BEHAVIOR OF RANDOM K-CNF FORMULAS

Peh Jun Siang A0201838H, Sunaga Shion A0210635R

April 20, 2023

Abstract

This paper studies the behavior of random CNF formulas with k-clauses, where k $\in$ [3, 5] with clause density r. Such formulas are commonly used as benchmarks to evaluate the performance of algorithms for solving the Boolean satisfiability problem (SAT). We focus on analyzing the density of the formula, the number of satisfying assignments, and the running time of algorithms for solving the formula. We also investigate the threshold and phase transition phenomena, which refer to the critical ratio of clauses to variables above which the formula is almost surely satisfiable and below which it is almost surely unsatisfiable, and the sudden change in the behavior of the formula as the ratio of clauses to variables crosses the threshold value. Our findings shed light on the fundamental properties of random CNF formulas with k-clauses and contribute to our understanding of the behavior of algorithms for solving SAT.

1 Introduction

The Boolean satisfiability problem (SAT) is a fundamental computational problem with a wide range of applications, including automated reasoning, circuit design, and software verification. SAT is NP-complete, making it computationally intractable in the worst case. To assess the performance of algorithms for solving SAT, researchers commonly use random conjunctive normal form (CNF) formulas as benchmarks. These formulas are generated by selecting variables and clauses uniformly at random. The clauses are conjunctions of literals, and the formula is satisfiable if there exists an assignment of truth values to the variables that satisfies all the clauses.

In this paper, we investigate the behavior of random CNF formulas with k-clauses, where k ranges from 3 to 5. Such formulas have gained significant attention in the literature as they exhibit essential properties related to the study of SAT. For instance, random CNF formulas with k=3 display a sharp threshold phenomenon, meaning there exists a critical ratio of clauses to variables above which the formula is almost certainly satisfiable and below which it is almost certainly unsatisfiable. This threshold value has been studied extensively and has significant implications for the design of algorithms for solving SAT.

Our analysis focuses on the density of the formula, the number of satisfying assignments, and the threshold and phase transition phenomena. By doing so, we aim to contribute to our understanding of the fundamental properties of random CNF formulas with k-clauses and their implications for the design of algorithms for solving SAT.

2 Related Work

The study of the satisfiability of random k-SAT is still in active research particularly in k $\geq$ 5 and k-CNF and XOR Phase Transitions [DMV17]. More recently, there has been growing interest in understanding the phase transition phenomenon in higher-order SAT problems, such as 5-SAT and 6-SAT. These problems are known to be significantly harder than 3-SAT and 4-SAT, and little is currently known about the structure of their solution spaces or the factors that influence their computational complexity.

2.1 Phase Transitions

A number of studies have explored the use of heuristics and metaheuristics for solving SAT problems, including genetic algorithms, simulated annealing, and tabu search. These approaches have been shown to be effective for solving some classes of SAT problems, but can be highly dependent on the specific problem instance and the choice of algorithm parameters.

There appears to be a critical limiting value of r which we shall denote as c_k such that

$$\lim_{n\to\infty} Prob(sat, F_k(n, rn)) = \begin{cases} 0, & \text{for } r > c_k \\ 1, & \text{for } r < c_k \end{cases} \tag{1}$$

It is easy to show that $c_0 = c_1 = 0$. It has been shown theoretically that $c_2 = 1$ and $3.003 < c_3 < 4.81$ [GW94]. More recently, research has also explored the impact of phase transitions on the performance of SAT solvers and the design of algorithms for solving SAT. For example, in a 2018 paper, M. Wahlström showed that for satisfiable instances, certain SAT algorithms have polynomial-time complexity in the satisfiable regime, but exhibit an exponential running time in the unsatisfiable regime. [LW18] This highlights the importance of understanding the phase transition phenomenon in order to design efficient algorithms for solving SAT.

