IJCSE International Journal of Computer Sciences and Engineering Open Access
Review Paper Vol.-6, Issue-5, May 2018 E-ISSN: 2347-2693

On Combinatorial Algorithms with Special Emphasize on Graph and Graph Algorithms

Debajit Sensarma

Department of Computer Science, Vivekananda Mission Mahavidyalaya, Vidyasagar University

**Corresponding Author: debajit.sensarma2008@gmail.com*

Available online at: www.ijcseonline.org

Accepted: 18/May/2018, Published: 31/May/2018

***Abstract*—** Combinatorial Algorithm frequently called Combinatorial Computing, deals with problem on how to carry out computations on discrete mathematical structures. It is all about finding patterns or arrangements that are best possible ways to satisfy certain constraints. Popularity of Combinatorial Algorithm is increasing day by day because outside the traditional areas of applications of mathematics to the physical sciences, discrete mathematical structures (e.g. permutation, graph etc.) occur more frequently than continuous ones, and the fraction of all computing time spent on problem that arise in the physical science is decreasing. Starting about 1970s, computer scientists experienced a phenomena called "Floyd's Lemma": problems that seemed to required n^3 operations could actually be solve in $O(n^2)$; problem that seemed to require n^2 time could be handled in O(nlogn) and also nlogn was often reducible to O(n). Besides this, running time of $O(2^n)$ can be reducible to $O(1.5^n)$ to $O(1.3^n)$ etc. Thus, though unlike other fields Combinatorial Algorithm does not have a few "fundamental theorem" that form the core of the subject matter and from which most of the result can be derived, the art of writing such algorithm in a tricky way or improvement of existing algorithms rather improvement in processor speeds, can save years or even centuries of computer time. Having back and forth over the territory of the Combinatorial Algorithm so often, the author is now charged to prepare the paper for looking the field from our view point in a nutshell with a special emphasize on Graph and real life applications of graph algorithms in the areas like network routing, information security, vulnerability analysis, storage of data and Coding and Information Theory.

***Keywords*—**Graph, Combinatorial Algorithms, Security, Integity, Storage Space

I. INTRODUCTION

Algorithm is the finite set of rules that gives a sequence of operations for solving a specific type of problem. The word algorithm comes from the name of a Persian mathematician, A.J.M. ibn Musa al Khowarizmi. Design and analysis of efficient algorithm is a very important aspect because it is the main of computer science. The notion of algorithm is basic to all of computer programming. An algorithm has five important features. They are in short, Finiteness, Definiteness, Input, Output, and Effectiveness [1,2].

A Graph G is a triple consisting of a vertex set V (G), an edge set E(G), and a relation that associates with each edge two vertices (not necessarily distinct) called its endpoints [28].

This Paper mainly focuses on classification of combinatorial algorithms with respect to graph algorithms. Also it consists of various real world graphs and the application of graph and graph algorithms in the areas like network routing, information security, vulnerability analysis, storage of data and Coding and Information Theory.

The work is organized as follows: section II contains algorithms and their classification based on graph. Section III and section IV depicts some real world graphs. The detailed applications of graph algorithms in real world have been given in section V. Finally, section VI concludes the article.

II. ALGORITHM AND ITS CLASSIFICATION:

There are various ways of classify algorithm. We have classified algorithm in five different categories. The broad categories are algorithm by implementation, algorithm by design paradigm, algorithm by representation of data, algorithm by field of study and algorithm by complexity. They are in detail depicted as follows:

2.1 Algorithm by implementation:

The algorithm can be classified according to their basic principles.

2.1.1. Recursive or iterative:

© 2018, IJCSE All Rights Reserved

Recursive algorithm repeatedly calls itself until a certain condition (also called termination condition) matches. Iterative algorithm use repetitive constructs like loops. Some problems are better suited for one implementation or some for other [3] [4].

