

A Review Study of Sat Algorithms: Complete and Incomplete

Devyani Mehul Vasistha^{1*} Dr. Manish Kumar²

¹ Research Scholar, Department of Computer Science, Arunachal University of Studies, Namsai

² Associate Professor, Department of Computer Science, Arunachal University of Studies, Namsai

Abstract – In Computer Science, the Boolean Satisfiability Problem (SAT) is the problem of deciding whether there exists a translation that fulfills a given Boolean equation. SAT is one of the principal problems that were demonstrated to be NP-complete, which is additionally central to artificial intelligence, algorithm and hardware design. This paper surveys the primary algorithm called as Serial Sat Algorithms which can complete or incomplete.

Keywords: Planning Systems, AI, SAT etc.

-----X-----

1. INTRODUCTION

At present, software applications have penetrated every corner of developed human society. Meanwhile, software vulnerabilities have also brought security risks. According to the National Institute of Standards and Technology (NIST) on the software defect investigation[1], in the United States, the losses caused by software defects are as high as 59.5 billion US software security. Automated execution of software verification usually applies the SAT dollars, which is equivalent to 0.6% of US GDP. Software verification is an effective way to ensure Solver, which can convert software verification problems into Boolean Satisfiability Problem. The SAT solving technology is key to promote the performance of software verification. The efficiency of software verification requires verification technology to find defects with speed and accuracy. A wide range of defect types and the low false-rate of all kinds of defects are also required to ensure an optimized software verification. Also, the locations and causes of the bugs need to be determined quickly. These requirements must be based on the execution of the algorithm. The key of the typical software verification algorithm is to solve the Satisfiability problem, which needs a large amount of irregular calculation. When the size of the constraint clause is over 10 million, it must rely on a high-performance computing platform, and the problem can be solved within an acceptable amount of time. Satisfiability Problems consist of a set of Boolean variables and a set of short clauses composed of these variables. The problem is to obtain a set of solutions satisfying the set of short clauses. SAT problem is the basic problems solving of

mathematical logic, reasoning, machine learning, and many other theoretical and practical problems. The SAT problem is the first discovered NP-complete problem, and is also the core of a large class of NP-complete problems. Therefore, solving the SAT problem plays an important role in the study of artificial intelligence systems and computational theory. Any improvement of the SAT algorithm has a great effect on Computer Science.

2. SERIAL SAT ALGORITHMS

There are two kinds of mainstream serial SAT algorithms. They are complete algorithms (ie, backtracking search algorithms) and incomplete algorithms. The complete algorithms search for the entire solution space. If solutions exist, they will be returned. If not, the problem can be proven to be unsatisfiable. The incomplete algorithms cannot prove anything because of their randomness. These two kinds of algorithms are different, and jointly promote the development of efficient SAT algorithm.

2.1 Complete Algorithms:

The complete algorithm searches for the solution space of a given SAT problem based on enumeration and backtracking search mechanisms. If the problem is satisfiable, then the solution of the logic formula will be given, and if not, a proof of completeness will be given. In fact, in the second case, it is extremely important and practical to give proof of completeness since many unsatisfiable instances require proof that they cannot be satisfied. The complete algorithm

determines the value of variables in the propositional logic one by one until all variables are assigned and the value of the formula is true, or all possible assignments are tried but remain unsatisfied. The advantage of the complete algorithm is that it can guarantee the solution of the corresponding SAT problem or prove that the formula is unsatisfiable. Unfortunately, the efficiency of the complete algorithm is extremely low. Although the average time complexity is a polynomial level, the worst-case time complexity is exponential. In fact, the complete algorithm is a process of depth-first search of the entire solution space, the search space is very large, which may make the computer unable to return the results in an acceptable time. Because of the combinatorial explosion problem, the calculation time spent cannot be tolerated. As early as the 1960s, Davis, Putname, Longemann, Loveland, and others have studied the SAT problem, and their basic algorithm is called DPLL algorithm[2], which is one of the most commonly used algorithms. And its ideas have, so far, been closely followed. In 1995, GRASP[3] was proposed, which summarized the basic DPLL algorithm flow, and the SAT algorithm was gradually used more frequently to solve practical problems. The most important contribution of GRASP was the introduction of the Conflict Driven Clause Learning (CDCL) learning process, which is very effective in narrowing the search space with the DPLL algorithm, which makes the SAT algorithm a quantum leap. Subsequent development of the technology can be summarized in the CDCL, including the clause-based learning of the Statute, a random restart, heuristic decision-making based on the active value, and effective reasoning to support the data structure. With the introduction of these new technologies, a number of more efficient SAT algorithms such as MiniSat[4], Chaff[5], BerkMin[6], CryptoMiniSAT[7], PicoSAT[8], and Lingeling[9] were introduced.