2.2 Regular Random k-SAT: Properties of Balanced Formulas

O. Dubois, Y. Boufkhad, and J. Mandler were among the first researchers to provide strong evidence for the phase transition phenomenon in random 3-SAT formulas. In their seminal paper, published in 1997, they showed that there exists a critical clause-to-variable ratio, known as the satisfiability threshold, above which random 3-SAT formulas are almost certainly unsatisfiable and below which they are almost certainly satisfiable.

Using extensive computational simulations, Dubois et al. estimated the satisfiability threshold for random 3-SAT formulas to be around 4.25, with a margin of error of approximately 0.1. This value has since been refined and confirmed by numerous subsequent studies, providing strong evidence for the existence of the phase transition phenomenon in random 3-SAT formulas.

The phase transition phenomenon in random 3-SAT formulas has important implications for the analysis of algorithms for solving SAT. Above the satisfiability threshold, the running time of SAT algorithms is expected to grow exponentially with the number of variables in the formula, while the threshold, efficient algorithms can be designed to solve the problem in polynomial time with high probability. [DBM02]

3 Experimental Setup

We ran experiments with k $\in$ [3,5] and different values of n. For each k-clause we uniformly selected k out of the variables $X_1, , X_n$ without replacement. For each selected variable X_i, we include exactly one of the literals X_i or $\neg X_i$ in the k-clause, each with probability 0.5. The disjunction of these k literals is a uniformly chosen k-clause. For each assignment of k, n and r, we evaluated satisfiability, using CryptoMiniSAT [SNC09] by constructing the conjunction of $\lceil rn \rceil$ k clauses.

For model counting, we add solutions obtained from CryptoMiniSAT to the set of clauses by taking the negation of the results obtained and run the solver until the formula becomes unsatisfiable.

4 Experimental Results

For each k, we plotted the probability of satisfiability $P_k(n, nr)$ against r for varying values of n at regular intervals with r-interval of 0.2. Each plot shows 2 coordinates which denotes the start and the end of the transition phase. We experimentally determine the critical limiting value c_k by averaging the r value at p=0.5 for differents values of n. We also compare the time complexity for increasing values of n and k.

4.1 3-CNF

We conducted 6 experiments for 3-cnf with n $\in$ {25,50,75,100,125,150}. The experiment results for 50,75,100,150 are shown below.

Based on our experiments, we can make several observations regarding the relationship between the satisfiability threshold and the size of the formula.

