

Article

On the Predictability of Classical Propositional Logic

Marcelo Finger * and Poliana M. Reis

Department of Computer Science, University of Sao Paulo, Sao Paulo, 05508-090, Brazil;
E-Mail: poliana.mreis@gmail.com

* Author to whom correspondence should be addressed; E-Mail: mfinger@ime.usp.br;
Tel.: +55-11-3091-6135/+55-11-3091-6134.

Received: 17 November 2012 / Accepted: 31 December 2012 / Published: xx

Abstract: In this work we provide a statistical form of empirical analysis of classical propositional logic decision methods called SAT solvers. This work is perceived as an empirical counterpart of a theoretical movement, called *the enduring scandal of deduction*, that opposes considering Boolean Logic as trivial in any sense. For that, we study the predictability of classical logic, which we take to be the distribution of the runtime of its decision process. We present a series of experiments that determines the run distribution of SAT solvers and discover a varying landscape of distributions, following the known existence of a transition of easy-hard-easy cases of propositional formulas. We find clear distributions for the easy areas and the transitions easy-hard and hard-easy. The hard cases are shown to be hard also for the detection of statistical distributions, indicating that several independent processes may be at play in those cases.

Keywords: SAT; SAT solver; phase transition; landscape of distributions

1. Introduction

In an article entitled *The enduring scandal of deduction*, Marcello D’Agostino and Luciano Floridi discuss the information content of classical propositional logic [1].

The term *scandal of deduction* was coined by Hintikka [2] to refer to the idea that first-order deductive reasoning gives us no new information. Such an idea has been justified on the fact that, because deductive reasoning is “tautological” or “analytical”, logical truths have no “empirical content” and cannot be used to make “factual assertions”. Many other researchers have considered it false, and thus the scandal.

There is a large support for the idea that, due to undecidability, deductive inference does have the power to extend our knowledge, and thus is generally perceived as highly valuable in epistemic terms [2–4]. However, the support for the information content (also called *epistemic value*) of decidable fragments of first-order is considerably weaker among logicians. In particular, Hintikka himself proposes a distinction between *degrees of informational depth* among logics, in which propositional logic is considered genuinely tautological and “analytical”, thus entirely uninformative. According to that position, this lack of information content is a consequence of the existence of a mechanical decision procedure.

Since that statement of triviality of propositional logic, a full theory of computational complexity has been developed, in which the complexity of the decision of classical propositional inference plays a central role [5–7]. According to such theory, this decision problem is in no sense trivial, and one of the great open questions in theoretical computer science is the existence of a tractable algorithm for it, namely the question $P \stackrel{?}{=} NP$.

The fact that this is still an open question does not imply that propositional logic *may be* trivial. In fact, no matter what the final answer to that question is, it is by now clear that the informational content of propositional logic cannot be considered null, as raised by Floridi [4], and later expanded in the *enduring scandal of deduction* [1].

The latter proposal contains a theoretical description of fragments of a family of sub-classical entailment relations, called *intelim deducibility of depth k* , $\vdash_k$, $k \geq 0$, which “approximate” classical entailment $\vdash$ in the sense that

$$\vdash_0 \subseteq \cdots \subseteq \vdash_k \subseteq \vdash_{k+1} \subseteq \cdots \subseteq \vdash$$

Furthermore, it provides a decision algorithm according to which each of the entailment relations $\vdash_k$ was shown to be decidable in time $O(n^{2k+2})$ which, for a fixed value of k , is considered polynomial time with respect to the number of variables n , and thus tractable. Of course, with respect to the number $k = n$ of approximation steps necessary to approximate a given classical inference, that algorithm is highly exponential ($O(n^{2n+2})$), a lot worse than the usual truth tables ($O(2^n)$).

Other approximations of classical entailment have been proposed in the literature [8–12], but *intelim deducibility* is the first form of approximation with all the following properties:

- (i) it is weaker than classical entailment;
- (ii) it is based on informational notions; and
- (iii) it treats as “uninformative” exactly those inferences that are “analytical” in the strict informational sense, that is, which do not appeal to virtual information.

This “appeal to virtual information” is what, in terms of Natural Deduction inferences, is known as introducing hypothesis that will later be discharged in the proof. In that respect, the first element of the *intelim* approximation, $\vdash_0$, is the only analytical deduction relation that employs no external, virtual hypothesis. All other entailment relations do employ some form of hypothesis generation and are thus considered non-trivial.

This, of course, is just a theoretical view of the complexity of Boolean inference. In this work, we take a more practical view, exploring the complexity in terms of the *statistical predictability* of the computational effort needed to solve a propositional logic decision problem.

In fact, we propose to study the predictability of classical propositional logic, which we take to be the possibility of having a clear distribution of some random variables associated with it. In our case, we study the distribution of the runtime of existing implementation of Boolean solvers.

We start by noting the existence of very efficient Boolean solvers, called SAT solvers, and a competition promoted bi-annually that reveal the best and fastest solvers [13]. SAT was a problem that two decades ago was thought to be impractical, except for problems employing just a few variables, but now existing SAT solvers can deal comfortably with problems with hundreds of thousands, or even a few million variables [14]. In this work, SAT solving is, in empirical terms, what propositional inference is in theoretical terms.

So we take the view that the informational content of a Boolean formula is higher the harder it is to predict how long it will take to reach a decision on the satisfiability of a formula. Note that this is different from simply taking the average time to reach a conclusion. The curve that shows, for a fixed number n of distinct Boolean variables, the average decision time for different sizes of randomly generated input formulas is known as the empirical *complexity profile* of a SAT solver, which typically displays a *phase transition* behaviour [15]. In fact, it was originally discovered that there are hard and easy instances of SAT formulas [16], and the empirical complexity profile associate the on average easy and hard SAT instances to determined regions in the profile.