2.1.2. Serial, Parallel or Distributed:

This classification purely depends on the computer architecture. Computers that execute one instruction of an algorithm at a time are called serial computers. Algorithm designed for such environment is called serial algorithm [5]. Parallel algorithm takes the advantages of computer architecture where several processors can work on a problem at the same time and distributed algorithm utilize multiple machines connected with a network [6]. In both parallel and distributed algorithm, problems are divided into sub problems and pass them to several processors or machines [7].

2.1.3. Deterministic or Non Deterministic:

The output for the same input is always the same irrespective of place and time for deterministic algorithm; e.g. sorting algorithm, searching algorithm, finding minimum spanning tree in a weighted connected graph [8]. In nondeterministic algorithm different outputs can be obtained with the same set of inputs while using the same algorithm, e.g. Travelling Salesman Problem [9].

2.1.4. Exact or Approximate:

An exact algorithm [10] that finds the exact or true solution while next one finds the solution close to the true solution (also called optimum solution) [11].

2.2. Algorithm by Design Paradigm:

Some common design paradigms, each of them different from other are described below:

2.2.1 Exhaustive Search:

This method tries every possible solution to see which the best is. Like eight queens [12] puzzle all possible arrangements of 8 pieces on the 64-square chess board will be examined and for each arrangements check whether each piece (queen) can attack another. But there are 64_{C_8} or 4, 426, 165, 368 possible arrangements but only 92 possible solutions.

2.2.2. Divide and Conquer:

A divide and conquer algorithm repeatedly reduces the instance of the problem to one or smaller instances of sub problems until the instances are small enough to solve the whole problem. It is a top down approach [13].

2.2.3. Dynamic Programming:

It solves the problem by breaking it down to simpler sub problems, solving each sub problem just once and storing their solutions ideally using a memory based data structure. When next time the same sub problem occurs, instead of recomputing their solutions, one simply looks into previously computed solution. This is called “memorization”. This is a bottom up approach, e.g. Travelling Salesman Problem [14].

2.2.4. Transform and conquer:

This technique solves the problem by transforming it into a better known problem. Of course, the goal is to find the reducing algorithm whose complexity will not surpasses the resulting reduced algorithm [15].

2.2.5. Search and Enumeration:

This algorithm tries to find items with specified properties among a collection of items that are coded into computer programs, e.g. finding Hamiltonian Circuits of a graph etc. Enumeration counts the number of that specified items, like-counting Hamiltonian Circuits of a graph [16]. This can be solved using Branch and Bound technique and Backtracking technique.

2.2.6. Randomized Algorithm:

This algorithm makes some choices randomly in each step. Mainly this type of algorithm is better where finding exact solutions are impractical [17].

2.2.7. Greedy Algorithm:

A greedy algorithm always makes the choice that looks best at the moment. It makes a locally optimal choice in the hope that this choices will lead to a globally optimal solution, e.g. Prim’s, Kruskal’s Algorithm for finding Minimum Spanning Tree [18].

2.2.8. Heuristic Algorithm:

Heuristic is a “Rule of Thumb”. This technique is designed for solving a problem more quickly when classic methods are too slow or for finding an approximate solution when classic methods fail to find any exact solution. Though it does not guarantee optimal solution, all that can be said for a useful heuristic is that it offers solutions which are good enough most of the time [19].

© 2018, IJCSE All Rights Reserved

2.2.9. Approximation Algorithm:

An approximation algorithm for a problem is an algorithm that generates approximate solution for the problem. Though it may fail to find optimal solution, it always provides some solution which is related to optimal solution by a bound of some kind [20].

2.2.10. Meta Heuristic:

A meta heuristic [21] is an iterative generation process which guides a subroutine heuristic by combining intelligently different concepts for exploring and exploiting search spaces using learning strategies to structure information in order to find efficiently near optimal solution. There are various types of meta heuristics like- Local Search or Global Search, Single Solution or Population based Solution, Hybridization or Memetic Algorithm, Parallel Meta Heuristic, Nature Inspired Meta Heuristic.