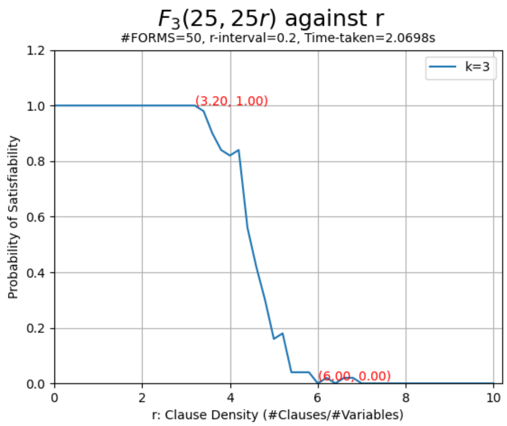

Figure 1: n=25, Time taken=2.06s, width=2.8

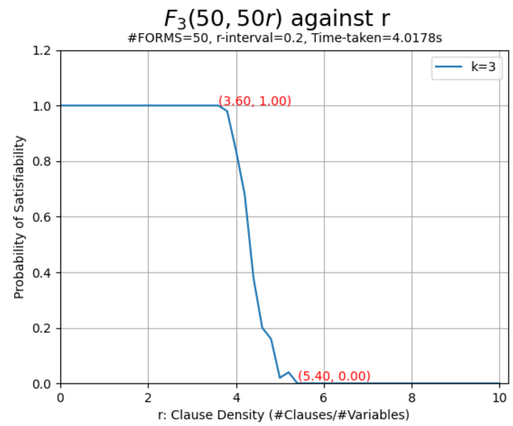

Figure 2: n=50, Time taken=4.01s, width=1.8

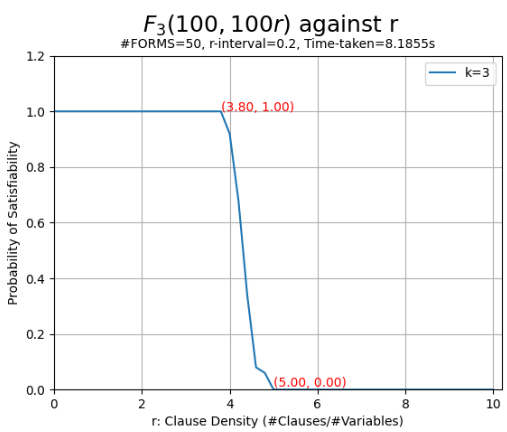

Figure 3: n=100, Time taken=8.18s, width=1.2

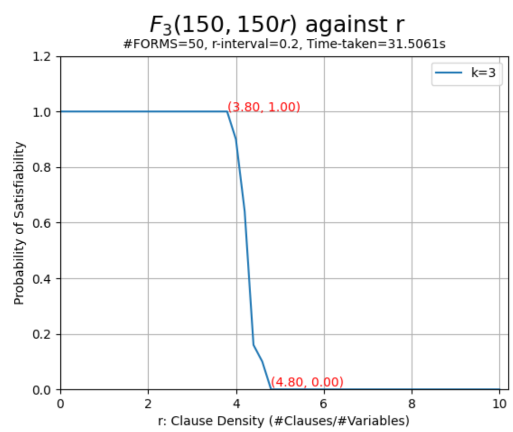

Figure 4: n=150, Time taken=31.5s, width=1

First, for values of n between 25 and 150, we observe that the formula is always satisfiable for r values below approximately 3.80. This is consistent with the satisfiability threshold of 4.25 observed by Dubois et al., providing additional evidence for the existence of a phase transition in random 3-SAT formulas. Taking the average r value at p=0.5, our threshold value c_3 is around 4.3 which is around the approximate value cited by the paper.

Second, we observe that as the value of n increases from 25 to 150 for k=3, the width of the satisfiable region (i.e., the range of r values for which the formula is satisfiable) decreases at a decreasing rate. Specifically, we see that the width of the band decreases from approximately 2.8 when n=25 (as shown in 1) to 1 when n=150 (as shown in 4).

This trend suggests that as the size of the formula increases, the range of r values for which the formula is satisfiable becomes narrower, making it more difficult to find a satisfying assignment. This is consistent with the intuition that larger formulas are generally more difficult to solve than smaller ones, and highlights the need for more efficient algorithms and heuristics to solve large random k-SAT instances.

4.2 3-SAT Solve time vs n

There is a linear relationship between the solve time and n for which n $\leq$ 100. From n $>$ 100, there appears to be an exponential increase in solve time. For n up to 100, the solve time typically grows at a roughly constant rate with n. This linear relationship between solve time and n can be explained by the fact that for small formulas, the search space of possible assignments is relatively small, and a solver can quickly explore all possible assignments to determine whether the formula is satisfiable.

However, as n increases beyond a certain threshold (around 100), the search space grows exponen-

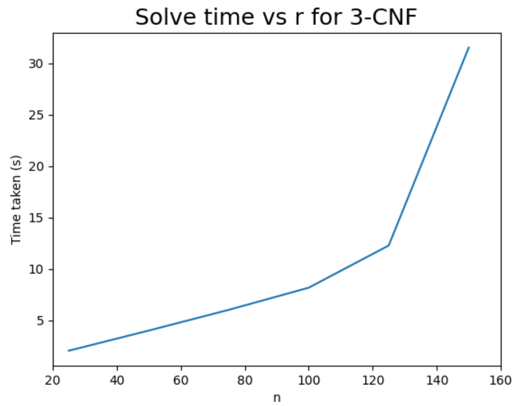

Figure 5: Solve time vs n for 3 SAT.

tially, and the time required to search the entire space quickly becomes prohibitive. As a result, SAT solvers often employ sophisticated heuristics and pruning techniques to explore the search space more efficiently, but even with these techniques, the solve time can still grow exponentially with n.

