Empirical Hardness Analysis of MaxSAT

Max Verbinnen[0009-0002-7696-2954] and Sander Verwimp[0009-0007-9458-9192]
Supervisor: prof. dr. Luc De Raedt
Mentors: Jaron Maene and Vincent Derkinderen

Department of Computer Science, KU Leuven, Belgium
{max.verbinnen, sander.verwimp}@student.kuleuven.be

Over the past 35 years, significant research has been dedicated to examining the hardness of random instances of NP-complete problems [3,4,8]. SAT, as the quintessential example of an NP-complete problem, has received particular attention in this area.

We will focus instead on the maximum satisfiability problem MaxSAT, which involves finding an assignment that satisfies the maximum number of clauses [1]. The literature on the hardness of MaxSAT, however, is comparatively sparse. This is unfortunate, as MaxSAT is a fundamental problem in computer science with numerous practical applications such as scheduling [2]. We therefore extend the existing research on MaxSAT by conducting empirical analyses similar to those already performed for SAT.

Those previous studies primarily investigated the hardness of SAT in terms of *order* and *density* [9,3,4]. Here, the order of an instance refers to the number of variables, while the density is defined as the ratio of the number of clauses to the order [4]. It is well established that SAT instances are easily solvable at both low and high densities [3]. Therefore, the most challenging problems are found in the intermediate density range. Mitchell et al. empirically demonstrated that the hardest instances occur at a density of approximately 4.3, which is nearly the exact density at which on average 50% of instances are satisfiable [9]. We will refer to this 50% point as the *crossover point* [5]. The transitions from easy to hard and back to easy as density increases are characterised as an easy-hard-easy *phase transition* [10].

To initiate the hardness analysis of MaxSAT, we set up a series of experiments utilising the *state-of-the-art* MaxSAT solver RC2 as implemented in PySAT [6,7]. As a measure of hardness, we consider the execution time. More precisely, we consider the median execution time of multiple repetitions performed by the solver on random formulas with the same order and density, as was done for SAT [4].

What exactly is the relationship between the hardness of a MaxSAT problem and its two crucial parameters: order and density? That is the first question we aim to answer. Zhang observed that the hardness of a MaxSAT problem increases when the density increases [11]. Our experiment builds on this finding by also including order as a parameter. We measure the median execution time of the solver for formulas with varying orders and densities. The results of this experiment are presented in Figure 1, with the vertical axis for execution time displayed on a logarithmic scale.

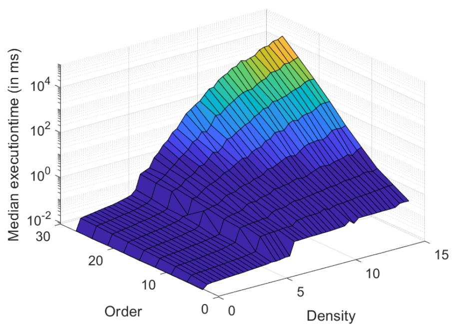

Fig. 1: Median execution time of RC2 as a function of density and order

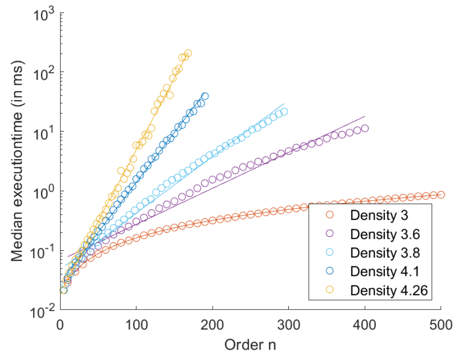

Fig. 2: Median execution time of RC2 as a function of order

This figure confirms the easy-hard phase transition described by Zhang [11]. MaxSAT is easily solvable up until it reaches the crossover point at density ≈ 4.26. Beyond the crossover point, unlike SAT, MaxSAT remains in the hard zone. Our experiment also shows that increasing the order results in longer execution times.

