Review study regarding an Evidence of PVSNP Inequality.

Xiao-Hui Zhou

Department of Mechanical Engineering, UET Lahore, Pakistan

How to cite this article: Xiao-Hui Zhou (2024). On a Proof of Inequality of PVSNP. REVISTA EdeDC | Vol.05 issue.01, 25-26.

Accepted date: 15-12-2024
Publication date: 25-12-2024
DOI : https://doi.org/10.79425/rev.v5i1.25-26

ABSTRACT: This study is a new version of a previous paper. Its purpose is to simplify some sections of the old version and, above all, to present the proofs of some theorems which had been omitted for the sake of brevity. The analysis discussed in this study and its previous version is based on a well- known NP-complete problem which is called the "satisfiability problem" or "SAT". From SAT a new NP-complete problem, called "core function", derives; this problem is described by a Boolean function of the number of the clauses of SAT. In this study, a new proof is presented according to which the number of gates of the minimal implementation of core function increases with n exponentially. Since the synthesis of the core function is an NP-complete problem, this result can be considered as the proof of the theorem which states that the class P of all the decision problems which can be solved in polynomial time does not coincide with the class NP of the problems for which an answer can be verified in polynomial time.

Keywords: P-NP Question, Complexity, Boolean Functions, Satisfiability, Polynomial or Exponential Increase, Core Function

I. INTRODUCTION

A paper devoted to the proof of the theorem according to which P and NP do not coincide was presented to the Journal of Computer Science on September 2020 and published (Meo, 2021). According to the Journal of Computer Science at the end of August 2022 more than 2200 readers had viewed that paper and more than 600 readers had downloaded it.

Some readers have asked some questions concerning a few theorems whose proofs had been omitted in that paper for the sake of brevity. To prove these theorems is the main purpose of this new version of that paper.

The proof of inequality on the question PvsNP which had been presented in the previous paper and which will be completed in this study is based on the following steps:

1. A new Boolean function called "core function" is derived from the well-known SAT function. The core function is equivalent to SAT according to the known definition of NP-completeness
2. The main properties of the core function are presented and discussed
3. It is shown that the number of gates necessary to implement core function increases exponentially with the size of the problem

At present, no reader of my papers has found any mistake in the three steps of that proof. Future work might concern some mistakes which will be discovered. For example, if it will be proved that core function is not NP-complete, another function will be presented and discussed.

The second line of future research might concern the direct synthesis of SAT function or some other function equivalent to SAT.

A pure mark can be considered equal to NMT1(n) while the value of an impure mark can be considered equal to NMT1(n)$\cdot 2^{-m}$, where m is the number of spurious or complemented compatibilities. Besides, the value of a Boolean function which is equal to a sum of marks is always less than or equal to the sum of the values of the considered marks.

For example, as we shall discuss in the following sum of remainders of CF (4):

II. MATERIALS AND METHODS

The Core Function

The Boolean function implemented by the core layer will be called the "core function" of order t, where t is the number of triplets. It will be denoted with the symbol CF(t) or CF(n). The core layer processes only the $9\cdot t\cdot(t-1)/2$ compatibility variables c(i, j; h, k) and produce the global result of the computation. The core function can be determined by proceeding as follows.

Consider one selection of variables appearing in Eq. (1), one and only one for each triplet, for all the triplets:

by Eq. (1) will be called Ct or Cn. Indeed**,** the number t of triplets appearing in Eq. (1) plays the role of symbol *n* used in the standard complexity theory. In the following analysis, we shall use the symbol *t* when it is necessary to remember the number of triplets and *n* in the other cases.

 ISSN: 0034-9372

To simplify the analysis, circuit C_n will be decomposed into two processing layers called the "compatibility layer" and "core layer".