2.3. Algorithm by Representation of data:

By representation of data [2], algorithm can be classified into two categories namely.

2.3.1. Continuous Algorithm:

This algorithm operates on data that represents the continuous quantities. The data is represented by discrete approximations. These problems arise in numerical analysis.

2.3.2: Combinatorial Algorithm:

This algorithm is the main concern of our studies. The detailed description of this is introduced in the previous section. Mainly some questions can arise while studying combinatorial algorithms.
i) **Existence:** Are there any arrangements P that conform to the pattern? e.g. Subset Sum Problem, finding Proper k-coloring of a graph.
ii) **Construction:** If so, can such a P be found quickly? e.g. generating all permutations, generating all combinations, generating all Set partitions, generating all trees
iii) **Enumeration:** How many different arrangements P exists? e.g. Compress representation to generate all such patterns quickly like Binary Decision Diagrams.
iv) **Generation:** Can all arrangements P_1, P_2,… be visited systematically? e.g. counting number of ones in a Bit string.
v) **Optimization:** What arrangements maximize or minimize f(P), given an objective function f? e.g. Travelling Salesman Problem, Bin Packing Problem.

2.4: Algorithm by Field of Study:

Every field has its own problems and needs efficient algorithms. Fields may overlap with each other and algorithm developed for one field may improve the solution of another field. There are some classes of algorithms like Searching Algorithms, Sorting Algorithms, Graph Algorithms, Medical Algorithms, Machine Learning, Cryptography, Data Compression Techniques etc. that are applicable for various field of study.

2.5. Algorithm by Complexity:

By complexity algorithm can classified mainly in two problems.

2.5.1. Undicidable Problem:

Problems that do not have any known time complexity are called undecidable problems; e.g. Halting Problem. It tells that whether a given program (A) will terminate for a given input (B), i.e. whether can we have a program which when supplied with A and B will tell us in finite amount of time that A terminate for the input B. Goldbach’s Conjecture that tells that every even integer greater than two is sum of two prime integers. It is unsolvable because we have infinite number of integers. Other problems are reachability problem on infinite graph [22].

2.5.2. Decidable Problem:

Decidable problems mainly have known time complexity. They are of three types.

2.5.2.1. Decision Problems:

In this class of problem the output is either ‘yes’ or ‘no’. For example, given two numbers x and y, does x evenly divide y?
Two important classes of decidable problems are P and NP.

2.5.2.1.1. Class P:

P is the class of decision problem that can be solved in polynomial time algorithm or in the other have deterministic polynomial time algorithms, e.g. sorting, searching, shortest path finding etc.

2.5.2.1.1.2. Class NP:

This is the class of decision problems solvable by non-deterministic algorithms in polynomial time.
This problem can have mainly two classes:

2.5.1.2.1 Class NP-Complete:
A decision problem C is NP-Complete if
i. C is in NP and

ii. Every problem in NP can be polynomial time reducible to C.
There are lots of problems in this class like Graph 3-colorability problem, finding all spanning in a graph, finding maximum clique of a graph etc [23].

2.5.2.1.1. Class NP Hard:

A decision problem C is NP-Hard when for any problem L in NP, there is a polynomial time reduction from L to C. Halting problem depicted above belongs to NP-Hard class [24].

2.5.2.1.2. Counting Problem:

The output of this class of problem is a natural number [25]. For example, how many distinct factorization of a given number are there?

2.5.2.1.3. Optimization Problem:

This class of algorithms optimizes some objective function based on the problem instance. For example, finding minimum spanning tree in a weighted connected graph is an optimization problem. For every optimization problem, we have a corresponding decision and counting problem [26].
There exist various subfields of combinatorics like Enumerative Combinatorics, Analytic Combinatorics, Partition Theory, **Graph Theory**, Design Theory, Finite Geometry, Order Theory, Metroid Theory etc. Among them in this paper we will mainly emphasize on Graph Theory and its various real life applications.