However, Coarfa et al. argued that the terms "easy" and "hard" are used too loosely when describing the phase transition [4]. We therefore conduct a second experiment studying the time complexity to determine where exactly the transition from easy to hard occurs, with easy and hard now defined as polynomial and exponential time complexity, respectively. For this experiment, we consider the execution time of the solver as a function of solely the order, consistent with the experimental setup of Coarfa et al.

The result of this experiment is shown in Figure 2 for several fixed densities. The execution time is again displayed on a logarithmic scale. For density 3, the execution time follows a polynomial growth, which appears as a logarithmic curve in our figure. For density 4.26, on the other hand, the linear curve clearly indicates exponential growth.

Further analysis reveals a gradual transition from polynomial to exponential complexity between densities 3 and 4.26. Starting at density 3.6, the solver follows a superpolynomial but subexponential trend. This subexponential trend continues up to density 4.2, after which the solver's complexity becomes exponential.

In conclusion, our experiments show that MaxSAT becomes harder as the density or order increases. They also reveal a novel phase transition from polynomial to exponential time complexity prior to the crossover point. Future work will need to determine the extent to which these results are solver-dependent. Finally, for our thesis we have developed a MaxSAT game[1] with varying difficulty levels based on the aforementioned results.

[1] The MaxSAT game is playable at https://maxsat.vercel.app/.

References

1. Biere, A.: Handbook of Satisfiability. Frontiers in artificial intelligence and applications, IOS Press (2009), https://books.google.be/books?id=YVSM3sxhBhcC
2. Bofill, M., Garcia, M., Suy, J., Villaret, M.: Maxsat-based scheduling of b2b meetings. In: Michel, L. (ed.) Integration of AI and OR Techniques in Constraint Programming. pp. 65–73. Springer International Publishing, Cham (2015)
3. Cheeseman, P., Kanefsky, B., Taylor, W.M.: Where the really hard problems are. In: Proceedings of the 12th International Joint Conference on Artificial Intelligence - Volume 1. p. 331–337. IJCAI'91, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1991)
4. Coarfa, C., Demopoulos, D.D., Aguirre, A.S.M., Subramanian, D., Vardi, M.Y.: Random 3-sat: The plot thickens. In: Dechter, R. (ed.) Principles and Practice of Constraint Programming – CP 2000. pp. 143–159. Springer Berlin Heidelberg, Berlin, Heidelberg (2000)
5. Crawford, J.M., Auton, L.D.: Experimental results on the crossover point in satisfiability problems. In: Proceedings of the Eleventh National Conference on Artificial Intelligence. p. 21–27. AAAI'93, AAAI Press (1993)
6. Ignatiev, A., Morgado, A., Marques-Silva, J.: PySAT: A Python toolkit for prototyping with SAT oracles. In: SAT. pp. 428–437 (2018), https://doi.org/10.1007/978-3-319-94144-8_26
7. Ignatiev, A., Morgado, A., Marques-Silva, J.: RC2: An efficient maxsat solver. Journal on Satisfiability, Boolean Modeling and Computation **11**, 53–64 (09 2019). https://doi.org/10.3233/SAT190116
8. Leyton-Brown, K., Hoos, H.H., Hutter, F., Xu, L.: Understanding the empirical hardness of np-complete problems. Commun. ACM **57**(5), 98–107 (may 2014). https://doi.org/10.1145/2594413.2594424, https://doi.org/10.1145/2594413.2594424
9. Mitchell, D., Selman, B., Levesque, H.: Hard and easy distributions of sat problems. In: Proceedings of the Tenth National Conference on Artificial Intelligence. p. 459–465. AAAI'92, AAAI Press (1992)
10. Monasson, R., Zecchina, R., Kirkpatrick, S., Selman, B., Troyansky, L.: Determining computational complexity from characteristic 'phase transitions'. Nature **400**(6740), 133–137 (jul 1999). https://doi.org/10.1038/22055
11. Zhang, W.: Phase transitions and backbones of 3-sat and maximum 3-sat. In: Walsh, T. (ed.) Principles and Practice of Constraint Programming — CP 2001. pp. 153–167. Springer Berlin Heidelberg, Berlin, Heidelberg (2001)