Here, we go one step further on the empirical analysis of SAT solvers and, instead of analysing average execution times, we analyse the probability distribution of this average time. Each formula is associated with a set of parameters (namely, the rate between the number of clauses and the number of variables) and the information content will be the same for all formulas sharing the same parameters. We consider as “less informative” class of formulas those with a “well behaved statistical distribution” of its decision time, and as a “more informative” class of formulas those with “undetermined or chaotic distribution” time to reach a decision. We will then compare the easy and hard regions according to this new view with that of the complexity profile. We investigate what new insights can be derived from this analysis with respect to the process of SAT solving and the empirical complexity of SAT formulas.

The paper develops as follows. The behaviour of SAT solvers, their empirical complexity profile and the phase transition phenomenon are presented in Section 2. Our experiments to obtain the runtime distribution are described in Section 3 and the results obtained are presented. Then in Section 4 an analysis of the varying landscape of distributions obtained is presented. Finally, we conclude this experimental work in Section 5 and propose future work.

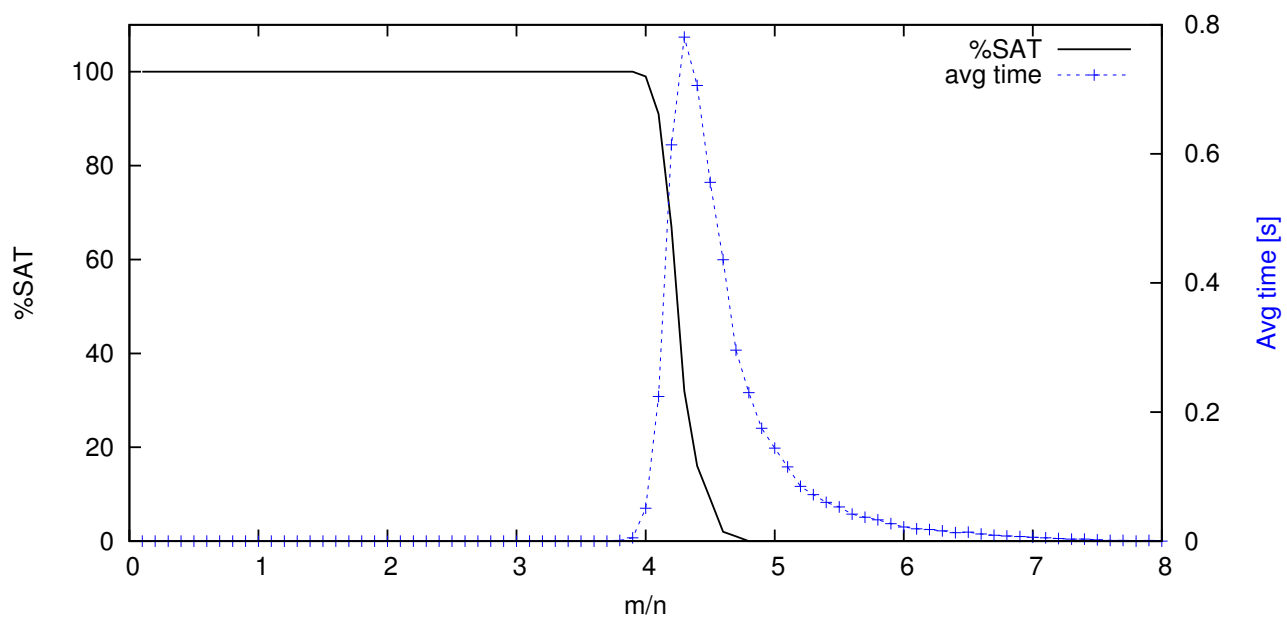
2. SAT Solvers and Phase Transition

Very efficient SAT solvers have appeared in the beginning of the century, with a performance several times faster than previous implementations, such as zChaff [17] and Berkmin [18]; the former was slightly changed in the form of an open source SAT solver, MiniSAT [19], which gave origin to chain of

ongoing improvements, reported in the SAT competitions [13]. For this work, we concentrate on a well established and studied zChaff 64-bit implementation of 2007 [20].

Cheeseman *et al.* [21] presented the phase transition phenomenon for NP-complete problems and conjectured that it is a property of all problems in that class. Gent and Walsh [15] studied phase transition for 3-SAT instances, Boolean formulas containing a conjunction of clauses formed by the disjunction of at most 3 literals. The graphic obtained is what we call the *empirical complexity profile*, which we show in Figure 1.

Figure 1. SAT Empirical Complexity Profile, $n = 200$.



Two curves are simultaneously plotted. For each point a number of randomly generated 3-SAT formulas is submitted to the SAT solver. In all cases, the number of propositional symbols n is fixed; in Figure 1, $n = 200$. We vary the number of clauses m in each 3-SAT formula and the x -axis refers to the rate m/n . For each value of m/n , we generate 100 random formulas and plot both the percentage of satisfiable formulas and the average time to reach a decision.

What is surprising is that, independent of algorithm, machine and the value of n , the general shape of the complexity is the same, and shows that the harder instances concentrate around a point where $m/n = P_t$, the *phase transition point*. For 3-SAT formulas, the profile is as illustrated in Figure 1. When the rate m/n is small (< 3) almost all instances are satisfiable, and when this rate is high (> 6) instances are unsatisfiable, and the decision time remains low at both cases. At the phase transition point P_t , the number of expected satisfiable instances is 50%, which for 3-SAT is $P_t \approx 4.3$. By increasing n , the transition becomes more abrupt, but the phase transition point, where 50% of the instances are satisfiable, remains basically the same and always corresponds to the peak in the measured average time.