III. GRAPH

Many real world problems can be formulated in terms of graph by taking it as a mathematical tool such that solving the later problem can give a suitable solution to the former one.For instance, the psychologist Lewin proposed that the „life space" of a person can be modeled by a planar graph, in which the faces represent the different environments [27]. As observed by D.E. Knuth [2] graph theoretical terminology and graph theorist are numerically comparable at this time. The field graph theory [28] started its journey from the problem of "Koinsberg bridge" in 1735.Graph algorithms can be treated as unified solution approach in many classical and modern application areas. The main concern is to design and adapt the art to various and numerous areas of real life industrial and engineering problems.

IV. EXAMPLES OF SOME REAL WORLD GRAPHS [29]

4.1. Social Graph:

A representation of the interconnection of relationships in an online social network, where nodes are actors and edges are relationship between them.

4.2. Call Graph:

A graph used to model calls where each telephone number is represented by a vertex and each telephone call is represented by an edge. When we are interested in who calls whom then we can use directed multigraph, otherwise undirected graph can be used.

4.3. Biological Network:

Graph representing protein and protein interactions.

4.4. Ecological Network:

A representation of the biotic interactions in an ecosystem, in which species (nodes) are connected by pairwise interactions (links).

4.5. Web Graph:

Represented by a directed graph, in which the vertices represent web pages and directed edges represent links from one page to another.

4.6. Chemical Network:

A representation of atoms and their bonding, where vertices represent atoms and edges bonds.

4.7. Program Flow Graph:

Graph representation of computation and control flow in the program.

4.8. Utility Graph:

Three Utilities Problem:

Suppose, there are three cottages on a plane and each needs to be connected to the gas, water, and electricity companies. Without using a third dimension or sending any of the connections through another company or cottage, is there a way to make all nine connections without any of the lines crossing each other?
This problem can be map to a complete bipartite graph K3,3 called Utility Graph and can be used to show that the above problem is unsolvable as the graph is non-planer.

4.9. Roadmaps:

Graphs can be used to model roadmaps where vertices represent intersections and edges represent roads. Undirected edges represent two-way roads; directed edges represent one way roads etc.

V. APPLICATIONS OF GRAPH AND GRAPH ALGORITHMS

5.1. Computer Network:

Computer networks are extremely popular in today's life. In computer networks nodes are connected to each other via links. This final network of nodes forms a graph.

5.1.1. Routing:

In computer network graph is used to form a network of nodes and enable efficient packet routing in the network. This includes finding the shortest paths between the nodes, analyze the current network traffic and find fasted root between the nodes, finding cost efficient route between the nodes. Standard algorithms such as Dijkstra's algorithm, Bellman-Ford algorithm are used to in the various ways with graph to find the solutions [30].
Finding QoS aware optimal route in Mobile ad hoc network (MANET) is very difficult due to dynamic topology, limited resources, and limited energy of nodes. S. Radhakrishnan et al. [31] proposed a distributed algorithm that adapts to the topology by utilizing spanning trees in the regions where the topology is stable, and resorting to an intelligent flooding-like approach in highly dynamic regions of the network. In [32,33,34,35,36,37] author proposed some nature inspired swarm based intelligent QoS aware routing protocols for MANET and theoretically showed some improvements over the existing QoS aware routing protocols.
Besides this, various graph theoretic approaches can be used to solve the problems of power consumption, coverage and congestion of wireless sensor networks [38].
Graph searching [39] becomes very popular in recent days especially in the field of social network. But often symmetries of graphs (often called graph automorphism) can complicate combinatorial searching algorithms. Such obstacles can often be removed by detecting and breaking symmetries early [40, 41].

5.1.2. Information Security:

Providing Information Security is the practice of defending information from unauthorized access, use, disclosure, disruption, modification etc. It is a general term that can be used regardless of the form the data may take (electronic, physical etc.) [42]. There can be many facets of information security. This work mainly concerned with information security using "proper designing of Ciphers", "cryptography', "steganography", "zero knowledge proof" protocol, "Anomaly Detection", "Graph Anonymization" etc.

