Exercises of chapter 4

February 19, 2020

Mayra Cristina Berrones Reyes

1 Boolean satisfiability problem

This problem, abbreviated SAT, determines if the variables of a boolean formula can be replaced by *TRUE* or *FALSE* values. If the case is *TRUE* the formula is satisfiable. The complementary case in which the value is *FALSE* for all possible assignments, then the formula is unsatisfiable.

SAT was the first problem that was proven to be NP-complete, stated by Cook-Levin theorem. This theorem basically states that if there exists a deterministic polynomial algorithm for solving Boolean satisfiablility, then every problem NP-complete can be solved by a deterministic polynomial algorithm. This sort of question can be equivalent to the P versus NP problem, which has been studied for many years, and to this day it is still considered a popular unsolved problem in computer science [1].

The SAT tool helps us determine the satisfiability of a boolean formula. An algorithm by Davis Putnam Logemann Loveland (DPLL) makes a search through all of the possible interpretations of the problem to prove if the algorithm can find an interpretation that satisfies the problem, or if the algorithm explores all the search space and does not find a solution, in which case the problem is unsatisfiable.

This method uses the Backtracking technique, which is a search method that analysis all possible combinations of the True values of the variables. This, in conjunction with the pruning function, determines if it is posible to arrive to an answer giving certain node in the tree [2].

2 Hamiltonian paths and cycles

A very important problem of graph theory is the search for one cycle or path that pases through every vertex exactly one time. One of the definitions of hamiltonian cycles is that is a circuit that starts in vertex a and goes through all of the other vertices exactly once and then returns to the starting vertex a. A path, similar to the cycle, starts in vertex a passes through all of the other vertices only once, and finally stops on the final vertex [3].

The problem of finding either a hamiltonian path or cycle is a NP-complete problem, making it highly unlikely to find a polynomial algorithm for solving it. Many studies have tried to give a linear solution to solving a hamiltonian path, giving it certain rules that do not hold up well when we change some of the parameters such as dimensionality. Another way is to use the algorithm of 3SAT in which the graph will be constructed of various parts to represent the variables of each clause that appear in the SAT problem. Then, by solving the SAT problem, we will determine if we can connect all the different paths [4].

Almost all studies agree in certain rules to prove which graphs can not hold a hamiltonian path or cycle, that depends on the number of nodes in the graph, if it has directions or not, etc. All of this, has not been sufficient to prove that not all the hamiltonian cycles can be solved in polynomial time.

3 Conclusions

On this investigation I found the NP vs P theme being mentioned quite repeatedly. I have always found that problem very intimidating because of all of the time and effort people have put in finding an answer. Now, seeing some of the examples used to find simpler solutions to this two problems that are considered to be NP, I can see that maybe the NP label is often used on problems with higher dimensionality, which happens in many other problems. If you increase the variables, parameters and dimensions, all problems will increase in time and computational complexity.

4 Reference

References

[1] Carlos Ansotegui, Felipe M. An introduction to Satisfiability algorithms. Dpto de informatica e Ing. Industrial. 2003

[2] T. Covelli, E. Horquin, M Santillan. SAT una herramienta didactica para el problema de satisfactibilidad.

[3] Notes of Eulerian and hamiltonian paths. `https://www.csd.uoc.gr/~hy583/reviewed_notes/euler.pdf`

[4] J. Bang-Jensen, G. Gutin. On the complexity of hamiltonian path and cycle problems in certain classes of digraphs.