3. The Runtime Distribution of a SAT Solver

We now study the statistical distribution of these values; see if they can be approached by known statistical distributions and, in case they can, we obtain the distribution parameters.

3.1. Experiments and Results

With the aim of studying the statistical behaviour of the SAT solvers output, we perform a series of experiments.

The running time of computers programs depends on the computer processor, the processes currently running and various other factors. Therefore instead of using as a parameter measuring the runtime of zChaff, we use the number of operations performed, defined as the sum of the number of implications (or, in the terminology of the resolution method, unit propagation steps) with the number of decisions (branching points) that a resolver performs to solve a particular problem.

A total of 910,000 problems were generated and used as input in zChaff, but initially, only 360,000 problems were generated with the following characteristics:

- All clauses were generated with exactly three literals ($k = 3$);
- The number of atoms (n) ranged in 50, 100, 200 and 300;
- For each value of n , the number of clauses (m) varied such that m/n resulted in 1, 2, 3, 4, 4.3 (at the phase transition point), 5, 6, 7 and 8.

So, we had:

$$360,000 \text{ problems} = 10,000 \text{ problems} \times 4 \text{ (possible values of } n) \times 9 \text{ (values of } \frac{m}{n}).$$

After running all these problems, we collected the output data of each case and plotted graphs showing the number of examples versus number of operations.

Thus, we can see that most graphics resembled a probability distribution.

To check which distribution represented each data set, we performed various Goodness-of-Fit (GoF) tests [22]. As the majority of the observed distributions had observed tails, we chose the Anderson–Darling test.

The Anderson–Darling tests were performed using the tool Minitab [23]. Initially, we could approximate the distribution for certain cases, but in some other cases the experiments were inconclusive. Then, for a better analysis of the inconclusive cases, were generated 550,000 3-SAT instances to be decided by zChaff to try to identify the data distributions. The following are the cases initially inconclusive:

1. $n = 100$, $\frac{m}{n} = 4.3$;
2. $n = 200$, $\frac{m}{n} = 4.3, 5, 6, 7, 8$; and
3. $n = 300$, $\frac{m}{n} = 4.3, 5, 6, 7, 8$.

Resulting in a total of 550.000 instances = 50.000 (problems) $\times$ 11 (cases)

3.2. Summary of Experiments

The purpose of this project was to verify the existence of a default behaviour of the SAT in accordance with the input parameters of the formulas. The set of all experiments are presented in [24]; here we present just a small set of the results used to justify our analysis in Section 4.

Initially we tested 360,000 samples. Of these samples, the compliance with a statistical distribution for the runtime of 110,000 instances was fully inconclusive. It was not clear whether that undefined behaviour was due to zChaff requiring a large number of samples, or if zChaff did behave according to a standard distribution for some set of values of $\frac{m}{n}$. So, it was necessary to generate more samples resulting a total amounting to about 1 million (910,000) instances.

Thus it was possible to identify the behaviour of the SAT in accordance with the 3-SAT formula input parameters variation.

For most cases, the distribution of the SAT solver runtime follows the pattern shown at following figures, independent of the value of n .

3.3. Compliance with a Known Distribution

The Anderson–Darling test measures how well the data follow a particular distribution being evaluated as:

For a given set of data and distribution, the lower the statistical distribution for a better fits the data.

The test parameters of Anderson–Darling are presented as follows:

AD The probability of rejecting the distribution in question

N Number of samples

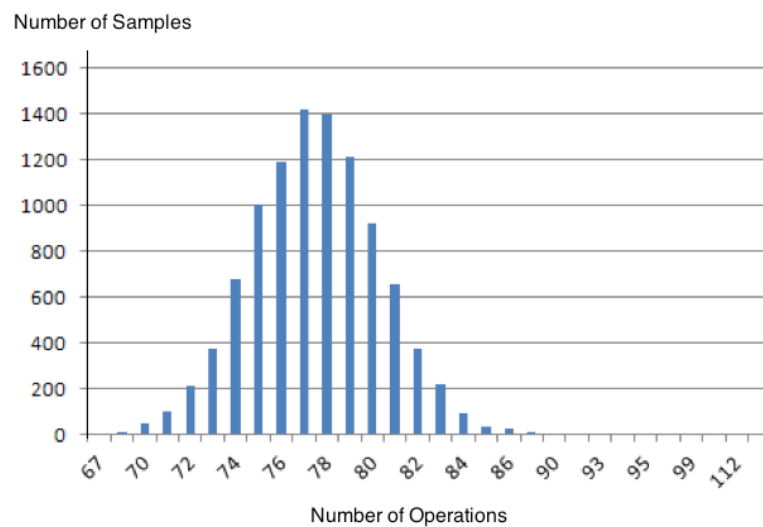
P-value Descriptive level

The known statistical distribution for the tools used are normal, log-normal, exponential, Weibull, extreme value type I, and logistic distributions.

Through graphical analysis of all distributions, we found some patterns of behaviour related to the ratio $\frac{m}{n}$. The cases were:

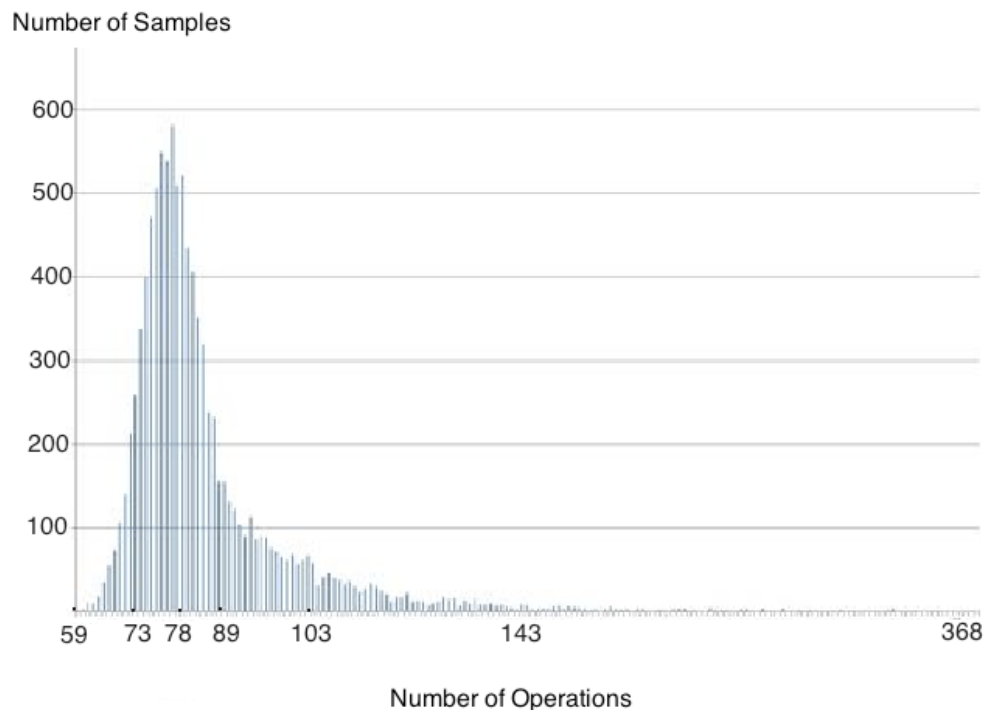
Small values: $\frac{m}{n} \leq 2$ In these cases, zChaff behaves as a discrete distribution of unknown origin where the tails have similar lengths. As an example we present the Figure 2.

Figure 2. Distribution at the output of zChaff running problems with $n = 50$ and $m = 50$, $\frac{m}{n} = 1$.



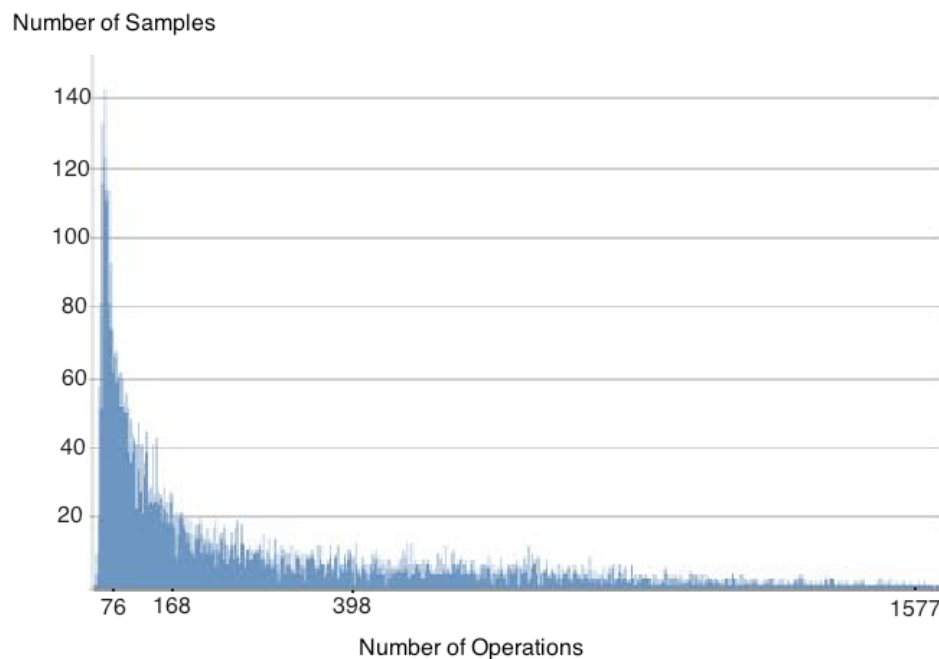
Low-medium: $\frac{m}{n} = 3$ As the ratio $\frac{m}{n}$ increases, the tail of the distribution begins to extend to the right and also increases the frequency of events. The distance between points decreases, approaching a continuous distribution. Figure 3 is an example for $n = 50$ and $m = 150$.

Figure 3. Distribution at the output of zChaff running problems with $n = 50$ and $m = 150$, $\frac{m}{n} = 3$.



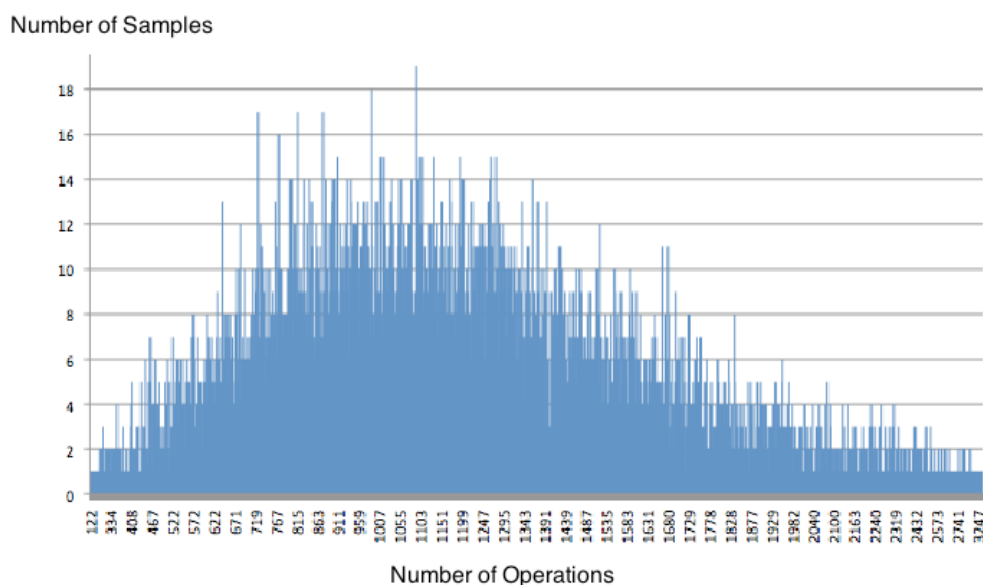
Near the phase transition point Around the phase transition point, there are considerable changes in the distribution. In all tests where $\frac{m}{n} = 4$, a graphical distribution resembles an exponential shape. Figure 4 displays this behaviour for $n = 50$ and $m = 200$.