4.3 4-SAT

The satisfiability of random 4-SAT formulas is an active area of research in theoretical computer science and mathematics and is less well studied compared to 3-SAT. Similar to the case of 3-SAT, random 4-SAT formulas exhibit a phase transition phenomenon.

We conducted 6 experiments for 4-SAT with n $\in$ {25,35,40,45,50,75}. The experiment results for 25,40,50,75 are shown below. Taking the average r value at p=0.5, the threshold value c_4 appears to be around 10 which is consistent with the approximate value of 9.76 reported by Ian PGent. [GW94].

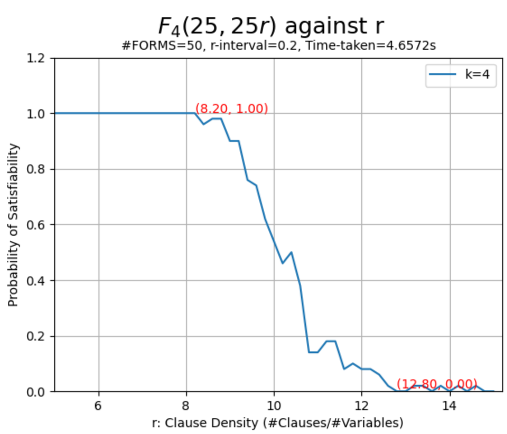

Figure 6: n=25, Time taken=4.65s, width=4.6

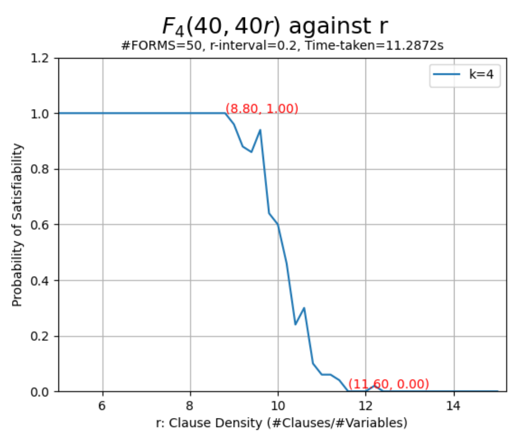

Figure 7: n=40, Time taken=11.3s, width=2.8

From n=25 to n=75, the width of the transition phase decrease similar to 3-SAT from 4.6 when n=25 in Figure 6 to 2.4 when n=75 in Figure 9. The width of the transition phase for 4-SAT appears to be wider than 3-SAT. For n=25, the width of 3-SAT is 2.8 compared to 4.6 in 4-SAT and for n=50, the width of 3-SAT is 1.8 compared to 3.0 for 4-SAT.

The wider transition phase in random 4-SAT formulas compared to 3-SAT could be attributed to the increased complexity of the formula and variance in the satisfiability as k increases due to the inclusion of larger clauses. While 4-SAT has more variables in each clause compared to 3-SAT, which could potentially make it easier to satisfy, the increased complexity of the formula due to the inclusion of larger clauses may result in more variance in the satisfiability of the formula. Another possible reason could be that the algorithmic techniques used to solve 3-SAT may not be as effective in solving

