Machine Learning and Artificial Intelligence
J.-L. Kim (Ed.)

doi:10.3233/FAIA220420

Empirical Evaluation of the Asymptotic Behavior of the Analysis Complexity of Hard Random 3-CNF Formulas

Sergey I. UVAROV[1]
Institute of Control Science RAS, Moscow, Russia

Abstract. This paper continued the study of the complexity behavior of the satisfiability analysis of hard random formulas given in the conjunctive normal form (CNF). In the 3-CNF formula, each clause contains three literals of logical variables. The number of logical variables in the formula is *N*. In this paper, the SAT solver is improved by introducing equality reduction and pure literal identification procedures. The solver improvement reduced the exponent (with base 2) from *N*/20.86 to *N*/21.41 with an *R*=4.6 ratio of the number of clauses to the number of variables. The results show that the efficiency of the pure literals identification procedure decreases as *R* increases. An important part of the study is an empirical estimation of the algorithmic complexity of the SAT problem with large number of variables. The proposed method gives a convenient lower bound on the complexity of analysis for random 3-CNF formulas. We estimated the algorithmic complexity for the range *N*=256÷8192. The exponential dependence of complexity on N for random 3-CNF formulas at a fixed value of R is demonstrated in this range.

Keywords. SAT-problem, CNF, clauses, literals, variables, complexity

1. Introduction

This paper continues the research on the question of whether the algorithmic complexity of the SAT problem is polynomial or exponential in the number of Boolean variables. The SAT problem is to establish the property of a logical formula to be satisfiable or unsatisfiable. Our study considers logical formulas given in the conjunctive normal form (CNF). CNF is the set of clauses. In 3-CNF any of clauses contain no more than three literals of logical variables (literal is the variable itself or it's negation).

Excellent surveys of the problems under consideration are presented in [1-3]. The SAT problem can be considered as the kernel of the class of NP-complete problems. In turn, the problems of this class cover a wide class of computationally complex problems related to optimization, verification, identification, and so on. SAT problem is directly related to automated theorem proving (also known as ATP or automated deduction). Due to the wide application of NP-complete problems in artificial intelligence systems, the question of estimating their algorithmic complexity is considered very important. Many publications are devoted to the algorithmic complexity of the SAT problem. References to the base papers are given in surveys [1-3].

[1] Corresponding Author: Sergey I. UVAROV, Institute of Control Science RAS, Moscow, Russia; E-mail: uvarov53@gmail.com.

The proposed paper develops a study of the algorithmic complexity of the analysis of the satisfiability/unsatisfiability property of hard random 3-CNF formulas. The development is carried out in the direction indicated in [4-6].

It is assumed that the study of such formulas will help to answer the question whether the algorithmic complexity of the SAT problem is polynomial or exponentially bounded with respect to the number of variables N.

In the study, we will try to evaluate the complexity of the analysis of 3-CNF formulas over a wide range of the number N of variables up to 2^{13} (8192).

2. Hard random 3-CNF formulas

The object of the study are 3-CNF formulas generated randomly. For generating 3-CNF formulas we use a very convenient method [1-3]. The total number of possible three literal clauses is $8\binom{N}{3} \sim O(N^3)$ where N is the number of logical variables. Each clause is numbered and is a disjunction of three terms (literals).

In the process of building the formula we randomly select required number M of clauses. A formula is satisfiable if there is some assignment for logical variables that gives it the value TRUE.

For random 3-CNF formulas the property (satisfiable/unsatisfiable) correlates with the ratio R of the number of clauses to the number of variables. It was shown [7, 8] that as the number of variables increases with $R \leq 3.52$, unsatisfiable formulas are generated negligibly rarely. On the other hand, satisfiable formulas are generated negligibly rarely if $4.51 \leq R$. The most difficult (hard) to analyze are the random 3-CNF formulas with $R \approx 4.3$ [4-6].

In this paper we describe experimental results relating to R=4.6 and R=5.0. Let us note that in full accordance with the theoretical prediction all constructed random 3-CNF formulas turned out to be unsatisfiable.

