International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-11 Issue-7, June 2022

Review on Intractability

Krishnendu Basuli

Abstract: ***In real life, the problems may be of infinite dimensions. Design of ingenious information structure, for minimizing complexity and redundancy of the problem space are versatile. This is unique in the sense: given an arbitrary 'n' node graph the problem becomes countable infinite and in some cases it is uncountable infinite. Problems which can be mapped as graphs are normally simple in nature but we remember the adage "Simple things are mighty things". What we mean that normally for example it is a practice to bring undue mathematics to make things complex but although the graph algorithms are of exponential complexity for large dimension.***

Keywords: Intractability, Algorithm, Complexity, problems, NP, NP-Complete, NP-Hard, Approximation Algorithm, Heuristic, Reducibility.

I. INTRODUCTION

Computers are information processing machines. They can solve or supposed to solve any [hayes,2nd] real life problem in reasonable amount of time and space. If so, the problems are called tractable otherwise it is intractable. Computer hardware is changing day-by-day and so the characteristics of tractable problems. Anyway, we can customize an intractable problem in such a way that for certain problem size which changes day-by-day, we approximate the problem to be tractable.

Some problems are unsolvable in practical sense. For example, can we write a [6] program that can test all programs, including itself for correctness? No such program can exist. Russel's paradox leads us to same conclusion [5]. However, for certain universe we can get the result we desire. Some problems are such that (Non-deterministic Polynomial) they belong to a class. Any member of the class will take years for the optimum solution by any super computer, but verifying a given instance of solution takes only seconds. The solution of any problem of this class means solutions of every other. The problems are Traveling Salesman Problem (TSP), optimum schedules, classification of data and as many as 3000 common problems for science, engineering and commerce. We call these problems intractable.

II. PROBLEMS, ALGORITHMS, COMPLEXITY

A problem will be a general question to be assigned, usually possessing several parameters, or free variables, whose values are left unspecified.

Manuscript received on 04 May 2022.
Revised Manuscript received on 18 May 2022.
Manuscript published on 30 June 2022.
* Correspondence Author
Dr. Krishnendu Basuli*, Assistant Professor, Department of Computer Science, West Bengal University, Barasat, Kolkata (West Bengal), India. E-mail: krishnendu.basuli@gmail.com

A problem is described by giving 1) a general description of all its parameters, and 2) a statement of what properties the answer, or solution, is required to satisfy. An instance of a problem is obtained by specifying particular values for all the problem parameters [6]. An algorithm is a finite set of rules that gives a sequence of operations for solving a specific type of problem, an algorithm has five important features.

i) Finiteness:- An algorithm must terminate after a finite number of steps.
ii) Definiteness:- Each step of algorithm must be precisely defined: the action to be carried out must be rigorously and unambiguously specified for each case.
iii) Input:- An algorithm has zero or more inputs.
iv) Output:- An algorithm has one or more outputs.
v) Effectiveness:- An algorithm is also generally expected to be effective, in the sense that its operations must be sufficiently basic that they can in principle be done exactly and in a finite length of time by someone using pencil and paper.

Analysis of algorithm is the name to use to describe investigate such as this. The general idea is to take a particular algorithm and to determine its quantitative behavior, occasionally we also study whether or not an algorithm is optimal in some sense . The theory of algorithm is another subject entirely, dealing primarily with the existence or nonexistence of effective algorithm to compute particular quantities. The complexity is the time and space required for the algorithm to run in the computer in respect to the input size. The complexity function for an algorithm express its time or space requirement by giving, for each possible input length, the largest amount of time and space needed by the algorithm to solve a problem instance of that size.

III. CLASSIFICATION OF GRAPH ALGORITHMS

Graph algorithms are classified according to the order of magnitude of number of operations expressed as a constant power of the number of vertices n of edges e. The lower the value of the power the better the algorithm is. whenever we are confronted with a new problem, a natural first question to ask is: Can it be solved with a polynomial time algorithm? If the answer to this question is obviously 'yes' then nothing further can be said about the problem from the standpoint of NP-Completeness. We should try to find out an efficient polynomial time algorithm. If no polynomial time algorithm is apparent, an approximate second question to ask is : Is the problem NP-Complete?

Decision problems:- Any problem for which the answer is either zero or one is called decision problem. Weather an element 'X' is in a set of number 'S'?

Optimal problems: Any problem that involves the identification of any optimal values of a given cost function is known as an optimization problem. An optimization algorithm is used to solve an optimization problem. Finding the minimum cost of the spanning tree of a given weighted graph 'G'.

NP(Non deterministic polynomial time): NP is the class of decision problems whose positive solutions can be verified in polynomial time given the right information or equivalently whose solution can be found in polynomial time on a non deterministic machine.

P: The class P consists of all those decision problems that can be solved on a deterministic algorithm in polynomial time.

NP-Complete: A decision problem X is NP-Complete if 1.X belongs to NP 2.Y<=X for every problem Y belongs to NP. We all know that finding the shortest path from a single source in a directed graph in polynomial time. But finding the longest simple path between two vertices is NP-Complete[3].

NP-Hard: A problem L is NP-Hard if and only if satisfy property 2 of NP-Complete but not necessity of property 1 is called NP-Hard. Optimization problem may be NP-Hard. Such as Halting problem is an example of NP-Hard problem.

NPI: NP-Intermediate class consist of those problems that are in NP but not in P or NPC. Graph Isomorphism is an open problem that belongs to NPI[2].

These problems are generally regarded as tractable, but for philosophical, not mathematical, reason[3].

