our knowledge, ours is the first study that uses a statistically sound way to assess the scaling of SAT solver performance with instance size, and to discriminate between different scaling models. In the context of our study, we have extended the methodology introduced by [Hoos, 2009; Hoos and Stützle, 2014] in two ways. The empirical scaling analysis we have performed here can easily be applied to other SAT solvers, other distributions of SAT instances (as long as reasonably large sets of instances for each n can be obtained), and to other problems. We believe that doing so can and will produce interesting and useful results that can inspire the design of algorithms and benchmark instance generators as well as, hopefully, theoretical work.

Acknowledgements

We thank the annoymous reviewers for their thoughtful comments and suggestions. We also thank Dimitris Achlioptas, Paul Beame, Henry Kautz, Donald Knuth, Bart Selman and Moshe Vardi for valuable advice regarding previous work on the 3-SAT phase transition and empirical performance of SAT algorithms. This work was supported by Compute Canada, by the Institute for Computing, Information and Cognitive Systems (ICICS) at UBC, and by NSERC, through a Discovery Grant to H.H.

References

[Achlioptas and Peres, 2004] D. Achlioptas and Y. Peres. The threshold for random k-SAT is $2^k \log 2 - O(k)$. *Journal of the American Mathematical Society*, 17(4):947–973, 2004.

[Balint and Schöning, 2012] A. Balint and U. Schöning. Choosing probability distributions for stochastic local search and the role of make versus break. In *SAT*, pages 16–29. Springer, 2012.

[Balint and Schöning, 2014] A. Balint and U. Schöning. probSAT and pprobSAT. *SAT Competition 2014*, page 63, 2014.

[Balint *et al.*, 2012] A. Balint, A. Belov, M. Järvisalo, and C. Sinz. SAT challenge 2012 random SAT track: Description of benchmark generation. *SAT Challenge 2012*, page 72, 2012.

[Biere, 2014] A. Biere. Yet another local search solver and lingeling and friends entering the SAT competition 2014. *SAT Competition 2014*, page 39, 2014.

[Cai and Su, 2012] S. Cai and K. Su. Configuration checking with aspiration in local search for SAT. In *AAAI*, 2012.

[Cheeseman *et al.*, 1991] P. Cheeseman, B. Kanefsky, and W. M. Taylor. Where the really hard problems are. In *IJCAI*, pages 331–337, 1991.

[Coarfa *et al.*, 2003] C. Coarfa, D. D. Demopoulos, A. San Miguel Aguirre, D. Subramanian, and M. Y. Vardi. Random 3-SAT: The plot thickens. *Constraints*, 8(3):243–261, 2003.

[Coja-Oghlan, 2014] A. Coja-Oghlan. The asymptotic k-SAT threshold. In *STOC*, pages 804–813. ACM, 2014.

[Cook, 1971] S. A. Cook. The complexity of theorem-proving procedures. In *STOC*, pages 151–158. ACM, 1971.

[Crawford and Auton, 1996] J. M. Crawford and L. D. Auton. Experimental results on the crossover point in random 3-SAT. *Artificial Intelligence*, 81(1):31–57, 1996.

[Dequen and Dubois, 2004] G. Dequen and O. Dubois. Kcnfs: An efficient solver for random k-SAT formulae. In *SAT*, pages 486–501. Springer, 2004.

[Efron and Tibshirani, 1993] B. Efron and R. J. Tibshirani. *An introduction to the bootstrap*. New York: Chapman and Hall, 1993.

[Franco and Paull, 1983] J. Franco and M. Paull. Probabilistic analysis of the Davis Putnam procedure for solving the satisfiability problem. *Discrete Applied Mathematics*, 5(1):77–87, 1983.

[Friedgut and Bourgain, 1999] E. Friedgut and J. Bourgain. Sharp thresholds of graph properties, and the k-SAT problem. *Journal of the American Mathematical Society*, 12(4):1017–1054, 1999.

[Gent and Walsh, 1993] I. P. Gent and T. Walsh. Towards an understanding of hill-climbing procedures for SAT. In *AAAI*, pages 28–33, 1993.

[Gent *et al.*, 1997] I. P. Gent, E. MacIntyre, P. Prosser, and T. Walsh. The scaling of search cost. In *AAAI*, pages 315–320, 1997.

[Heule and van Maaren, 2009] M. J. H. Heule and H. van Maaren. march_hi: Solver description. *SAT Competition 2009*, pages 27–28, 2009.

[Heule, 2013] M. J. H. Heule. march_br. *SAT Competition 2013*, page 53, 2013.

[Hoos and Stützle, 2014] H. H. Hoos and T. Stützle. On the empirical scaling of run-time for finding optimal solutions to the travelling salesman problem. *European Journal of Operational Research*, 238(1):87–94, 2014.

[Hoos, 2009] H. H. Hoos. A bootstrap approach to analysing the scaling of empirical run-time data with problem size. Technical report, Technical Report TR-2009-16, University of British Columbia, 2009.

[Kirkpatrick and Selman, 1994] S. Kirkpatrick and B. Selman. Critical behavior in the satisfiability of random boolean expressions. *Science*, 264(5163):1297–1301, 1994.

[Kirousis *et al.*, 1998] L. M. Kirousis, E. Kranakis, D. Krizanc, and Y. C. Stamatiou. Approximating the unsatisfiability threshold of random formulas. *Random Structures and Algorithms*, 12(3):253–269, 1998.

[Li *et al.*, 2014] C. Li, C. Huang, and R. Xu. Balance between intensification and diversification: a unity of opposites. *SAT Competition 2014*, pages 10–11, 2014.

[Malitsky *et al.*, 2013] Y. Malitsky, A. Sabharwal, H. Samulowitz, and M. Sellmann. CSCH-based portfolio using CCSat and march. *SAT Competition 2013*, page 28, 2013.

[Mertens *et al.*, 2006] S. Mertens, M. Mézard, and R. Zecchina. Threshold values of random k-SAT from the cavity method. *Random Structures and Algorithms*, 28(3):340–373, 2006.

[Mitchell *et al.*, 1992] D. Mitchell, B. Selman, and H. Levesque. Hard and easy distributions of SAT problems. In *AAAI*, pages 459–465, 1992.

[Parkes and Walser, 1996] A. J. Parkes and J. P. Walser. Tuning local search for satisfiability testing. In *AAAI*, pages 356–362, 1996.

[Roussel, 2011] O. Roussel. Controlling a solver execution with the runsolver tool system description. *Journal on Satisfiability, Boolean Modeling and Computation*, 7:139–144, 2011.

[SatCompetition.org, 2014] SatCompetition.org. The international SAT competitions web page, 2014.

[Selman *et al.*, 1992] B. Selman, H. J. Levesque, and D. G. Mitchell. A new method for solving hard satisfiability problems. In *AAAI*, pages 440–446, 1992.

[Selman *et al.*, 1994] B. Selman, H. A. Kautz, and B. Cohen. Noise strategies for improving local search. In *AAAI*, pages 337–343, 1994.

[Tompkins and Hoos, 2005] D. A. D. Tompkins and H. H. Hoos. UBCSAT: An implementation and experimentation environment for SLS algorithms for SAT and MAX-SAT. In *SAT*, pages 306–320. Springer, 2005.

[Yokoo, 1997] M. Yokoo. Why adding more constraints makes a problem easier for hill-climbing algorithms: Analyzing landscapes of CSPs. In *CP*, pages 356–370. Springer, 1997.