Figure 4. Distribution at the output of zChaff running problems with $n = 50$ and $m = 200$, $\frac{m}{n} = 4$.



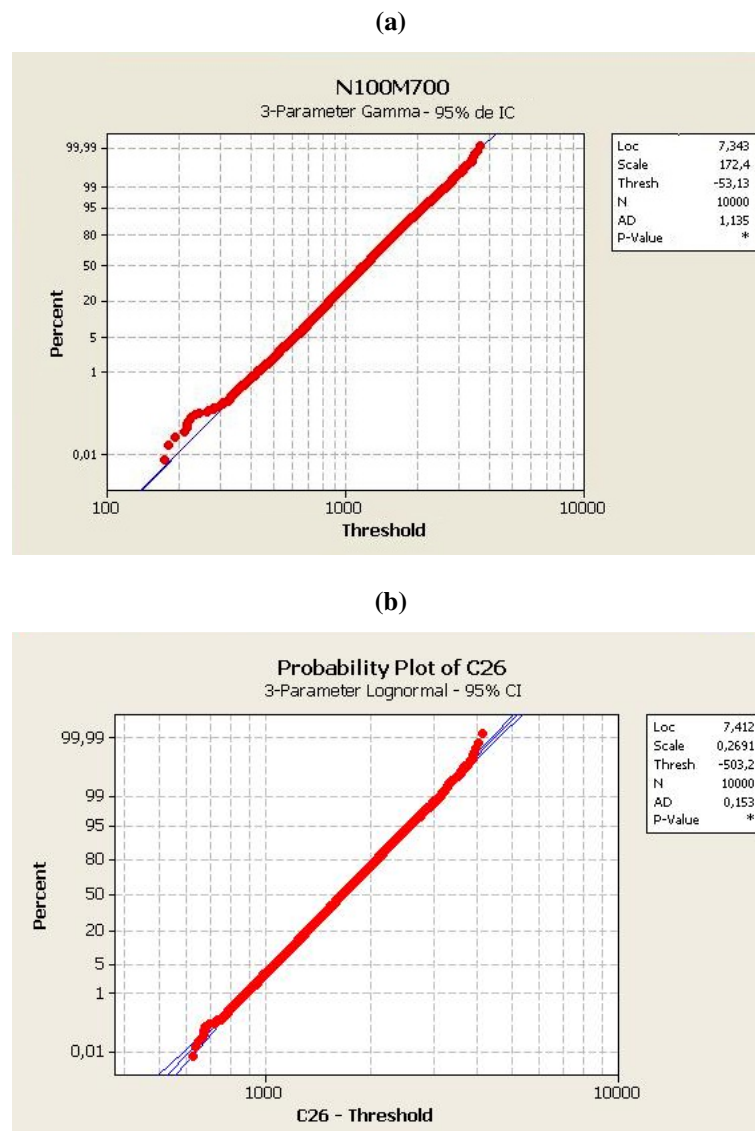
High: $\frac{m}{n} \geq 6$ For cases with values of $\frac{m}{n} = 6, 7$ and 8 , zChaff behaviour approximates a log-normal distribution, as confirmed by GoF tests, as illustrated in Figure 5.

Figure 5. Distribution for $n = 100$ and $m = 700$, $\frac{m}{n} = 7$.



In Figure 6, we can say that the distribution found is a log-normal distribution, because we had $AD = 0.153 < AD_{\text{critical}} = 1.035$. We show the tests of compliance for this data with the gamma distribution with 3 parameters (Figure 6a), and with the log-normal distribution with 3 parameters (Figure 6b). The latter displays a better compliance with the log-normal distribution.

Figure 6. Anderson–Darling tests of compliance with distributions. (a) Compliance with the gamma distribution with 3 parameters; (b) Test of compliance with the log-normal distribution with 3 parameters.