Compatibility Layer

A variable *j* of triplet *i* will be defined as “compatible” with variable *k* of triplet *h* when and only when, either:

- The sign s_{ij} of the former variable is equal to the sign s_{hk} of the latter variable, or
- The name <n_{ij1} n_{ij2} … n_{ijm}> of the former variable is different from the name <n_{hk1} n_{hk2} …n_{hkm}> of the latter variable

From that definition it follows that two “not compatible” variables have different signs and the same name; therefore, their AND is identically FALSE.
The compatibility layer is composed of $3 \cdot t \cdot (3 \cdot t-3)/2$ identical cells, one for each pair of variables belonging todifferent triplets.

Core Layer
The Boolean function implemented by the core layer will be called the “core function” of order *t*, where *t* is the number of triplets. It will be denoted with the symbol $CF(t)$ or $CF(n)$. The core layer processes only the $9 \cdot t \cdot (t-1)/2$ compatibility variables $c(i, j; h, k)$ and produce the global result of the computation. The core function can be determined by proceeding as follows.

Definition of Extended Prime Implicant
A term *T* of core function, that is, an implicant of core function (a product of literals implying core function), contains all the uncomplemented literals of a prime implicant. Therefore, it may be defined as an “extended prime implicant” (only) to remember that it contains all the compatibilities of a prime implicant.
It may be a spurious extended prime implicant or an impure extended prime implicant or both a spurious and impure extended prime implicant.
Notice that an extended prime implicant can be viewed as a (possibly spurious or impure) mark.

III. CONCLUSION

Since the number of minterms of $ECF(n)$ contained in $CF(n)$ is equal to $3^{n} \cdot NMT1(n)$ and the value of a gate, that is the number of new minterms produced by a gate, is less than:

$$valmax(n) = 9 \cdot (1 + (1/2) + (1/4))^{n-2} \cdot NMT1(n)$$

the number of gates necessary to implement $CF(n)$ is larger than $3^n/(9 \cdot ((1+1/2+1/4)^{(n-2)}))$ and, therefore, it increases exponentially with *n*.

Since the synthesis of core function $CF(n)$ is an NP-complete problem, this result is equivalent to proving that *P* and *NP* do not coincide.

Acknowledgment
The Academy of Science of Turin who published a preliminary version of this study.

Funding Information
The authors have not received any financial support or funding to report.

Ethics
This article is original and contains unpublished material. The corresponding author confirms that all of the other authors have read and approved the manuscript and no ethical issues involved.

REFERENCES

[1]. Cook, S. (1997) “The P versus NP problem,” in J. Carlson,

[2]. A. Jaffe, & A. Wiles (eds.), The Millennium PrizeProblem, pp. 88–104, Providence: American Mathematical Society. https://www.cs.toronto.edu/~toni/Courses/Complexity2015/handouts/cook-clay.pdf

[3]. Fortnow, L. (2009). The status of the P versus NP problem. Communications of the ACM, 52(9), 78-86. https://dl.acm.org/doi/fullHtml/10.1145/1562164.15 62186

[4]. Meo, A. R. (2021). “On the P versus NP question: A new proof of inequality”, Journal of Computer Science, 17/5, 511-524, 2021.

[5]. Meo, A. R. (2018). “On the P vs NP question: A proof ofinequality”, arXiv:1802.05484

[6]. Meo, A. R. (2008). Some theorems concerning the core function. In Concurrency, Graphs, and Models (pp.778-796). Springer, Berlin, Heidelberg. https://link.springer.com/chapter/10.1007/978-3-540-68679-8_48

[7]. Meo, A. R. (2022). On the P versus NP question.

[8]. Mulmuley, K. D., & Sohoni, M. (2001). Geometric complexity theory I: An approach to the P vs. NP and related problems. SIAM Journal on Computing, 31(2), 496-526.

[9]. https://doi.org/10.1137/S009753970038715X Razborov, A. A. (1985). Lower bounds for the monotone complexity of some Boolean functions. In Soviet Math. Dokl. (Vol. 31, pp. 354-357).