5.1.2.1. Security using Cryptography:

Cryptography is the art of codifying messages, so that they become unreadable to provide information security [43]. The word cryptography comes from the Greek word "Kryptos", that means hidden, and "graphikos" which means writing. It is the transformation of readable and understandable data into a form which cannot be understood in order to secure the data [44]. So it is very necessary to design an efficient encryption algorithm. An encryption algorithm is computationally secure if it cannot be intruded with the standard resources. Graph can be used in encryption and decryption [45]. In [46], graph is used as the encryption key. The concept of Graph Automorphism can be used to design a constraint based cryptographic protocol. So, authors have designed a graph based modified secure classical Data Encryption Standard algorithm (GMDES) [47] based on partial symmetric key. The advantages of this algorithm are (a) it is not fully depended on secret key and produces different cipher text by applying same key on the same plain text which can prevent various types of malicious attacks, (b) if intruder can identify the secret key then also due to the huge search space of the key it is practically not possible to decrypt the required cipher text, (c) it can prevent Replay Attack, Chosen Cipher text Attack, Cipher text only Attack, Chosen Plain text Attack, Brute force Attack, Man-in-the-middle Attack etc., (d) in this algorithm a secure mapping table is used which is encrypted with senders private key, so at the time of decryption only authenticated receiver can decrypt the required cipher text, (e) our algorithm requires less encryption time than classical DES algorithm but the decryption time is depends on the receiver.

5.1.2.2. Security using Steganography, Watermarking and Fingerprinting:

The word "Steganography" comes from the Greek words "Steganos" meaning covered, concealed, or protected and "Graphein" meaning writing [48]. So, this is an art of concealing a file, message, image or video within another file. In recent days, graph theoretic approach to steganography has become very popular [49, 50]. Here, a graph is constructed from the cover data and the secret message and ultimately the steganographic embedding problem is reduced to the well known combinatorial problem of finding a maximum matching in graph. Steganography can be used for digital watermarking, where a message (being simply an identifier) is hidden in an image so that its source can be tracked or verified.

Graph theoretic approach can be used to design software watermark in robust fashion [51]. This method works with control/data flow graphs and uses abstractions, approximate

© 2018, IJCSE All Rights Reserved

k-partitions, and a random walk method to embed the watermark. Graph based IP protection schemes are also classical [52]. This scheme uses a generic graph corresponding to a digital system design and watermarking of the graph and its encryption are achieved using a new linear feedback shift register (LFSR)-based locking scheme. Various Constraint based techniques have been used in designing watermarking Intellectual Properties (IP), e.g. graph coloring based IP protection scheme [53], SAT based IP protection scheme [54] etc. In another point of view, there are some graph based algorithms that produces same watermarked graph with the different messages or signatures. This violates the uniqueness of the watermark [55].

Graph theoretic approach of software Fingerprinting are there [56] which is a form of watermarking in which an individualized mark is embedded into a copy of the media.

5.1.2.3. Security using Zero Knowledge Protocol:

Zero Knowledge Protocol is an interactive proof system where one party (Prover) proves a statement to another party (Verifier) without yielding anything beyond the validity of assertion. Now-a-days, communication protocols are vulnerable to impersonation attack and the performance of the protocol degrades because without proper security, any time an eavesdropper or an intruder may listen in on the communication. Zero Knowledge protocol tries to cope with this type of situation. Graph Isomorphism [57], 3-coloring Problem [58], Hamiltonian Cycle Problem [59] etc. can be used to design efficient Zero Knowledge Protocol [60].