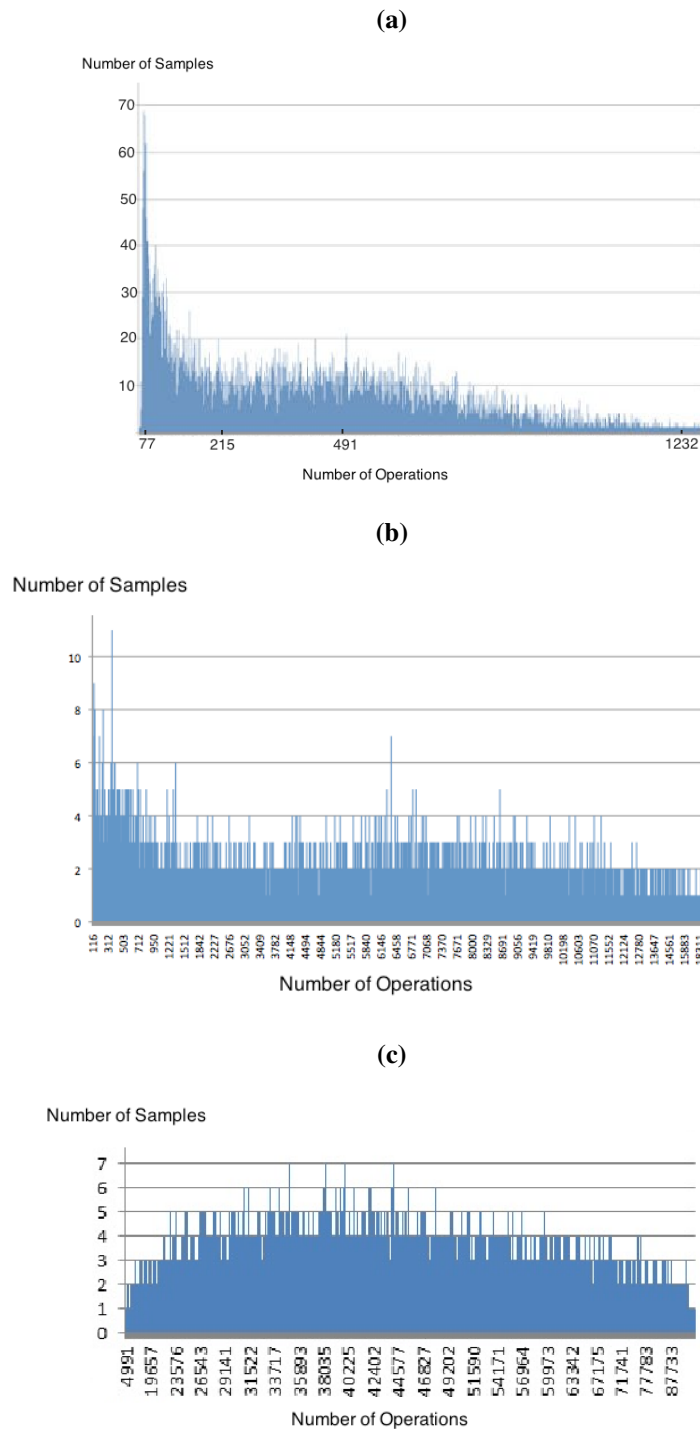


3.4. The Inconclusive Cases

At the phase transition point ($\frac{m}{n} = 4.3$) and points with values of $\frac{m}{n}$ slightly higher, such as $\frac{m}{n} = 5$, the data distribution did not fit any of the patterns listed above. Furthermore, the shape of the data distribution is not preserved for different values of n , a fact that is also very different from the well behaved cases “away from the transition point”.

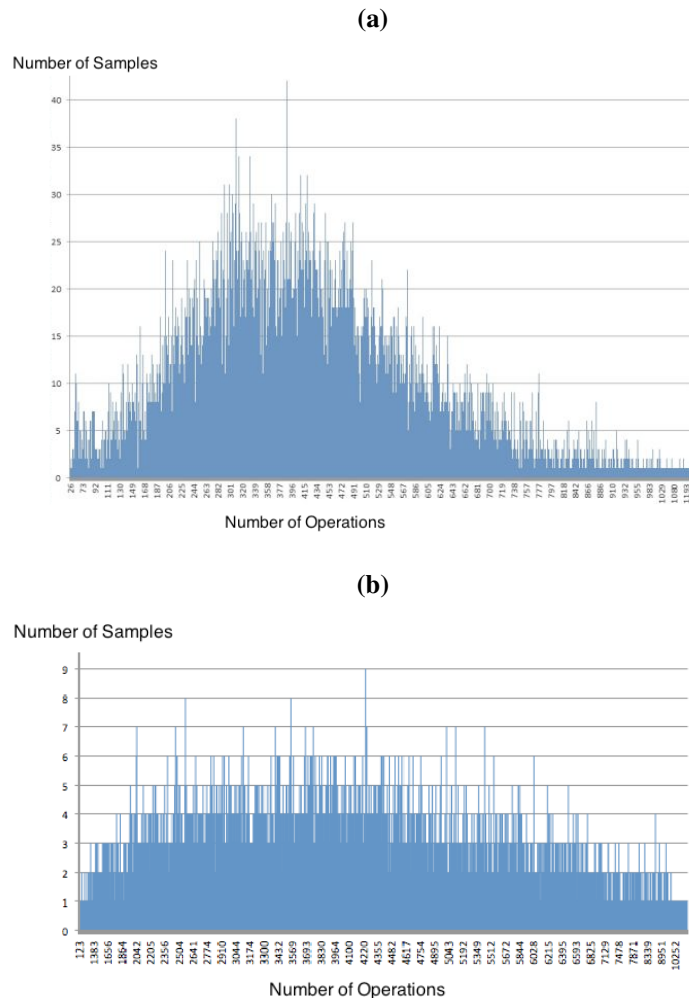
By inspecting the empirical data distributions grouped in Figure 7 we can observe this non-uniform behaviour, by noting the differences of distribution found over 300,000 instances at the phase transition when as in Figure 7a, $n = 50$; as in Figure 7b, $n = 100$; and as in Figure 7c, $n = 200$.

Figure 7. Behaviour unstable at the phase transition point ($\frac{m}{n} = 4.3$). (a) Distribution found for $n = 50$ and $m = 215$, $\frac{m}{n} = 4.3$; (b) Distribution found for $n = 100$ and $m = 430$, $\frac{m}{n} = 4.3$; (c) Distribution found for $n = 200$ and $m = 860$, $\frac{m}{n} = 4.3$.



