An Interesting Perspective to the P versus NP Problem

Wenming Zhang

School of Economics and Management, Northwest University, Xi'an, 710127, China
wenming@nwu.edu.cn

Abstract. We discuss the P versus NP problem from the perspective of addition operation about polynomial functions. Two contradictory propositions for the addition operation are presented. With the proposition that the sum of k ($k \leq n$) polynomial functions on n always yields a polynomial function, we prove that $P = NP$, considering the maximum clique problem. However, we also get a contradiction if we accept the proposition. So, we conclude that the sum of k polynomial functions may yield a exponential function. Accepting this proposition, we prove that $P \neq NP$ by constructing an abstract decision problem Π.

Keywords: *P versus NP Problem; NP-Complete Problem; Turing Machine; Addition Operation; Binomial Theorem*

1 Introduction

As one of the most important problems in mathematics and computer science, the P versus NP problem is to determine whether every language accepted by some nondeterministic algorithm in polynomial time is also accepted by some (deterministic) algorithm in polynomial time [1]. Since first mentioned in a 1956 letter written by Kurt Gödel to John von Neumann [2] and precisely stated in 1971 by Stephen Cook [3], the problem has been considered by many papers [4]. However, it is still open [5]. For detailed introduction of the P versus NP problem, please refer to the excellent survey articles by eminent authors (see [4]-[10]).

In this paper, we will discuss the P versus NP problem based on the addition operation of polynomial functions. It is often thought that the sum of k ($k \leq n$) polynomial functions on n always yields a polynomial function. Applying the binomial theorem, we can prove that $P = NP$ in this situation. However, we can also get a contradiction if we accept this proposition. So we conclude that the sum of k polynomial functions may yield an exponential function. With this proposition, we construct an problem separating P from NP.

The rest of this paper is organized as follows. Section 2 presents two simple and interesting properties about the binomial theorem. Section 3 presents two propositions for the addition operation. Section 4 discusses the P versus NP problem according to the propositions. Finally, Section 5 concludes this paper.

2 Story of Binomial Theorem

Firstly, let's remember the binomial theorem which is also called Yang Hui triangle in China [11].

$$(a+b)^n = C_n^0 a^n + C_n^1 a^{n-1} b + C_n^2 a^{n-2} b^2 + \cdots + C_n^{n-1} a b^{n-1} + C_n^n b^n \qquad (1)$$

where $C_n^k = \frac{n!}{k!(n-k)!}$ for $k = 0, 1, \cdots, n$ and $C_n^{k+1} = C_{n-1}^k + C_{n-1}^{k+1}$ for $k = 0, 1, \cdots, n-1$.

We present two simple and interesting properties about Eq. (1) in the following.

Lemma 1. *Let k be an arbitrary integer number such that $0 \leq k < n-1$. We have that $C_n^{k+1} = C_{n-1}^k + C_{n-2}^k + \cdots + C_{k+1}^k + 1$.*

Proof. Noting that $C_n^{k+1} = C_{n-1}^k + C_{n-1}^{k+1} = C_{n-1}^k + (C_{n-2}^k + C_{n-2}^{k+1}) = \cdots = C_{n-1}^k + C_{n-2}^k + \cdots + (C_{k+1}^k + C_{k+1}^{k+1})$ and $C_{k+1}^{k+1} = 1$, the lemma follows. □

Lemma 2. $2^n = C_n^0 + C_n^1 + \cdots + C_n^{n-1} + C_n^n$.

Proof. Replacing $a = b = 1$ in Eq. (1), the lemma follows. □

3 Two Contradictory Propositions

Let $h_1(n), h_2(n), \cdots, h_k(n)$ be arbitrary k polynomial functions on n, and $H(n) = h_1(n) + h_1(n) + \cdots + h_k(n)$, where $k \leq n$.

Is $H(n)$ polynomial or exponential? Noting that $H(n) \leq k \max_{1 \leq j \leq k} \{h_j(n)\} \leq n \max_{1 \leq j \leq k} \{h_j(n)\}$, you may say $H(n)$ is polynomial since $\max_{1 \leq j \leq k} \{h_j(n)\}$ is polynomial. Is this always right? Maybe not! And we have the following two propositions, which are applied to prove that $P = NP$ and $P \neq NP$, respectively. It may sound interesting.

Proposition 1. *Given arbitrary k ($k \leq n$) polynomial functions on n, i.e., $h_1(n), h_2(n), \cdots, h_k(n)$, and $H(n) = h_1(n) + h_2(n) + \cdots + h_k(n)$, $H(n)$ is always polynomial.*

Proposition 2. *There exist k ($k \leq n$) polynomial functions on n, i.e., $h_1(n)$, $h_2(n)$, $\cdots$, $h_k(n)$, such that $H(n)$ is exponential, where $H(n) = h_1(n) + h_2(n) + \cdots + h_k(n)$.*

4 Discussion for the P versus NP Problem

Two subsections are considered according to the two propositions above.

4.1 Proposition 1 holds

Considering the maximum clique problem, which is NP-complete [12], we will prove $P = NP$ in the following. Given a graph G with n vertices, we can find the maximum clique of G by Enumerative Algorithm(EA) with the worst case run time $f(n) = C_n^0 + C_n^1 + \cdots + C_n^{n-1} + C_n^n$.

