Comments on Evaluating the Claims of "SAT Requires Exhaustive Search"

Junping Zhou
School of Information Science and Technology, Northeast Normal University, China

Jiaxin Liang
School of Information Science and Technology, Northeast Normal University, China

Minghao Yin
School of Information Science and Technology, Northeast Normal University, China
Key Laboratory of Applied Statistics of MOE, Northeast Normal University, China

Abstract

In this paper, we examine the issues raised by Chavrimootoo et al. in their paper "Evaluating the Claims of "SAT Requires Exhaustive Search" [1]. They point out that there is a flaw in Xu and Zhou's "SAT Requires Exhaustive Search" [2], which causes Xu and Zhou's claim that P $\neq$ NP does not follow from their arguments. However, we find some misunderstandings in the expositions presented by Chavrimootoo et al. [1], which led to their one-sided analyses and conclusions.

1 Introduction

The P versus NP problem is a major unsolved problem in theoretical computer science. Xu and Zhou in [2] prove a class of random CSP instances of model RB cannot be solved without an exhaustive search, which presents a lower bound on the complexity of RB and implies P $\neq$ NP. Chavrimootoo et al. identify that some arguments of Xu and Zhou in [2] are incorrect and have no firm evidence to support their main results. However, we find some misunderstandings in the expositions presented by Chavrimootoo et al., which led to their one-sided analyses and conclusions. First, the analysis of the difficulty of subproblems of a Model RB instance, which they discussed, is beside the proof of the main results in [2]. Second, the 3-SAT problems argued in [1] are not identical to and quite different from the SAT problems in [2].

In Section 2, we outline some basic definitions used in the paper and the main results in [2]. In Section 3, we analyze the arguments that Chavrimootoo et al. misunderstand. Finally, a conclusion is presented.

2 Preliminaries

In this section, we give some necessary preliminaries.

▶ **Definition 1 (Constraint Satisfaction Problem (CSP) Instance).** *A constraint satisfaction problem (CSP) instance is defined as a triple* (X, D, C)*.* $X = (x_1, x_2, ..., x_n)$ *is a set of* n *variables.* D *is a mapping from* X *to a set of domains* $D = (D_{(x1)}, D_{(x2)}, ..., D_{(xn)})$*, where* $D_{(xi)}$*,* D_i *for short, is the finite domain of its possible values. A constraint* $C_{i1,i2,...,ik} \in C$ $(2 \leq k \leq n)$ *is defined as a pair* (X_i, R_i) *such that* $X_i = (x_{i_1}, ..., x_{i_k})$ *is a subset of* X *called the constraint scope and* R_i*, as called constraint relations, is a subset of the Cartesian product* $D_{i_1} \times ... \times D_{i_k}$ *and it specifies the allowed combinations of values for the variables in* C_i*.*

▶ **Definition 2 (Constraint Satisfaction Problem, CSP).** *The constraint satisfaction problem (CSP) for a CSP instance consists in deciding whether there exists an assignment satisfying all the constraints.*

▶ **Definition 3 (Model RB [2]).** *A class of random CSP instances of model RB is denoted by a tuple* (k, n, α, r, p)*, where for each instance: 1)* $k \geq 2$ *denotes the arity of each constraint, 2)* $n \geq 2$ *denotes the number of variables, 3)* $\alpha > 0$ *determines the domain size* $d = n^{\alpha}$ *of each domain, 4)* $r > 0$ *determines the number* $m = rn \ln d$ *of constraints, 5)* $0 < p < 1$ *determines the number* $t = (1-p)d^k$ *of permitted tuples of each relation.*

▶ **Theorem 1.** *Model RB cannot be solved in time* $O(d^{cn})$ *time for every constant* $0 < c < 1$ *[2].*

▶ **Lemma 1.** *If a CSP problem with* n *variables and domain size* d *can be solved in* $T(n) = O(d^{cn})$ *time* ($0 < c < 1$ *is a constant*)*, then at most* $O(d^c)$ *subproblems with* $n-1$ *variables are needed to solve the original problem* [2].