3. Improvement in the SAT solver

A previous study [6] was conducted using a SAT solver 1 based on the well-known Davis-Putnam-Logemann-Loveland (DPLL) algorithm [1-3] with the *BinRes* clause learning procedure described in [9]. This procedure assumes that all resolutions whose resolvents are units or two-literal clauses are executed during formula analysis.

In this paper we use improved SAT solver 2. This solver 2 includes procedures of e*quality reduction* and *pure literal* identifying [9]. Both procedures are aimed at reducing the number of variables handled by the solver as quickly as possible.

The possibility of e*quality reduction* arises if in the formula we find (for example) a pair of two-literal clauses (2-clauses) such as $\{a, \neg b\}$ and $\{\neg a, b\}$ (i.e., $a \rightarrow b$ and $b \rightarrow a$, hence $a \leftrightarrow b$). In this case we can replace variable b by variable a.

If there is a single literal among the two possible literals of a variable in the clause set, then we have found a *pure literal*. Then, obviously, we can assign a TRUE value to the pure literal.

Incorporating these procedures into the SAT solver reduces the number of branches of the analysis tree. However, depending on the implementation of the solver, these procedures may require significant additional computations.

Any time we need to choose a variable to branch, the SAT solver chooses the variable that is represented in the maximum number of shortest clauses.

4. Experiment on complete analysis of 3-CNF formulas

Experimental results are presented in the Table 1. For comparison, Table 1 shows the results for both solvers 1 and 2. This allows us to evaluate the contribution of the introduced procedures for identifying equal literals and identifying *pure literals*.

Table 1. Results of complete analysis of 3-CNF formulas by two solvers.

N	Solver 1		Solver 2	
	***R*=4.6**	***R*=5.0**	***R*=4.6**	***R*=5.0**
256	896·10^1 (4,0%) 13.13	301·10^1 (1.0%) 11.59	812·10^1 (1.0%) 12.99	268·10^1 (0.3%) 11.38
320	799·10^2 (2.2%) 16.29	191·10^2 (1.0%) 14.22	625·10^2 (3.8%) 15.93	158·10^2 (2.5%) 13.95
384	767·10^3 (3.1%) 19.55	145·10^3 (19%) 17.15	575·10^3 (11%) 19.13	105·10^3 (7.0%) 16.69
448	592·10^4 (2.1%) 22.49	719·10^3 (5.0%) 19.46	364 10^4 (11%) 21.8	570·10^3 (4.3%) 19.12
512	422·10^5 (12%) 25.34	458·10^4 (3.0%) 22.13	334 10^5 (2.5%) 24.99	364·10^4 (0.6%) 21.8

The cells of Table 1 show $\mu E_J(N, R)$, J=1,2, median (for 65 instances) values of the complexity of proving unsatisfiability of the formulas, expressed through the number of branches constructed. The notation $10\text{^}X=10^X$ is used. The index J corresponds to the number of the solver used (solver 1 or solver 2). Then the deviation of the median value $\mu E_J(N, R)$ from the value given by the obtained approximation formula (for the corresponding solver) is given in brackets, expressed as a percentage.

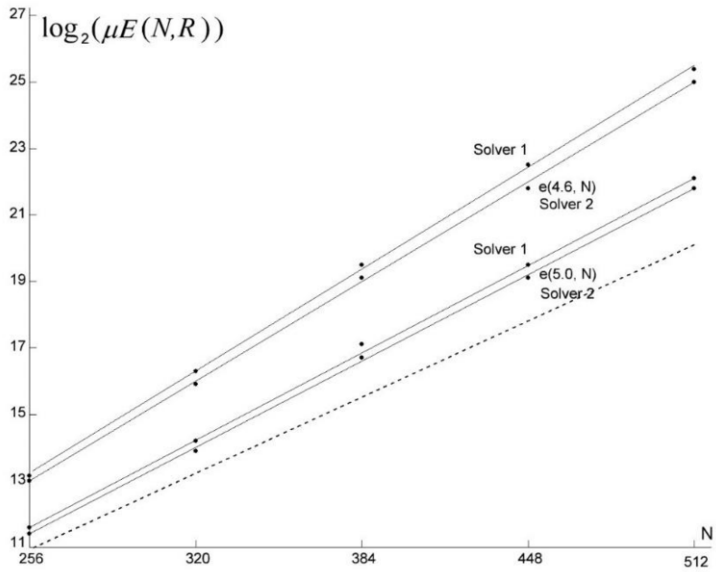

Figure 1. Graphical representation of the empirical complexity values $\mu E_J(N, R)$.