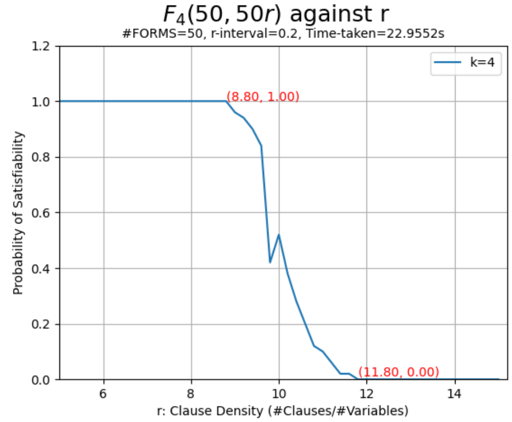

Figure 8: n=50, Time taken=22.95s, width=3.0

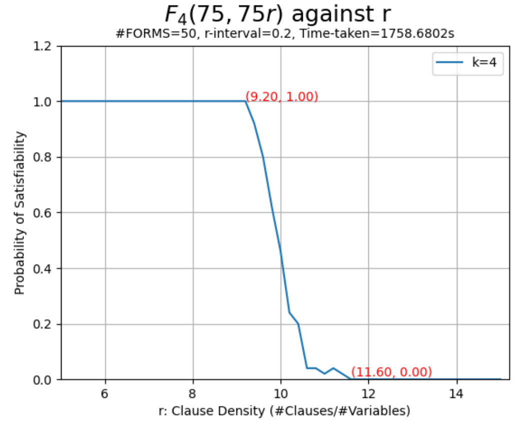

Figure 9: n=75, Time taken=1758.6s, width=2.4

4-SAT, leading to wider transition phases. Further research is needed to investigate these possibilities and fully understand the underlying reasons.

4.4 4-SAT Solve time vs n for 4-SAT

There is a linear relationship between the solve time and n for which n $\leq$ 35 as shown in figure 10. From n $>$ 35, there appears to be an exponential increase in solve time. For n $>$ 50, there is an explosion in time complexity as shown in figure 11 with n=70 taking 76.6 times longer than n=50 from 22.95s to 1758s.

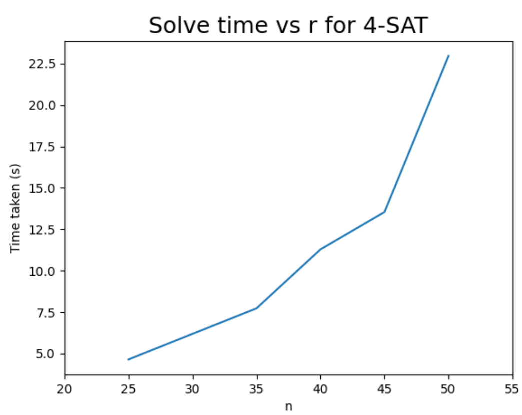

Figure 10: Time taken n={25,35,40,45,50}

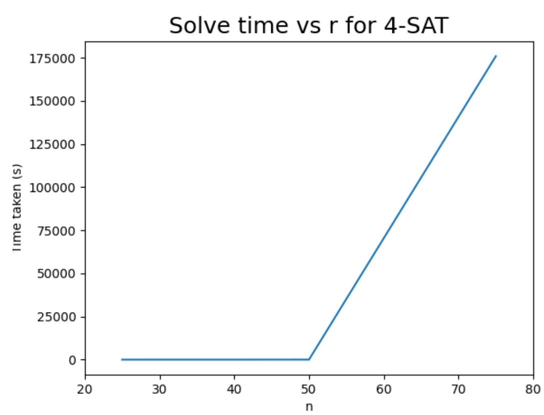

Figure 11: Time taken n={25,35,40,45,50,75}

For n $\leq$ 35, the solve time for 4-SAT is roughly double that of 3-SAT and grows at a roughly constant rate with n. The reason why the solve time for 4-SAT is roughly double that of 3-SAT for small n is likely due to the increased complexity of 4-SAT compared to 3-SAT. In 4-SAT, each clause involves four variables, while in 3-SAT, each clause involves only three variables. This means that there are more possible combinations of variable assignments that need to be checked in 4-SAT, which leads to a longer solve time. However, for larger values of n, the exponential growth of the search space in both 3-SAT and 4-SAT causes the solve time to increase dramatically for both, with 4-SAT generally exhibiting a greater increase due to the higher complexity of the problem.