Theorem 1. *$P = NP$ if Proposition 1 holds.*

Proof. It is sufficient to prove that $f(n)$ is polynomial. Noting Proposition 1, it is now sufficient to prove that C_n^0, C_n^1, $\cdots$, C_n^n are all polynomial. Mathematical induction is applied in the following.

Firstly, it is obvious that C_n^0 and C_n^1 are both polynomial. Now we suppose that C_n^k is polynomial for some k $(1 \leq k \leq n-1)$ and try to prove that C_n^{k+1} is also polynomial. Noting that $C_{k+1}^k < C_{k+2}^k < \cdots < C_{n-1}^k < C_n^k$ and C_n^k is polynomial, we have that $C_{k+1}^k, C_{k+2}^k, \cdots, C_{n-1}^k$ are all polynomial. Remembering Lemma 1 and Proposition 1, we have that C_n^{k+1} is polynomial. The theorem follows. □

However, according to Lemma 2, we have that $f(n) = 2^n$, which is exponential, contradicting to the proof for Theorem 2. So, we have an extraordinary conclusion that Proposition 1 does not hold.

Remark 1. Proposition 1 does not hold.

4.2 Proposition 2 holds

We will prove that $P \neq NP$ in the following. It is often thought that proving $P \neq NP$ involves proving a superpolynomial lower bound on the run time of any algorithm for some NP-complete problem such as SAT [6]. However, instead of any NP-complete problem, we will construct an abstract problem Π, such that $\Pi \in NP$ and $\Pi \notin P$.

From Proposition 2, we know that there exist k $(k \leq n)$ polynomial functions on n, i.e., $h_1^*(n), h_2^*(n), \cdots$, and $h_k^*(n)$, such that $H^*(n)$ is exponential, where $H^*(n) = h_1^*(n) + h_2^*(n) + \cdots + h_k^*(n)$.

Remember that the return of a decision problem is just a "yes" or "no". Let π_i $(i = 1, 2, \cdots, k)$ denote an abstract decision problem with input I_i, where the length of I_i is n and the worst case run time for π_i is $h_i^*(n)$. Moreover, we let Π be a decision problem which is to ask if there exists a "yes" in the returns of $\pi_1, \pi_2, \cdots$, and π_k. Note that the input of Π are $I_1, I_2, \cdots$ and I_k with a total length of $nk < n^2$.

Theorem 2. *$P \neq NP$ if Proposition 2 holds.*

Proof. It is sufficient to prove that $\Pi \in NP$ and $\Pi \notin P$.

Note that $I_1, I_2, \cdots$ and I_k are unrelated. So, for any deterministic Turing Machine, the worst case of solving Π is to check the k returns of $\pi_1, \pi_2, \cdots$ and π_k. And the total run time is $h_1^*(n) + h_2^*(n) + \cdots + h_k^*(n)$. Noting that

$H^*(n) = h_1^*(n) + h_2^*(n) + \cdots + h_k^*(n)$ and $H^*(n)$ is exponential, we get that $\Pi \notin P$.

For a non-deterministic Turing Machine, it is sufficient to check the return of one of the k decision problems with polynomial run time. So it is that $\Pi \in NP$.

The theorem follows. □

5 Conclusion

In this paper, we discuss the P versus NP problem from the perspective of addition operation about polynomial functions. According to the two propositions, i.e. whether the sum of k ($k \leq n$) polynomial functions on n always yields a polynomial function, $P = NP$ and $P \neq NP$ are proved, respectively. However, we can also get a contradiction if the first proposition holds. And we have to conclude that $P \neq NP$.

However, it is still hard for us to understand Proposition 2. Buddha Sakyamuni said that inequality of heart yields annoyance. Maybe that the polynomial and exponential functions are not absolutely different but naturally interrelated.

Acknowledgements

The authors would like to acknowledge the financial support of Grants (No. 71201123.) from NSF of China.

References

1. S.Cook: The P versus NP problem, Clay Mathematics Institute. http://www.claymath.org/sites/default/files/pvsnp.pdf
2. J.Hartmanis: Gödel, von Neumann, and the P = NP problem, Bulletin of the European Association for Theoretical Computer Science, 38, 101-107 (1989)
3. S.Cook: The complexity of theorem proving procedures, Proceedings of the Third Annual ACM Symposium on Theory of Computing, 151-158 (1971)
4. C.Papadimitriou: NP-completeness: A retrospective, International Conference on Automata, Languages, and Programming, LNCS 1256, 2-6 (1997)
5. L.Fortnow: The status of the P versus NP problem, Communications of the ACM, 52, 78-86 (2009)
6. E.Allender: A status report on the P Versus NP question, Advances in Computers, 77, 117-147 (2009)
7. M.Sipser: The history and status of the P versus NP question, ACM Symposium on Theory of Computing (STOC), 603-618 (1992)
8. S.COOK: The importance of the P versus NP question, Journal of the ACM, 50(1), 27-29 (2003)
9. R. Impagliazzo: Computational complexity since 1980, Conference on Foundations of Software Technology and Theoretical Computer Science, LNCS 3821, 19-47 (2005)
10. A. Wigderson: P, NP and mathematics - a computational complexity perspective, Proceedings of the ICM 2006, 1, 665-712 (2007)

11. A.K.Bag: Binomial theorem in ancient India, Indian Journal of History of Science, 1(1), 68-74 (1966)
12. R.M.Karp: Reducibility Among Combinatorial Problems, Complexity of Computer Computations, New York: Plenum, 85-103 (1972)