5.1.2.4. Security using Anomaly Detection:

A thing, situation etc. that is different from what is normal or expected is called Anomaly. Graph Anomaly detection is analogous to finding intentional or unintentional ambiguity in the data patterns represented as graph. Since Graph based Anomaly detection is an open problem, authors concentrated mainly on existing techniques to find the truth value of the real life problems. In work [61] anomaly detection methods for detecting anomalous data where data represented as graphs are depicted in a nutshell and it contains a short review of recent existing graph based anomaly detection methods and its outcomes. Also some real world applications of graph based anomaly detection methods have been given with future directions to improve the approach of detecting anomalies in data in short.

5.1.2.5. Security using Graph Anonymization:

The Graph Anonymization problem can be stated as, given a graph G, asks for the k-degree anonymous graph that stems from G with the minimum number of graph modification operations. It is a well known problem [62] and the technique [63] applied for social network graphs can help in privacy preserving network publication.

5.1.3. Vulnerability Analysis:

In communication network a greater degree of stability or less vulnerability is very much required for better communication. A network can be represented as graph where the node represents the communicating entity and the edges represents the communication links between them. If some of the network nodes disrupted then the effectiveness of the network becomes very less and it becomes more vulnerable. As a consequence there exits some graph parameters which can quantify the vulnerability of the network. There are many graph theoretic parameters for measuring vulnerability of network like connectivity, integrity, toughness, binding number, tenacity etc. [64]. Among them, "Integrity" [65] measures of the network which is one of the most important graph parameters for measuring vulnerability. The integrity I(G) of the graph G is given by I(G) = $\min_{S \subset V(G)} \left(|S| + m(G-S) \right)$ where m(G) denotes the maximum order of a component of a graph G. But measuring graph Integrity is a NP-Complete problem.

5.1.4. Graph of Things:

The Internet of Things (IoT) with billions of connected devices has been generating an enormous amount of data every hour. Connecting every data item generated by IoT to the rest of the digital world to turn this data into meaningful actions will create new capabilities, richer experiences, and unprecedented economic opportunities for businesses, individuals, and countries. However, providing an integrated view for exploring and querying such data at real-time is extremely challenging due to its Big Data natures: big volume, fast real-time update and messy data sources. To address this challenge a unified intsegrated and live view for heterogeneous IoT data sources using Linked Data, called the Graph of Things (GoT) has been introduced in [66]. GoT is backed by a scalable and elastic software stack to deal with billions of records of historical and static datasets in conjunction with millions of triples being fetched and enriched to connect to GoT per hour in real time.

5.2. Data Storage:

5.2.1. Graph Database:

Gathering huge amounts of complex information (data and knowledge) is very common nowadays. These calls for the necessity to represent, store and manipulate complex information (e.g. detect correlations and patterns, discover explanations, construct predictive models etc.). Furthermore, being autonomously maintained, data can change in time or

© 2018, IJCSE All Rights Reserved

even change its base structure, making it difficult for representation systems to accommodate these changes. Current representation and storage systems are not very flexible in dealing with big changes and also they are not concerned with the ability of performing complex data manipulations of the sort mentioned above. On the other hand, data manipulation systems cannot easily work with structural or relational data, but just with flat data representations. So, to bridge the gap between the two, a new type of database structure called Graph Databases (GDB) has been introduced based on a natural graph representation. In [67] author shows that Graph Databases are able to represent as graphs any kind of information, naturally accommodate changes in data, and they also make easier for Machine Learning methods to use the stored information.

5.2.2. Graph Partitioning in database design:

Graph partitioning is a technique to distribute the whole graph data as a disjoint subset to a different device. The need of distributing huge graph data set is to process data efficiently and faster process of any graph related applications. A good graph partitioning algorithm always aims to reduce the communication between machines in their distributed environment and distribute vertices roughly equal to all the machines. It is used for efficient query design of graph database [68].