4.5 5-SAT

We could only conduct experiments up to n=40 as the time taken for checking the satisfiability of 5-SAT is much longer than 3-SAT and 4-SAT due to an exponential increase in number of satisfying assignments.

We conducted 4 experiments for 5-SAT with n $\in$ {25,30,35,40}. The experiment results are shown below. The threshold value c_5 appears to be around 22.

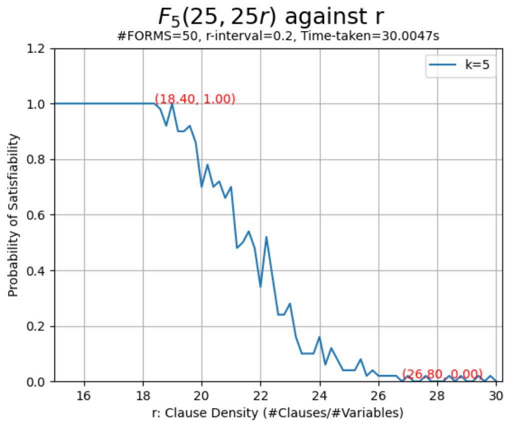

Figure 12: n=25, Time taken=30s, width=8.4

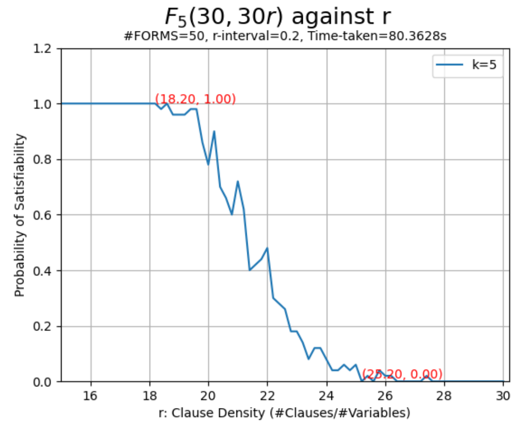

Figure 13: n=30, Time taken=80.4s, width=7.0

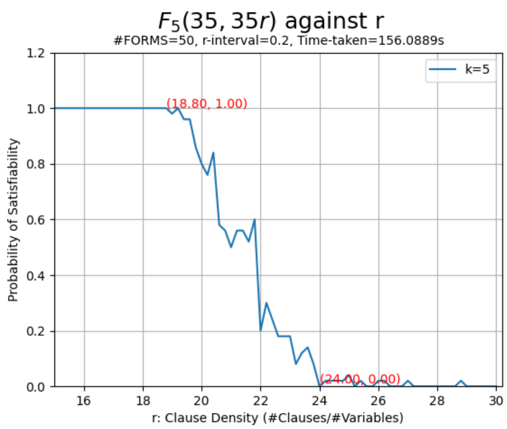

Figure 14: n=35, Time taken=156.1s, width=5.2

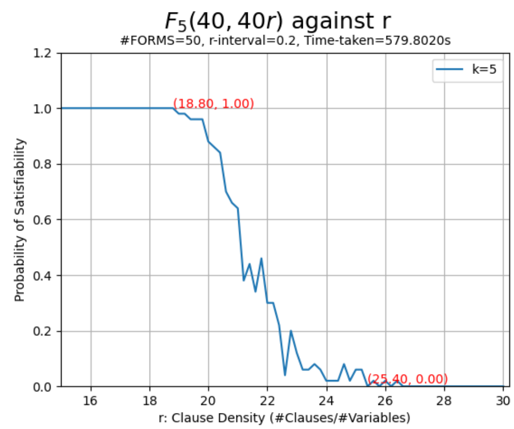

Figure 15: n=40, Time taken=579.8s, width=5.6

Similar to 3-SAT and 4-SAT, there appears to be a decreasing trend between n and the width of the satisfiable region. However, for n=40, the width is 5.6 which is slightly larger than 5.2 for n=35. This might be because the random formula generated for n=40 happened to be satisfiable by chance. The width of the satisfiable region appears to increase by approximately 2 times when k increases by one from 3-5 for n=25 as shown in Table 1. The increase in the width of the satisfiability region as k increases may be because as the number of variables and clauses increases, the search space becomes larger and more complex. This means that higher k values require a greater number of variables and clauses to achieve the same level of complexity as lower k values.