Cook's Theorem:-CNF Satisfiability is NP-Complete.

2.3.6. **Reducibility:-** We reduce a problem A to another problem B by proving a transformation that.

1) take any instance Ia of A and

2) Returns an instance Ib of problem B.

3) Such that an answer to Ib gives an answer to Ia.

Example: Multiply every value by -1 to reduce finding the maximum to finding the minimum.

e.g. { 10,-3,4,2,6}= {-10,3,-4,-2,-6} if the transformation is easy to compute , problem A is easier.

IV. HEURISTIC ALGORITHM

A heuristic is a techniques which seeks good (i.e. near-optimal) solutions at a reasonable computational cost without being able to guarantee either feasibility or optimality a particular feasibility solution is close to optimality a a particular feasibility solution is.

In heuristic algorithm, we make an interesting guesswork to obtain either an optimal or some good solutions.

Heuristic are typically used when there is no known method to find an optimal solution under the given constraints (time, space etc.) or at all.

Two fundamental goal in computer science are finding algorithms with probably good runtimes and with probably good or optimal solution quality.

5. Approximation algorithm:- An approximation algorithm for P is an algorithm that generates approximation solution for P [2].

We design approximate algorithm to find some kind of solution, which is related to the optimal solution by a bound of some kind. Even if exact algorithm is available a good approximation algorithm may be helpful some times. Approximation algorithms are usually just heuristic or rules that on the surface look they might solve the optimization problem exactly. However they do not . Instead, they only guarantee to generate feasible solutions with values with in some constant or some factor of the optimal value. Being heuristic in nature these algorithms are very much dependent on the individual problems being solved.

V. ANALYSIS OF GRAPH ALGORITHMS:

Analysis of an algorithm refers to the process of determining how much computing time and storage an algorithm will require. The study of information organization in combinational data structures is one of the major field of Computer Science. The use of well suited data organization is often the key for simplifying the representation of problem and its solution. Specifically, structured information, representing aggregates of objects and the relationship among them in the form of mathematical group are frequently used in Computer Science theory and applications as a very useful abstraction for real data. Since graphs are a highly general form of data structure, their sub classes i.e. sequence and trees also are frequently applied to represent the analyzed domain. Algorithms on graphs are worth studying not because they only solve some complex problems of graph theory but because they also give a general approaches for the solutions of a wide spectrum of computer algorithms.

VI. CONCLUSION

Computer science is the science and technology associated with design and analysis of algorithm [3]. Our observation shows that algorithms design and complexity analysis are inherently tide with graph theocratic examples. For a computer scientist exploration of graphs is like a space maker in real life. As observed by Knuth [7] graph theoretical terminology and graph theorist are numerically comparable at this point of time. However we have chosen this area because it may give us a life line in algorithm, efficiency, deficiency and performance profile. Its pleasure to explore the analogy between graphs and algorithms. Think it may inspire human endeavor in this area in such a way that a person can be happy to study it instead allurement of lady nicotine.

ACKNOWLEDGEMENT

My special thanks to my Ph. D guide Prof. Samar Sen Sarma. I want to acknowledge my parents and student throughout my career.

REFERENCES

1. Arumugam S, Ramachandran S, Invitation to graph Theory, Scitech Pvt,2002.

2. Cormen T.H., Leiserson C.E, Rivest R.L., Stein C., Introduction to Algorithms,PHI,2008.
3. Denning P.J.et al, Computing as a decipline, Communications of ACM, January 1989 Vol 32,Number 1. [CrossRef]
4. Deo N.,Graph Theory with applications to Engineering and Computer Science,PHI.,2006.
5. Farrell E. J, Wahid S. A., On the Reconstruction of the matching Polynomial and the Reconstruction Conjecture, Int. J. Math & Math Sci., Vol.10,No.1,1987,pp.155-162. [CrossRef]
6. Garey M. R. and Jhonson D.S, Computers and Intractability: A Guide to the Theory of NP-Completeness, Freeman, San Fracisco,1979.
7. Knuth D.E.,The art of computer programming, Volume 4,Fascicle 0,2008.
8. Minty , G.J.,A simple algorithm for listing all the trees of a graph, IEEE Trans. Circuit Theory,12(1965),120. [CrossRef]
9. Peikarski M., Listing of all possible trees of a Linear Graph,ibid.,CT-12, Corresn.,pp. 347-359,1961.
10. Reingold E.M, Neivergelt J., Beo N. , Combinatorial Algorithms: Theory and Practice, PHI, 1977.
11. Sarma Sen S., etal, All circuits of a symmetric Graph, Editor Report, IEEE,1981.
12. Sarma Sen S., et al, An efficient tree generation algorithm, Journal of Institution of Electronics and Telecommunications Engg.(IETE),Vol 27,No 3,pp 105-109,1981. [CrossRef]
13. Srinivas M., Patnaik L.M. Genetic Algorithms: A Survey.,IEEE,1994. [CrossRef]
14. Tutte W.T., Graph Theory As I Have Known It, Oxford Science Publication,1998.

AUTHOR PROFILE

Dr. Krishnendu Basuli, Assistant Professor in the Department of Computer Science, West Bengal State University. He qualified UGC-NET with JRF in June 2005. He was associated with Surendranath Evening College. He published more than 40 international journal and 4 books. He presents various international conferences regarding his work. His research interest is in Algorithm, Soft Computing, Graph Theory, Bioinformatics and Bio-Chip design etc. His teaching experience is about 15+ years now.