

ON THE DUALITY FEATURE OF P-CLASS PROBLEMS AND NP COMPLETE PROBLEMS

WenhongTian^{1,2}

¹University of Electronic Science and Technology of China,
Chengdu, China.

²Chongqing Institute of Green and Intelligent Technology,
Chinese Academy of Science.

ABSTRACT

In term of computational complexity, P-class (abbreviated as P) problems are polynomial-time solvable by deterministic Turing machine while NP complete (abbreviated as NPC) problems are polynomial-time solvable by nondeterministic Turing machine. P and NPC problems are regularly treated in different classes. Determining whether or not it is possible to solve NPC problems quickly, is one of the principal unsolved problems in computer science today. In this paper, a new perspective is provided: both P problems and NPC problems have the duality feature in terms of computational complexity of asymptotic efficiency of algorithms.

KEYWORDS

P Problems; NP problems; NP Complete Problems; the P versus NP Problem; The Duality Feature.

1. INTRODUCTION

In 1971, Cook [1] firstly established a theorem that a class of problems can be P-reducible (polynomial time reducible) to each other and each of them can be P-reducible to Boolean Satisfiability (SAT) problem, this class of problems is called NP (nondeterministic polynomial time) problems. Karp [2] applied Cook's 1971 theorem that the SAT problem is NP Complete (also called the Cook-Levin theorem) to show that there is a polynomial time reduction from the SAT problem to each of 21 combinatorial problems, thereby showing that they are all NP complete (NPC). This was one of the first demonstrations that many natural computational problems occurring throughout computer science are computationally intractable, and it drove interest in the study of NP-completeness and the P versus NP problem [3]. The P versus NP problem, determining whether or not it is possible to solve NP problems quickly, is one of the principal unsolved problems in computer science today and listed as one of seven millennium problems [3,4], challenging tens of thousands of researchers.

Simply speaking, P problems mean that the class of problems can be solved exactly in polynomial time while NPC problems stands for a class of problems which might not be solvable in polynomial time. NPC problems has far-reaching consequences to other problems in mathematics, biology, philosophy and cryptography. More specifically, in Big O-notation (refer

to Definition 1 and 2) of computational complexity in asymptotic efficiency of algorithms, P problems can be solved in polynomial time of $O(n^k)$ for some constant k where n is the size of input to the problem, while NPC problems may have computational complexity of $O(2^{cn})$ including both exponential time and sub-exponential time, where c is a positive constant larger than zero.

Karp [2] ever claimed that if any of NPC problems have efficient polynomial time algorithms, then they all do. It is for this reason that research into the P versus NP problem centers on NPC problems, i.e., looking for efficient polynomial time algorithms for NPC problems. Through decades' efforts by many researchers, it is still an open question. There are quite many results but none of them is commonly accepted yet by the research community.

The current author also classified NPC problems based on their natures and other methods [5] and found that NPC problems are not equivalent in computational complexity.

In the following, we show a new perspective: both P problems and NPC problems have duality features of each other in terms of computational complexity of asymptotic efficiency of algorithms, especially by considering the representation of the input.

2. PROBLEM FORMULATIONS

The following definitions are based on the computational complexity of different problems in the worst case.

Definition 1: The asymptotic efficiency of algorithms [6,7]: concerns with how the running time of an algorithm increases with the size of the input *in the limit*, as the size of the input increases without bound.

Definition 2: The O-notation of computational complexity of an algorithm: asymptotically bounds a function from above within a constant factor [6,7]. For a given function $g(n)$, we denote by $O(g(n))$:

$O(g(n)) = \{f(n) : \text{there exist positive constants } c \text{ and } n_0 \text{ such that } 0 \leq f(n) \leq cg(n) \text{ for } n \geq n_0\}$.

Fig.1 shows the intuition behind O-notation. For all values n to the right of n_0 , the value of function $f(n)$ is on or below $cg(n)$.