5.2.3. Graph Compression:

Nowadays, graphs are everywhere. Examples are social networks, communication networks, biological networks and World Wide Web. With the development of technology and the discovery of new knowledge, the size of these graphs becomes larger and larger. Some graphs contain millions or even billions of nodes and edges. Such a bulky graph presents new challenges to graph mining. One issue is that a huge graph may severely restrict the application of existing pattern mining technologies. Additionally, directly visualizing such a large graph is beyond our capability. To tackle these large number of nodes and edges compressed graph comes in the scene which conserve the characteristics of the original graph. It can be easily visualized and the goal of compressing a graph is to make the high-level structure of the graph easily understood. Therefore, informative graph compression techniques are required and have wide application domains [69,70]. Other facet is the impossibility to reduce the complexity of the algorithms for graph query answering (e.g., graph pattern matching) has pushed many researchers to devise alternative solutions for improving performance. One of them is to reduce the size of the input by transforming the original graph into a smaller graph.

5.2.4. Storage Space Minimization:

In the age of Big Data an open problem and an important research aspect in recent days is to store huge volume of data in a minimum storage space by utilizing available resources properly. As there is a close relationship between this problem and the famous NP-Hard combinatorial optimization problem, authors [71] try to design an heuristic algorithm to cope with the one dimensional offline bin packing problem. The goal is to assign each item (data) to bins (available storage space), such that total weight of the items in each bin does not exceed the capacity C and the number of bins used for packing all items is minimized. This problem also has several applications in real world, among them resource and storage space minimization is one facet. Some formulations of real world storage minimization problem usin-g Bin Packing Problem are as follows: i) Placing computer files with specified size into the identical disk with same capacity with constrained that each file must be entirely on one disk. The objective is to minimize the number of disks needed for the set of files. This can be formulated using Bin Packing Problem where items are files, disks are bins and disk capacity is the bin capacity which is fixed. The problem is to minimize the number of bins. ii) Server Consolidation is an approach to the efficient usage of computer server resources in order to reduce the total number of servers or server locations that an organization requires. In this case, existing servers can be treated as items, resource utilizations are item sizes, bins are destination servers and the bin capacity is the utilization threshold of the destination servers. The goal is to minimize the destination servers and maximizing resource utilization. With one resource the problem is same as one dimension Bin Packing Problem.

In another view since in most of the cases minimized circuit should be less expensive to build than the unminimized version and minimized networks will be faster (have fewer propagation delays) than unminimized networks. So it is very much necessary to represent the Boolean function in a compressed form such that the circuit takes less space i.e in minimized form. Keeping the need in mind authors has considered the two-level logic minimization problem which is a classical problem. The problem is, there is a Set $U=\{u_1,u_2,\ldots u_n\}$ of variables and the instance contains complete truth table, i.e. disjoint sets A and B $\{1,0\}^n$, such that A B = $\{1,0\}^n$. The objective is to minimize the given Sum of Product (SOP) function such that for all truth assignments in A the outcome of the function is true and for all false assignments in B the outcome is false. In the field of optimization problem heuristic technique is one of the important parts. Authors have developed a heuristic algorithm [72] for Boolean SOP function minimization taking the advantages of implicit representation of the graphical data structure Binary Decision Diagram (BDD). A BDD is the compact and canonical representation of Boolean functions and with proper variable ordering the size of the BDD increases linearly with the number of *variables. In this method, at first a heuristic has been designed to minimize the number of disjoint cubes by proper reordering of the

© 2018, IJCSE All Rights Reserved

variables and minimizing number of 1-paths of BDD. But it does not necessarily give the minimum cover. Thus, a minimum cover with the help of divide and conquer technique has been proposed to produce a near optimal solution.

5.3. Coding and Information Theory:

Information is nothing but a sequence of symbols. The symbols themselves have no possible meanings and all that assumed is that they can be uniquely recognized. “Sending information from here to there” (transmission of information) is exactly the same as “Sending information now to then” (storage of information). Both these situations occur when handling with the information. The essential thing is to encode the information for efficient storage as well as reliable recovery in the presence of “noise” [73]. To protect the information from unauthorized access security of information is needed. For maintaining consistency and accuracy, information integrity is necessary and as symbols are uniquely represented by codes, there is a need for storage of codes in minimal space to increase the efficiency.