▶ **Corollary 1.** *SAT with* N *Boolean variables cannot be solved in* $O(2^{cN})$ *time for any constant* $0 < c < 1$ *[2].*

3 Comment 1: the analysis of subproblem difficulty is beside the proof of Theorem 1.

The authors in [1] believe that the proof of Theorem 1 lacks the argumentation about the difficulty of each subproblem, i.e., each subproblem is also an instance of Model RB, requiring an exhaustive search on the order of $O(d^n)$. As a matter of fact, the analysis of subproblem difficulty is beside the proof of Theorem 1. That's because the proof of Theorem 1 in [2] is different from the previous SAT and CSP algorithm analysis methods. The usual methods for SAT and CSP compute the worst-case upper bounds according to a given divide-and-conquer algorithm [3, 4, 5]. Specifically, the execution of a divide-and-conquer algorithm can be considered as a branching tree, that is, a tree such that formulas labeling children are obtained by assigning values to arbitrary variables in formulas labeling parents. One can compute the upper bound of an algorithm by estimating the number of leaves in the branch tree. In [2], Xu and Zhou's purported proof of Theorem 1 is based on the assumption that there exists some constant $0 < c < 1$ such that a RB instance can be solved in $O(d^{cn})$ time. Then the instance can be divided into subinstances based on Lemma 1, which is applicable to all general CSP instances. They prove the difficulty of Model RB argued in Theorem 1 under a given upper bound of the divide-and-conquer algorithms, which are indeed the only type of exact algorithms available for solving CSP by far. In the process of proof, they use divide-and-conquer algorithm to assign values to an arbitrary variable of a CSP instance, which divides the original problem with n variables into subproblems with $n-1$ variables, with the aim of deducing the upper bound of the number of subproblems. In the whole proof process, the difficulty of each subproblem is not used. Thus, the analysis of subproblem difficulty is beside the proof of Theorem 1.

4 Comment 2: the SAT argued in Corollary 1 are not the 3-SAT problems.

Chavrimootoo et al. [1] mention that the claim in [2] contradicts known results that it is possible to solve 3-SAT problems without an exhaustive search, but still in exponential time. We find that the main results in [2] do not discuss the difficulty of the 3-SAT problems. The SAT problems argued in Corollary 1 are not the 3-SAT problems. As we know, CSP can be encoded into SAT by using the log-encoding, which can keep the characteristics of the original problems. In [2], by using the log-encoding, the CSP instances in Model RB are converted into SAT instances, which have no restriction on the clause length. These SAT instances are very long in clause length and must be solved by an exhaustive search.

5 Conclusion

This paper discusses the misunderstandings of the expositions held by Chavrimootoo et al. in [1]. First, the analysis of the difficulty of subproblems of a Model RB instance they discussed is beside the proof of main results in [2]. Second, the 3-SAT problems argued in [1] are not identical to and quite different from the SAT problems in [2].

References

1. Michael C. Chavrimootoo, Yumeng He, Matan Kotler-Berkowitz, Harry Liuson, and Zeyu Nie. Evaluating the Claims of "SAT Requires Exhaustive Search". arXiv:2312.02071 [cs.CC], December 4, 2023.
2. K. Xu and G. Zhou. SAT requires exhaustive search. arXiv:2302.09512 [cs.CC], September 21, 2023.
3. J. Zhou, M. Yin, C. Zhou. New Worst-Case Upper Bound for #2-SAT and #3-SAT with the Number of Clauses as the Parameter. AAAI, 2010.
4. E. Dantsin, E. Hirsch. Worst-Case Upper Bounds. Handbook of Satisfiability, 2021.
5. H. Chu, M. Xiao, Z. Zhang. An Improved Upper Bound for SAT. AAAI, 2021.