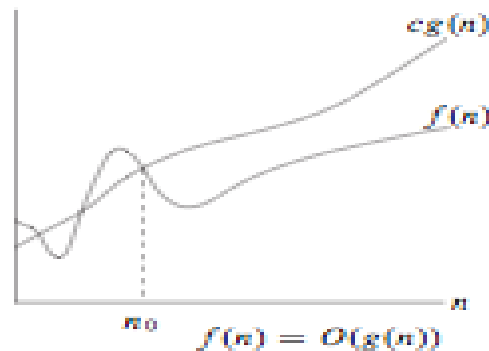


Figure 1. $f(n) = O(g(n))$ [6]

Since O-notation describes an upper bound, when we use it to bound the worst-case running time of an algorithm, we have a bound on the running time of the algorithm on every input. For example, the doubly nested loop structure of the insertion sort algorithm has an $O(n^2)$ upper bound; equivalently, we mean that the worst-case running time is $O(n^2)$.

Definition 3: P-class problems in term of O-notation of computational complexity: in the worst case, their exact algorithms have computational complexity of $O(n^k)$ for some constant k where n is the size of input to the problem.

Definition 4: NPC problems: in the worst case, their exact solutions may have computational complexity of $O(2^{cn})$ **in O-notation**, where n is the size of input to the problem and c is a positive constant.

Table 1 provides a summary of variables used in this paper. Table 2 shows computational complexities of exact solutions to some NPC problems (abstracted from [8]).

Table 1. The summary of variables

Variables	Meaning
k, c, n_0	a positive constant
n, e	The size of inputs to a problem
s, m	The number of bits (binary)
B_0	$\min(2^{cn}, n^k, 2^m, 2^s)$
W, C, b	a positive number

Table 2. The complexities of exact solutions to some NPC problems from [8]

Problem	Complexity	Genre
MIS (maximum independent set)	1.1996^n	Packing
Set covering problem (SCP) *	1.1996^n	Covering
3CP (3-coloring of planar graph)	1.3446^n	Partitioning
SSP (subset sum problem) [7]	nW	Numerical
3SAT	1.4802^n	Satisfaction
TSP	$n^2 2^n$	Sequencing

*: derived from MIS since SCP and MIS are complementary [7].

We also consider that the representation of the input to the problem by m bits (in binary) and the space complexity of a problem by s bits (in binary) in memory.

Fact 1: A regular system (the computer hardware, called the system in the following) has capability of handling B_0 numbers in term of computational and space complexity, i.e., handling the input efficiently within reasonable time (may be within a few hours or minutes depending on the applications) without overflowing the system, where B_0 may be related to the memory size or whichever (CPU, memory, disk et.) is the bottleneck of the system.

Based on the definitions above, the capability B_0 can be represented by

$$B_0 = \min(2^{cn}, n^k, 2^m, 2^s) \quad (1)$$

3. RESULTS

Lemma 1. Dynamic programming algorithm for Subset Sum problem (SSP) is pseudo-polynomial time.

Proof. removed ■

Lemma 2. Any NPC problem can be reduced to SSP.

Theorem 1: The instances of SSP can have computational complexity of NPC problems and P-class problems, this is called the duality feature in this paper.

Proof: removed ■

Fact 2: The instances of other NPC problems also have the duality feature.

Proof: removed. ■

In Table 3, records of optimum solutions to TSP problem are provided. One can see that the size of the problem is increasing as year goes. Notice that the instance with 1904711 nodes is still not yet solved exactly but just has a good lower bound [14]. Similar results are also observed for other NPC problems. For real-life problems with small or moderate number of variables, they can be solved to exactly very easily (LKH [15] and Concorde [16] are considered as two of the best such solvers which can now find exact solutions for medium size TSP Problems). However, if the problem size increases to very large, finding efficient solutions to NPC problems just become intractable.

Theorem 2: The P-class Problems can have the duality feature in terms of computational complexity.

Proof: removed. ■

Remarks: Actually SSP can be treated as polynomial time solvable problem (P-class problem) when both n and W is not very large while it is of exponential complexity when both n and W are very large.

Observation 1: Even one can find efficient solution (polynomial time algorithm) to one or more NPC problems, some P problems may become exponential time solvable in computational complexity when the representation of the input to them become very large.

4. DISCUSSIONS AND CONCLUSION

NPC problems have different natures, they can be classified into six basic genres [2, 7], i.e., Satisfaction, Packing, Covering, Partitioning, Sequencing, Numerical computing.

