

NP Complete Problems – A Survey

Saranya Vatti

“Once more, we have decreased the number of open questions in the field - without, alas, increasing much the number of answers!”

(C.H. Papadimitriou and M. Yannakakis, "Optimization, Approximation, and Complexity Classes." JCSS 43:425-440, 1991.)

Introduction

In 1936, Alan Turing invented something he called the a-machine – known to us as the Turing Machine. He defined a theoretical computational model that can be constructed into a machine, given any algorithm, that would edit the symbols on an infinite tape. This formed the foundation for the computational complexity theory that would be defined formally much later. A number of scientists worked on the problem of complexity and on solutions with proofs for the computational complexities. The paper by Hartmanis and Stearns that was based on the work of Turing, formally defined and established the field of computational complexity. Stephan Cook, now considered one of the forefathers of computational complexity theory, and Leonid Levin independently published papers that not only defined but also showed the existence of NP complete problems. In his paper, Cook gave the first proof for the decision problem, Boolean Satisfiability problem based on Turing reductions. In 1972, Richard Karp used Cook's paper to publish “Reducibility among Combinatorial Problems,” consolidating a number of problems including Clique, Vertex Cover, Hamiltonian Cycle to be NP-Complete. This was among the first of its kind and among the most important. It paved the way for many other notable scientists to build on existing proofs and to reduce and connect more problems.

NP-Completeness and its importance

Shortly after the spurge of initial papers that define and prove complexity theory, there came a volley of publications that give a more precise and a more formal definition of complexity classes and proofs. Over time and with many man-efforts spent in trying to solve some hard problems that would later be classified as NP-Hard or NP-Complete, scientists gradually searched for a formal proof that the problems could not be solved as opposed to searching for an efficient solution. NP-Completeness filled this need for the proof.

By definition, NP-Complete problems are a set of problems that belong to both NP and NP-hard. A problem belongs to the “NP” complexity class if there is a Non-deterministic algorithm that can solve the problem in Polynomial time. Informally, the problem has to be verifiable in polynomial time. A problem belongs to NP-Complete if it is in NP and every problem in NP is reducible to it in polynomial time. NP-Complete problems are also among the hardest problems to solve. This makes the identification and reduction of NP-Complete problems all the more important. There are many real world algorithms and practical applications that demand the solution of any one among NP-Complete problems. Having a repository of pre-reduced and pre-proved problems will enable future scientists to further the research on

diverse problems, perhaps coming ever so close to solving them. With this classification and reduction, an advance in the algorithm of one problem can easily help improve the efficiency in the current solution of another problem – which can improve drastically.

Once a problem was proved to be NP-Complete, scientists could then look to other ways of solving/approximating the solution. This was of a great boost to the Computer Science community as practical algorithms were invented, where none existed earlier. Computational problems are now classified into a number of classes and the relationships between the properties of problems within the same classes can be studied.

Conclusion

P vs NP problem is one of the seven problems still open for the Millenium Prize by the Clay Mathematics Institute with a one million dollar fund allocated to the winner who solves the $P \stackrel{?}{=} NP$ problem. Although not proven, it is believed that P (the class of problems that can be solved in some known polynomial function of the input time and space) is strictly greater than NP.

References

1. Wikipedia [https://en.wikipedia.org/wiki/List_of_NP-complete_problems; <https://en.wikipedia.org/wiki/NP-completeness>]
2. Johnson DS (2012) A brief history of NP-completeness, 1954–2012. Documenta Mathematica. Extra Volume ISMP, pp 359–376
3. Hartmanis, J., and R. E. Stearns, [1965]. "On the computational complexity of algorithms," Trans. Amer. Math. Soc., 117, 285--306.
4. Fortnow, L. and Homer, S. A short history of computational complexity. Bulletin of the European Association for Theoretical Computer Science 80, (June 2003).