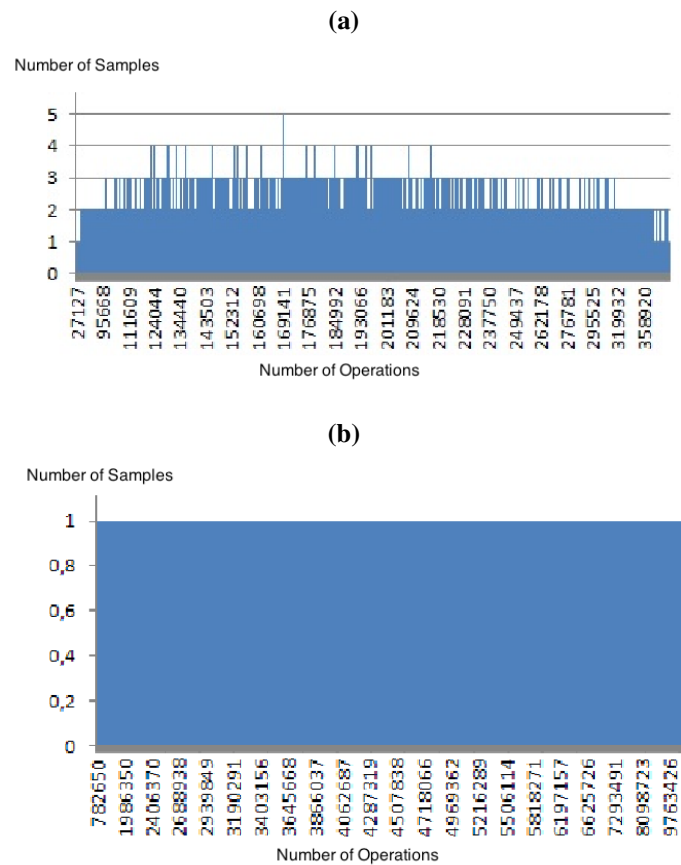
This non-uniform behaviour is not seen a little bit before the phase-transition point, when $\frac{m}{n} = 4$, but the non-uniform behaviour persists a little bit after the phase-transition point, when $\frac{m}{n} = 5$, as illustrated in Figures 8 and 9.

Figure 8. Behaviour unstable near the phase transition ($\frac{m}{n} = 5$). (a) Distribution found for $n = 50$ and $m = 250$, $\frac{m}{n} = 5$; (b) Distribution found for $n = 100$ and $m = 500$, $\frac{m}{n} = 5$.



For a fixed value of $\frac{m}{n} = 5$ we can verify the empirical data distribution obtained when $n = 50$ (Figure 8a), $n = 100$ (Figure 8b), $n = 200$ (Figure 9a) and $n = 300$ (Figure 9b). The non-uniformity is visually apparent, and the compliance tests tell us that none of these empirical data distributions fit any of the distributions known to the tool.

Figure 9. Behaviour unstable near the point of phase shift ($\frac{m}{n} = 5$). (a) Distribution found for $n = 200$ and $m = 1000$; (b) Distribution found for $n = 300$ and $m = 1500$.



A visual inspection of Figures 7–9 indicate that the distributions for $\frac{m}{n} = 4.3$ and $\frac{m}{n} = 5$ do not resemble each other. In fact, when $n = 50$, the distribution for $\frac{m}{n} = 4.3$ has the shape of a spiky exponential, while the distribution for $\frac{m}{n} = 5$ has the shape of a spiky, somewhat distorted normal curve. Even more disturbing is the shape of the curves for $n = 300$. Due to the long times involved, we did not obtain that curve in the phase transition point, but for $\frac{m}{n} = 5$ the distribution visually looks like a constant distribution.

4. Analysis

The observed behaviour with respect to the runtime distribution is summarized in Table 1.

Table 1. Runtime distribution results.

$\frac{m}{n}$	Uniform	Shape
≤ 2	yes	binomial/normal
$\in [3, 4]$	yes	gamma
$\in [4.3, 5]$	no	—
≥ 6	yes	log-normal

We can interpret the results in Table 1 on the values $\frac{m}{n} \leq 2$ and $\frac{m}{n} \geq 6$ in the following way. When $\frac{m}{n} \leq 2$, the instances are mostly satisfiable, and the solutions is “easy” in the sense that there are very few restrictions over a large number of variables, so almost any valuation satisfies the formula.

On the other hand, when $\frac{m}{n} \geq 6$ the problems are mostly unsatisfiable and the distribution is log-normal. A log-normal distribution is a continuous probability distribution of a random variable whose logarithm is normally distributed, and it occurs when the variable can be seen as the product of many independent random variables each of which is positive [25]. We can see this behaviour occurring in the SAT context in case we have several satisfiable formulas of binomial/normal type, which are jointly unsatisfiable. The SAT instance is the union of such jointly unsatisfiable formulas, and the SAT solver tries to, independently, satisfy all such component formulas, which provides the multiplicative effect on the runtimes, and fails only when the last formula is added.

The intermediate values of $\frac{m}{n}$, near the phase transition point, present a variation and a combination of these two cases. A little bit before the phase transition, most formulas are still satisfiable, but a satisfiable valuation is harder to find, so the tail of the distribution grows and the distribution is best modelled as a gamma-distribution, a distribution that usually models waiting-times.

At the phase transition point and a little bit after it, we have formulas that are not independently satisfiable; the set of jointly unsatisfiable formula overlaps in subtle, unpredictable and near-chaotic ways, generating non-uniform distributions with n . Several processes may be co-occurring, and the identification and separation of those processes is needed to obtain a better description. It was not surprising that the phenomenon that was harder to analyse did occur around the phase transition point, as this remains the most interesting area for a SAT solver.

5. Conclusions and Further Work

In this paper we have provided empirical evidence that a significant subset of propositional formulas is far from trivial, and any implication that propositional classical logic is trivial is rightfully labelled as scandalous.

If we understand the predictability of classical logic as the possibility of having a clear distribution of some random variables associated with it, then this work has demonstrated that the runtime of existing SAT solvers is a random variable whose distribution is anything but trivial.

