P != NP – FINAL PROOF OF THE MILLENNIUM THEOREM

Mirzakhmet Syzdykov

mspmail598@gmail.com

Satbayev University, Almaty, Kazakhstan

ABSTRACT

We state that if there's an order in the target function for the set of variables, then P != NP according to the dynamic programming which is the most optimal way of solving the combinatorial problems using its recurrent function at each step of the algorithm.

INTRODUCTION

The problem was first stated by Stephen Cook, from that time on now we have no defined answer if there's the possibility of finding the polynomial solution for NP-complete problems.
We assume that the growing speed of polynomial and non-polynomial functions is compared to each other:

$$O(n^t) < O(2^n) < O(n!) < O(n^n) \quad , \qquad (1).$$

PROOF

Since there's no known algorithms whose polynomial speed is faster than the speed of growth of non-polynomial functions like power-set or factorial, then it naturally follows that if P and NP are equal, then speed of algorithm steps in P is equal or greater than number of steps required for the NP-complete task:

$$\frac{\partial x^t}{\partial x} \geq \frac{\partial 2^x}{\partial x} \rightarrow P = NP \quad , \qquad (2).$$

The same applies to the factorial function and its equivalent value as function in form x^x.

As *t* is a free parameter then we see that:

$$t \cdot x^{t-1} < \ln(2) \cdot 2^x \quad , \qquad (3).$$

The inequality (3) is true for factorial also.

Thus, from (2) and (3) it naturally follows that the P != NP as the speed of growing non-polynomial function is much bigger than the same speed defined by derivative of polynomial function in P.
The plot of P and NP functions along with their derivatives can be seen on Figure 1.

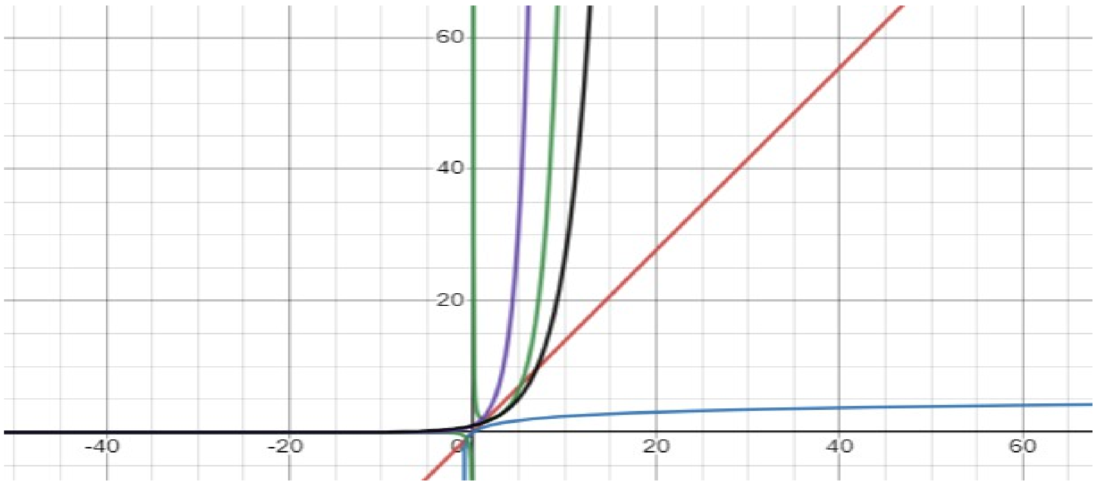

Figure 1. Graphical plot of functions and derivatives

From all the above, it still doesn't follow that P != NP, however, for ordered set there's no optimal solution in P as the fastest possible algorithm's complexity is *O(n)* for iterative methodology, as it's presented at each step before inclusion at the stage set using dynamic programming, which is the most optimal as it's defined by recurrence relation at each stage.

CONCLUSION

We have shown that classes P and NP aren't equal over the fact of derivatives defining the speed of the polynomial and non-polynomial functions respectively. It naturally follows that even for infinite order of derivative comparison the speed of non-polynomial functions is still increasing while the speed of polynomial function converges to absolute zero:

$$\lim_{n\to\infty}\frac{\partial^n x^t}{\partial t^n} < \lim_{n\to\infty}\frac{\partial^n 2^x}{\partial t^n} < \lim_{n\to\infty}\frac{\partial^n x^x}{\partial t^n} \text{ , (4).}$$

If there exist the ordered set in target function like in Traveling Salesman Problem (TSP), then P != NP according to the obvious statement that dynamic programming cannot cover all the space of possible cases.

REFERENCES

Cook S. The P versus NP problem //Clay Mathematics Institute. – 2000. – T. 2.

Cook S. The importance of the P versus NP question //Journal of the ACM (JACM). – 2003. – T. 50. – №. 1. – C. 27-29.

Syzdykov M. Functional hypothesis of complexity classes //Advanced technologies and computer science. – 2022. – №. 3. – C. 4-9.

Fortnow L. The status of the P versus NP problem //Communications of the ACM. – 2009. – T. 52. – №. 9. – C. 78-86.

Sipser M. The history and status of the P versus NP question //Proceedings of the twenty-fourth annual ACM symposium on Theory of computing. – 1992. – C. 603-618.

Bellman R. Dynamic programming //Science. – 1966. – T. 153. – №. 3731. – C. 34-37.