After that the value of $\log_2(\mu E_J(N, R))$ is given in the second line of cells.

Experimental results presented in the Table 1 are illustrated in Figure 1. In logarithmic scale are presented values of $\mu E_J(N, R)$. The data corresponding to solver 2 is approximated by the least squares method as:

$$e(4.6, N) = N/21.41 + 1.046, \quad (1)$$

$$e(5.0, N) = N/24.61 + 0.982. \quad (2)$$

In [5] an approximation formula was obtained for solver 1:

$$\mu E_1(N, R) \approx 2 \cdot 2^{N/(8.4R-17.81)}. \quad (3)$$

It was not possible to obtain a similar formula for solver 2. This is because the efficiency of the pure literal identification procedure increases with decreasing R value, while the efficiency of the *BinRes* procedure decreases. For solver 2, we have separate approximation formulas for each value of R. We obtain:

$$\mu E_2(N, 4.6) = 2^{e(4.6, N)} \approx 2.045 \cdot 2^{N/21.41}, \quad (4)$$

$$\mu E_2(N, 5.0) = 2^{e(5.0, N)} \approx 1.975 \cdot 2^{N/24.61}. \quad (5)$$

The experiment showed that the introduction of the procedures of equality reduction and identification of pure literals significantly reduces the number of analyzed branches. For R=4.6 we get 2 to the power of N/21.41 versus the former N/20.86 (for R=5.0 we have N/24.61 versus N/24.2).

The experiment also showed that in the range N=256÷512, the improved algorithm (solver 2) shows an exponential on N increase in the complexity of the analysis of random 3-CNF formulas generated with a fixed ratio R (R=4.6 and R=5.0) of the number of clauses to the number of variables.

5. Computational experiment with an enlarged number of variables

We want to get a lower bound on the complexity for the number N of variables that significantly exceeds 512. However, when the number of variables exceeds 512, it becomes difficult to perform a computational experiment with complete analysis of 3-CNF formulas because of the huge amount of calculations.

To get around this obstacle, the following technique is used. Our approach is to select a limited number of branches from the huge tree of formula analysis. For these branches the median value $\mu\mathcal{E}(N, R)$ of the number of variables that were used to branch in the tree is calculated. Each of these (used for branching) variables corresponds to a node in the analysis tree. From the obtained value of $\mu\mathcal{E}$ we can try to estimate the total number of branches in the complete analysis tree. The first thought is to use $2^{\mu\mathcal{E}}$ as an estimate: $\hat{E}(N, R) = 2^{\mu\mathcal{E}(N, R)}$.

In the conducted experiment, 1024 random 3-CNF formulas were generated (with ratio R=4.6) for each of the desired numbers of 256, 512, 1024, 2048, 4096, and 8192 variables. In the analysis tree of each formula 2^10 (1024) branches are selected (using

random numbers). For each of the considered numbers of N variables the median value $\mu\mathcal{E}(N, 4.6)$ of the number of nodes among 2^20 (1048576) branches is obtained. The computational experiment was conducted using solver 2. The results of the experiment are presented in Table 2.

Table 2. The median number of nodes in the branch as a function of N.

N	256	512	1024	2048	4096	8192
$\mu\mathcal{E}$	11	20	39	76	150	298
$\mu\mathcal{E}$"	10.88	20.15	38.68	75.74	149.88	298.14

The linear least-squares approximation $\mu\mathcal{E}"(N, 4.6) = \alpha \cdot N + \beta$ for the first two lines of the table as gives us the values $\alpha = 0.0362$ and $\beta = 1.61$.

The approximation values are presented in the third row of Table 2. Linear approximation $\mu\mathcal{E}"(N, 4.6)$ provides a sufficiently high correlation with empirical data.