Previous work has concentrated on pointing the existence of an easy-hard-easy separation of empirical complexity of Boolean formulas, mostly based on the average runtime of a SAT solver. Our work went one step further, identifying varying runtime distributions along the same classes of easy-hard-easy problems. The two extremities of easy problems were separated, with the left extremity having a discrete binomial distribution and the light extremity a log-normal distribution, possibly the multiplicative product of those distributions. In the middle, a landscape of distributions was detected, going from left to right: normal, gamma, non-uniform/chaotic/inconclusive and log-normal. The non-uniform distribution coincides with the vicinity of the phase-transition point.

Further work should contemplate a deeper analysis of these results. First, one should try to decompose the processes that seem to be occurring at the area with non-uniform distribution. We could try to identify the existence of clusters and their distributions, for example. Also, the multiplicative nature of the mostly

unsatisfiable cases should be related with that of the satisfiable cases, and we should try to analyse the factors in that multiplication.

We are aware that the kind of empirical analysis presented here is just the start of possible types of analyses that can be done and we hope that the empirical and theoretical analyses of NP-complete problems may eventually cross-fertilise.

Acknowledgements

This work was supported by Fapesp Thematic Project 2008/03995-5 (LOGPROB). Marcelo Finger was partially supported by CNPq grant PQ 302553/2010-0.

References

1. D'Agostino, M.; Floridi, L. The enduring scandal of deduction. Is propositional logic really uninformative? *Synthese* **2009**, *167*, 271–315.
2. Hintikka, J. *Logic, Language Games and Information. Kantian Themes in the Philosophy of Logic*; Clarendon Press: Oxford, UK, 1973.
3. Dummett, M. *The Logical Basis of Metaphysics*; Duckworth: London, UK, 1991.
4. Floridi, L. Is information meaningful data? *Philos. Phenomen. Res.* **2005**, *70*, 351–370.
5. Cook, S.A. The Complexity of Theorem-Proving Procedures. In *Conference Record of Third Annual ACM Symposium on Theory of Computing (STOC)*; ACM: Ohio, USA, 1971; pp. 151–158.
6. Papadimitriou, C.H. *Computational Complexity*; Addison-Wesley: 1994.
7. Arora, S.; Barak, B. *Computational Complexity: A Modern Approach*, 1st ed.; Cambridge University Press: Cambridge, UK, 2009.
8. Schaerf, M.; Cadoli, M. Tractable Reasoning via Approximation. *Artif. Intell.* **1995**, *74*, 249–310.
9. Dalal, M. Anytime Families of Tractable Propositional Reasoners. In *International Symposium of Artificial Intelligence and Mathematics AI/MATH-96*, Fort Lauderdale, Florida, USA, 1996; pp. 42–45.
10. Finger, M.; Wassermann, R. Approximate and Limited Reasoning: Semantics, Proof Theory, Expressivity and Control. *J. Logic Comput.* **2004**, *14*, 179–204.
11. Finger, M.; Gabbay, D. Cut and Pay. *J. Log. Lang. Inf.* **2006**, *15*, 195–218.
12. Finger, M.; Wassermann, R. The Universe of Propositional Approximations. *Theor. Comp. Sci.* **2006**, *355*, 153–166.
13. The international SAT Competitions web page. Available online: <http://www.satcompetition.org/> (accessed on 31 December 2012).
14. These very large problems arise from industrial applications or may be randomly generated.
15. Gent, I.P.; Walsh, T. The SAT Phase Transition. *ECAI94—Proceedings of the Eleventh European Conference on Artificial Intelligence*; John Wiley & Sons: Amsterdam, The Netherlands, 1994; pp. 105–109.
16. Mitchell, D.; Selman, B.; Levesque, H. Hard and Easy Distributions of SAT Problems. *AAAI92—Proceedings of the 10th National Conference on Artificial Intelligence*, San Jose, California, USA, 1992; pp. 459–465.

17. Moskewicz, M.W.; Madigan, C.F.; Zhao, Y.; Zhang, L.; Malik, S. Chaff: Engineering an Efficient SAT Solver. In *Proceedings of the 38th Design Automation Conference (DAC'01)*, Las Vegas, Nevada, USA, 2001; pp. 530–535.
18. Goldberg, E.; Novikov, Y. Berkmin: A Fast and Robust SAT Solver. *Design Automation and Test in Europe (DATE2002)*, Paris, France, 2002, pp. 142–149.
19. Eén, N.; Sörensson, N. An Extensible SAT-solver. *SAT 2003, LNCS*; Springer, S. Margherita Ligure - Portofino, Italy, 2003; Volume 2919, pp. 502–518.
20. Available at: <http://www.princeton.edu/~chaff/zchaff/zchaff.64bit.2007.3.12.zip>.
21. Cheeseman, P.; Kanefsky, B.; Taylor, W.M. Where the really hard problems are. *12th IJCAI*; Morgan Kaufmann: Sydney, Australia, 1991; pp. 331–337.
22. Taylor, J. *An Introduction to Error Analysis: The Study of Uncertainties in Physical Measurements*; Physics–chemistry–engineering, University Science Books, 1997.
23. Minitab. Available online: <http://minitab.com> (accessed on 31 December 2012).
24. Reis, P.M. Analysis of Runtime Distributions in SAT solvers (in Portuguese). Master's Thesis, Department of Computer Science, Institute of Mathematics and Statistics, University of São Paulo, Sao Paulo, Brazil, 2012.
25. Limpert, E.; Stahel, W.A.; Abbt, M. Log-normal distributions across the sciences: Keys and clues. *BioScience* **2001**, *51*, 341–352.

© 2013 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/3.0/>).