2.2 Incomplete Algorithms:

The local search algorithm is the most ordinarily utilized in the incomplete algorithm. This technique utilizes a specific procedure to search randomly in the assignment space of the variable, and step by step moves toward the last arrangement under the direction of the objective function. There are two kinds of objective functions: one is to limit the quantity of clauses with the genuine estimation of bogus, and the other is to fulfill the most extreme number of imperative clauses. Local search algorithms chiefly incorporate greedy local search (GSAT[10]) and random walk GSAT (WSAT[11]). The two algorithms start randomly producing an underlying assignment, and afterward execute the circle. On the up and up, the GSAT algorithm over and again checks the present assignment's neighbors (eg, assignments that have just a single variable assignment diverse to the present assignment) and chooses another assignment to boost the quantity of clauses that are fulfilled. The WSAT algorithm randomly chooses a variable in an

unsatisfied provision and upsets its worth. TSAT[12] is likewise founded on the GSAT. This algorithm is equivalent to the first WalkSat in system. Two better approaches to choose variables (Novert, R Novert) are proposed to abstain from falling into local minima. These two techniques have demonstrated to be viable. Sparrow[14] presents the likelihood circulation function in the determination of variables, every variable being chosen with some likelihood. ProbSAT [15] is additionally reliant on the likelihood circulation of the solver, however the acknowledgment is basic, there is no mind boggling heuristic technique, yet is successful. Another approach to take care of SAT issues is to change the satisfiability issue into another issue in another field, which can be explained rapidly, for example, Genetic Algorithm (GA)[16], Simulated Annealing(SA)[17], and Survey Propagation(SP)[18] in measurable material science, etc. The SP algorithm has been extremely successful in the investigation of the 3-SAT issue. The possibility of the SP algorithm strategy is gotten from the turn glass rule in factual material science and has significant essentialness in tackling the SAT issue. It builds preparing power from 104 to 106 sets of greatness in variable size.

3. CONCLUSION

This research paper is basically concerned with artificial intelligence based issues like Serial Sat Algorithms. Here we have also discussed is two important Complete Algorithms, incomplete algorithms. It will be an important paper for the researcher looking in the field of artificial planning.

4. REFERENCES

1. G. Tassej (2002). National Institute of Standards and Technology, RTI Project 7007.
2. M. Davis, G. Logemann, and D. Loveland (1962). Communications of the ACM 5, pp. 394–397.
3. J. P. M. Silva and K. A. Sakallah (1997). "Graspla new search algorithm for satisfiability," in Proceedings of the 1996 IEEE/ACM international conference on Computer-aided design (IEEE Computer Society), pp. 220–227.
4. N. Een and N. Sorensson (2004). Springer Berlin /Heidelberg 10, pp. 978–3.
5. M. W. Moskewicz, C. F. Madigan, Y. Zhao, L. Zhang, and S. Malik (2001). "Chaff: Engineering an efficient sat solver," in Proceedings of the 38th annual Design Automation Conference (ACM, 2001), pp. 530–535.

6. E. Goldberg and Y. Novikov (2007). *Discrete Applied Mathematics* 155, pp. 1549–1561.
7. K. Wu, T. Wang, X. Zhao, and H. Liu (2011). "Cryptominisat solver based algebraic side-channel attack on present," in *International Conference on Instrumentation*, pp. 561–565.
8. A. Biere, *Journal on Satisfiability* (2008). *Boolean Modeling and Computation* 4, pp. 75–97.
9. A. Biere (1992). *Proceedings of SAT Challenge* 33–34 (2012). [10] B. Selman, H. J. Levesque, D. G. Mitchell, et al., "A new method for solving hard satisfiability problems." in *AAAI*, Vol. 92, pp. 440–446.
10. B. Mazure, L. Sais, and E. Gregoire (1997). "Tabu search for sat," in *AAAI/IAAI*, pp. 281–285.
11. D. Mcallester, B. Selman, and H. Kautz (2003). "Evidence for invariants in local search," in *PROCEEDINGS OF AAAI-97*, pp. 321–326.
12. A. Balint and A. Frohlich (2010). "Improving stochastic local search for sat with a new probability distribution," in *International Conference on Theory and Applications of Satisfiability Testing* (Springer, 2010), pp. 10–15.
13. A. Balint and U. Schoning (2012). "Choosing probability distributions for stochastic local search and the role of make versus break," in *International Conference on Theory and Applications of Satisfiability Testing* (Springer, 2012), pp. 16–29.
14. C. Solnon (2002). *IEEE transactions on evolutionary computation* 6, pp. 347–357.
15. W. M. Spears (1993). *Cliques, Coloring and Satisfiability: Second DIMACS Implementation Challenge* 26, p. 33.
16. A. Braunstein, M. Mezard, and R. Zecchina (2005). *Random Structures & Algorithms* 27, pp. 201–226.
17. Y. Hamadi, S. Jabbour, and L. Sais (2008). *Journal on Satisfiability, Boolean Modeling and Computation* 6, pp. 245–262.
18. H. Zhang, M. P. Bonacina, and J. Hsiang (1996). *Journal of Symbolic Computation* 21, pp. 543–560.

Corresponding Author

Devyani Mehul Vasistha*

Research Scholar, Department of Computer Science, Arunachal University of Studies, Namsai