Originally, PRIMES and Graph Isomorphism were hard to determine to be in NPC or not in Cook's paper [1]. In 2004, PRIMES is found in P-class and accepted by the research community [10]. And in 2015, Graph isomorphism is reported to have Quasipolynomial time solution [11,

12], though the results are still under verification. These show some new perspectives and trends on NPC problems.

As another perspective, it is recently proved mathematically that memcomputing machines (a novel non-Turing paradigm) have the same computational power of nondeterministic Turing machines [13]. Therefore, they can solve NPC problems in polynomial time with resources that only grow polynomially with the input size.

NPC problems and the P versus NP problem challenge many researchers to tackle them through decades of efforts. In this paper, a new perspective is provided: both P problems and NPC problems have the duality feature in terms of computational complexity of asymptotic efficiency of algorithms.

Table 3. Records of optimum solutions to TSP problems [14] where n is the number of nodes in TSP. All TSP problems in the table are solved to optimum except for the last one

n	Year (solved)	Node type
48	1954	USA cities
64	1971	random nodes
80	1975	random nodes
120	1977	Germany cities
318	1987	cities
532	1987	USA cities
666	1987	World cities
1002	1987	cities
2392	1987	cities
3038	1992	cities
13509	1998	USA cities
15112	2001	cities
24978	2004	Sweden cities
85900	2006	cities
100000	2009	Japan
1904711	2010*	World TSP Challenge

ACKNOWLEDGMENTS

The abstract version of this manuscript is posted at my research gate.

REFERENCES

- [1] S.A. Cook (1971). The Complexity of Theorem Proving Procedures, Proceedings of the third annual ACM symposium on Theory of computing. pp. 151-158, March of 1971.
- [2] R. M. Karp (1972), Reducibility Among Combinatorial Problems, In R. E. Miller and J. W. Thatcher (editors). Complexity of Computer Computations. New York: Plenum. pp. 85-103.
- [3] Clay Mathematics Institute, <http://www.claymath.org/millennium-problems/millennium-prize-problems>.

- [4] Wikipedia, <http://en.wikipedia.org/wiki/NP-complete>.
- [5] WenhongTian, NP Complete Problems Are not All Equivalent, under review, submitted 2017.
- [6] T. Cormen, C. Leiserson, R.L. Rivest, C. Stein, Introduction to Algorithm, 2nd edition, MIT Press, 2004.
- [7] J. Kleinberg, E. Tardos, Algorithm Design, 2006, Pearson Education Asia Limited, pp. 475-479.
- [8] G. J. Woeginger, Exact algorithms for NP-hard Problems: A Survey, Combinatorial optimization - Eureka, you shrink! Pages 185 - 207 , Springer-Verlag New York, Inc. New York, NY, USA ©2003
- [9] WenhongTian, C. Huang, X. Wang, A Near Optimal Approach for Symmetric Traveling Salesman Problem in Eclidean Space, In Proceedings of ICORES 2017, Portugal, also available at arxiv <https://arxiv.org/pdf/1502.00447.pdf>
- [10] M. Agrawal, N. Kayal, N. Saxena, PRIMES is in P, The Annals of Mathematics, Pages 781-793 from Volume 160, Issue 2, 2004.
- [11] L. Babai, Graph Isomorphism in Quasipolynomial Time, <https://arxiv.org/abs/1512.03547>, Dec 11, 2015.
- [12] A. Cho , Science news, mathematician-claims-breakthrough-complexity-theory, <http://www.sciencemag.org/news/2015/11/mathematician-claims-breakthrough-complexity-theory>, Nov. 11, 2015.
- [13] F. L. Traversa, C. Ramella, F. Bonani and M.D. Ventura, memcomputing NP-complete problems in polynomial time using polynomial resources and collective states, Science, Vol. 1, no. 6, e1500031, Nov. 2015.
- [14] W. Cook, In Pursuit of the Traveling Salesman, Princeton University Press, 2012.
- [15] LKH codes, <http://www.akira.ruc.dk/~keld/research/LKH/>
- [16] Concorde codes, <http://www.math.uwaterloo.ca/tsp/concorde.html>