It is necessary to explain the way of selecting the analyzed branches. A random number is generated each time a branching variable is assigned. If this number is 1, the value of the branching variable is TRUE, if 0, it is FALSE. In the case of an unsatisfiable formula, the branch ends in a contradiction, and we fix the number of branching variables (nodes).

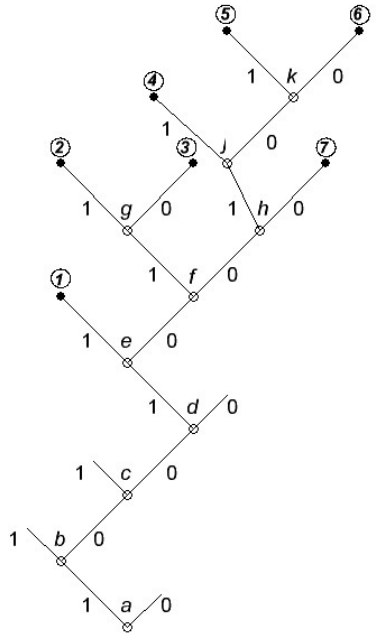

Figure 2. Predominant choice of short branches.

Note that the branch selection method used has the unavoidable drawback that it predominantly selects short branches. This drawback illustrated on Figure 2. This figure shows an example of a fragment of the analysis tree. It is easy to see that the probability of choosing branch 1 is equal to the total probability of choosing branches 2÷7. Also, the

probability of choosing branch 7 is equal to the total probability of choosing branches 4÷6. This property of the proposed branch selection method can be seen as a way to obtain a bottom estimate of the complexity of the analysis of random 3-CNF formulas.

Having a method for finding a lower estimate of the complexity of the analysis (performed by a given solver) of generated (with a given value of the ratio R) 3-CNF formulas is very valuable. The exponential nature of the lower estimate is evidence in favor of the exponential nature of the complexity of the analysis (performed by fixed solver) of generated random 3-CNF formulas.

In conducted experiment the resulting lower estimate $\hat{E}_2(N, 4.6)$ gives us:

$$\hat{E}_2(N, 4.6) = 2^{\mu\varepsilon''(N,\,4.6)} \approx 3.05 \cdot 2^{N/27.62}. \quad (6)$$

In Figure 1, logarithmic value of $\hat{E}_2(N, 4.6)$ is represented by the bold dashed line.

The empirical distribution of the number of branches (vertical axes) by the number of nodes (horizontal axes) for 3-CNF formulas with N=256, 512, 1024, 2048, 4096, and 8192 is shown in Figure 3 and Figure 4.

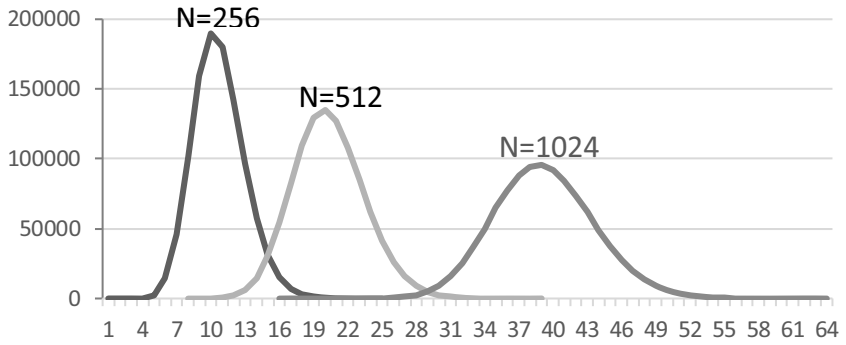

Figure 3. Distribution of the number of branches by the number of nodes for N=256, 512, 1024.