k	width
3	2.8
4	4.6
5	8.4

Table 1: Width of satisfiable region for n=25.

4.6 5-SAT Solve time vs n

There is a linear relationship between the solve time and n for which n $\leq$ 35. From n > 35, there appears to be an exponential increase in solve time. Beyond n=40, the time complexity increases drastically requires specialised algorithms to find the threshold value c_k.

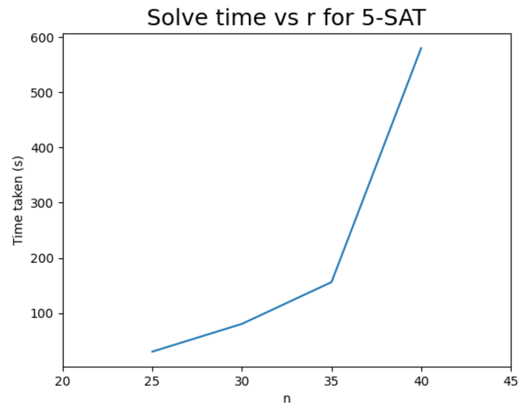

Figure 16: 5-SAT Solve time vs n.

4.7 Evaluation

Time complexity increases as k increases from 3 to 5 as shown in our results. The comparison for the time complexity is shown in Figure 17 below. For k > 3, we see an exponential increase in time complexity for n > 40. The location of the threshold value c_k also appears to increase as k increases

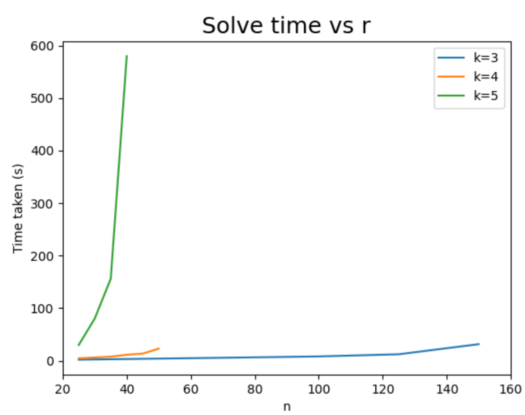

Figure 17: Comparison of time complexity.

from c_3=4.3 when k=3, c_4=10 when k=4 to c_5=22 when k=5. As n increases, the width of the satisfiablitly region decreases in general and the volatility of the satisfying probability also appears to decreases as the graph becomes smoother. This is because the variance due to the randomness decreases as the number of clauses and the number of variables increases in the formula as n increases.

5 Number of Modeling Assignments

The number of assignments that models the formula decreases in a decreasing rate as r increases. As n increases, there is an exponential increase in #models. We experimented with n=10 and n=13. Beyond n=13, the size of #models gets too large to compute. As k increases, there are more assignments that satisfies the formula as shown in Figure 18 and Figure 19. This is because as the clause gets larger,