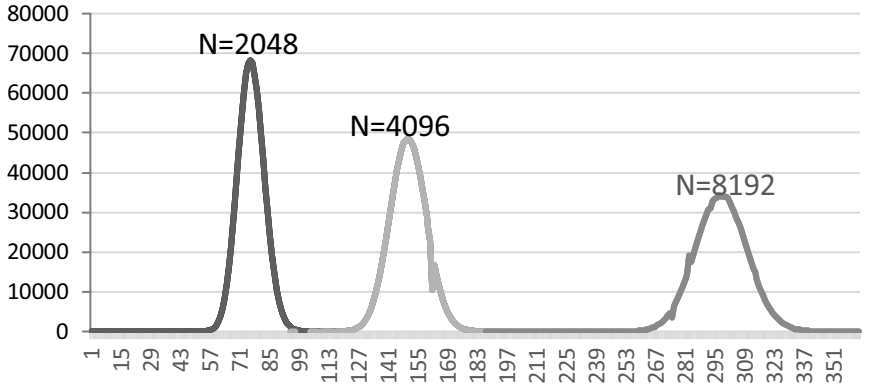

Figure 4. Distribution of the number of branches by the number of nodes for N=2048, 4096, 8192.

The distribution of the number of nodes in the analysis tree for a given N number of variables is close to a normal distribution.

6. Conclusions

We performed the computational experiment in two modes. The first mode is the complete analysis of a Boolean formula. Complete analysis implies the possibility of proving the unsatisfiability of the formula by considering all branches of the proof tree.

The second mode is a partial (randomized) analysis of formulas with an extremely large (for a complete analysis) number of variables.

The first mode focuses on the question whether or not the introduction of some additional procedures into the solver can violate the exponential behavior of the algorithmic complexity estimate $\mu E(N, R)$.

The introduction of equality reduction and pure literal identifying procedures into the solver significantly reduced the number $\mu E(N, R)$ of branches analyzed by the solver. For hard random 3-CNF formulas, the value of $\mu E(N, 4.6)$ decreased by a factor of 1.4 on average. At the same time, the behavior of the estimate $\mu E(N, R)$ for the range N=256÷512 remains exponential.

For the second method of the computational experiment, we presented a method for obtaining the lower bound $\hat{E}(N, R)$ of the complexity of the analysis of the 3-CNF formula using a concrete SAT solver. We used solver 2.

This method involves constructing a limited number of branches from the full analysis tree. The set of branches to be analyzed is chosen randomly. The number of branching nodes in the selected branches is used in forming the estimate $\hat{E}(N, R)$.

The presented method allowed us to estimate the complexity of analysis of formulas with the number of variables in the range from 256 to 8192.

The estimate obtained $\hat{E}(N, R)$ is not exact, it is greatly underestimated. Empirically, it depends exponentially on N. For solver 2 we obtained $\hat{E}(N, 4.6) \approx 3 \cdot 2^{N/27.62}$. This suggests that a more accurate estimate $\mu E(N, R)$ should also be exponential from N.

References

[1] Biere A, Heule M, Maaren H. Walsh T. Handbook of Satisfiability. IOS Press 2009. p. 1-966.
[2] Biere A, Heule M, Maaren H. Walsh T. Handbook of Satisfiability 2nd. IOS Press 2021. p. 1-1484 (in 2 parts).
[3] Gomes C, Rautz H, Sabharwal A, Selman B. Satisfiability Solvers. in Handbook of Knowlege Representation, Elsevier B.V. 2008. p. 88-134.
[4] Crawford J, Auton I. Experimental Results on the crossover point in satisfiability problems. Proceeding of AAAI-93, Washington, DC. 1993. p. 21-27.
[5] Mitchell D, Selman B, Levesque H. Hard and easy distribution of SAT problem. Proceeding of AAAI-92, San Jose, CA. 1992, p.459-465.
[6] Uvarov SI. About Strong dependence of the complexity of analysis of the random 3-CNF formulas on the ratio of number of clauses to the number of variables. Proc. of MMBD 2021 and MLIS 2021, FAIA, IOS Press, 2021: p. 496-501.
[7] Kaporis Alexis C, Kirousis Lefteris M, Laias Efthimios G. The probabilistic analysis of a greedy satisability algorithm. Random Structures & Algorithms 2006; 28(40):444-480.
[8] Merzard M, Parisi G, Zecchina R. Analytic and Algorithmic Solution of Random Satisfiability Problems SCIENCE V. 2002; 297: 812-815.
[9] Bacchus F. Enhancing Davis Putnam with extended binary clause reasoning. In: Proc. AAAI, AAAI Press (2002) 613–619.