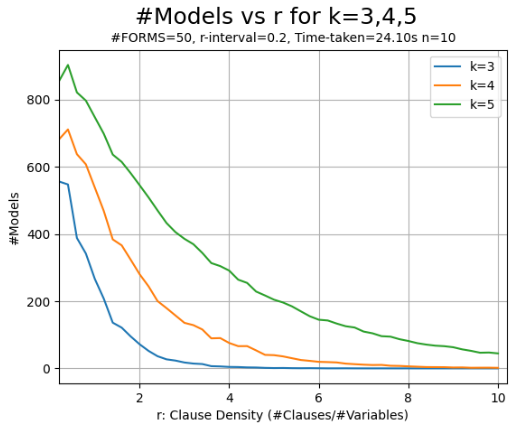

Figure 18: Comparison of #Models for n=10

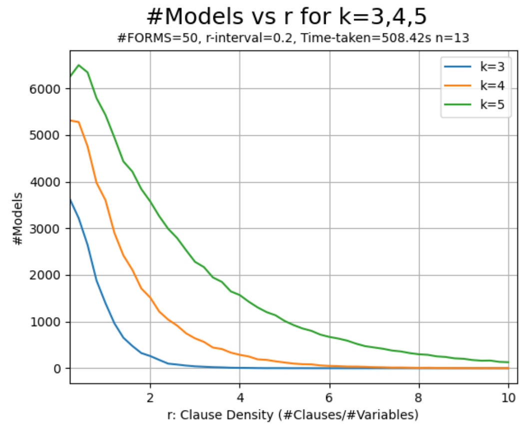

Figure 19: Comparison of #Models for n=13

there are more literals that can satisfy each clause. Figure 19 also reflects our results in Section 3 where we experimentally showed the threshold value c_k. For k=3, the #models is 0 for r $>$ 4.3 while there are still some assignments at r=10 for k=5 as c_5 is approximately 22 from section 4.5.

6 Replicating Our Results

We have included 2 programs **main.py** and **count_assignments.py** which plots the probability $P_k(n, nr)$ against r and #Models againsts r respectively. Each program can take in variable number of n, k-start, k-end, number of formulas, r-start, r-end and r interval which allows for different configurations.

7 Conclusion and Future work

In conclusion, our study provides some evidence for the existence of a phase transition phenomenon in random k-SAT formulas for k=3, 4, and 5. Our findings suggest that the width of the satisfiable region decreases as n increases and increases as k increases. We also observe that the solve time for 4-SAT is roughly double that of 3-SAT for n $\leq$ 35 and grows at a roughly constant rate with n.

Future work in this area could focus on exploring the phase transition phenomenon in larger k-SAT formulas and investigating the relationship between the width of the satisfiable region and the solve time for SAT algorithms. Additionally, it would be interesting to investigate the phase transition phenomenon in other types of random constraint satisfaction problems and to explore the potential applications of these findings in the design and analysis of algorithms for solving combinatorial optimization problems.

References

[GW94] Ian P Gent and Toby Walsh. “The SAT phase transition”. In: *ECAI*. Vol. 94. PITMAN. 1994, pp. 105–109.

[DBM02] Olivier Dubois, Yacine Boufkhad, and Jacques Mandler. “Typical random 3-SAT formulae and the satisfiability threshold”. In: *arXiv preprint cs/0211036* (2002).

[SNC09] Mate Soos, Karsten Nohl, and Claude Castelluccia. “Extending SAT Solvers to Cryptographic Problems”. In: *Theory and Applications of Satisfiability Testing - SAT 2009, 12th International Conference, SAT 2009, Swansea, UK, June 30 - July 3, 2009. Proceedings.* Ed. by Oliver Kullmann. Vol. 5584. Lecture Notes in Computer Science. Springer, 2009, pp. 244–257. DOI: 10.1007/978-3-642-02777-2_24. URL: https://doi.org/10.1007/978-3-642-02777-2%5C_24.

[DMV17] Jeffrey M. Dudek, Kuldeep S. Meel, and Moshe Y. Vardi. *Combining the k-CNF and XOR Phase-Transitions*. 2017. arXiv: 1702.08392 [cs.DM].

[LW18] Victor Lagerkvist and Magnus Wahlström. *Which NP-Hard SAT and CSP Problems Admit Exponentially Improved Algorithms?* 2018. arXiv: 1801.09488 [cs.DS].