In Coding theory point of view graph can be used to design linear block codes. The circuit matrix (or cut-set matrix) of a linear graph ‘G’ generates a binary linear code of distance ‘d’ and length n-an (n, d) code-where n is the number of branches in G and d is minimum number of branches in a circuit (or a cut-set) of G [74]. Such code is referred to as graph theoretic codes. The main problem in coding theory is to find good code (n, M, d) [n=total length of a code, M= efficiency, d=distance] that has small n (for fast transmission of messages), large M (to increase efficiency) and large d (to correct many errors).This is very useful for post quantum cryptography.

For probabilistic decoding of LDPC and turbo codes in belief network Factor graph is used. The bipartite graph can also be used in Query Log Analysis, which is used for improve search engine capability [75].

Generally in modern coding theory Bipartite graph is used for decoding the code words, which are, receives from the channel. For example Factor graph and Tanner graph is manly used for decoding the code. Tanner graph is an application of bipartite graph so, vertices are divided into two parts in which first bipartition represent the digit of code word, and the other side bipartition represent the combination of digits that are expected to sum zero in a code word without errors [76].

Network coding [77] is a networking technique in which transmitted data is encoded and decoded to increase network throughput, reduce delays and make the network more robust. In network coding, algebraic algorithms are applied to the data to accumulate the various transmissions. Graphs are also used in this field also.

VI. CONCLUSION

This article gives a detailed classification of combinatorial algorithms specially based on graph and graph algorithms. Also to prove the importance of the graph based algorithms some real world applications of these algorithms has been brought into the limelight. In future author will try to bring more applications of graph algorithms that may help the human being.

REFERENCE

[1]. Reingold, E. M., Nievergelt, J., & Deo, N. (1977). Combinatorial algorithms: theory and practice. Prentice Hall College Div.

[2]. Knuth, D. E. (2005). The Art of Computer Programming. Fascicle 4A-Generating All Trees, vol. 4.

[3]. Chekuri, C., & Pal, M. (2005, October). A recursive greedy algorithm for walks in directed graphs. In Foundations of Computer Science, 2005. FOCS 2005. 46th Annual IEEE Symposium on (pp. 245-253). IEEE.

[4]. Even, S. (2011). Graph algorithms. Cambridge University Press.

[5]. Reif, John H. (1985). Depth-first search is inherently sequential. Information Processing Letters. 20 (5). doi:10.1016/0020-0190(85)90024-9.

[6]. Quinn, M. J., & Deo, N. (1984). Parallel graph algorithms. ACM Computing Surveys (CSUR), 16(3), 319-348.

[7]. Erciyes, K. (2018). Distributed Graph Algorithms. In Guide to Graph Algorithms (pp. 117-136). Springer, Cham.

[8]. Graham, R. L., & Hell, P. (1985). On the history of the minimum spanning tree problem. Annals of the History of Computing, 7(1), 43-57.

[9]. Hoffman, K. L., Padberg, M., & Rinaldi, G. (2013). Traveling salesman problem. In Encyclopedia of operations research and management science (pp. 1573-1578). Springer US.

[10]. Sewell, E. C. (1996). An improved algorithm for exact graph coloring. Cliques, coloring, and satisfiability: second DIMACS implementation challenge, 26, 359-372.

[11]. Even, G., Naor, J., Rao, S., & Schieber, B. (1999). Fast approximate graph partitioning algorithms. SIAM Journal on Computing, 28(6), 2187-2214.

[12]. Golomb, S. W., & Baumert, L. D. (1965). Backtrack programming. Journal of the ACM (JACM), 12(4), 516-524.

[13]. Wang, X., Wang, X., & Wilkes, D. M. (2009). A divide-and-conquer approach for minimum spanning tree-based clustering. IEEE Transactions on Knowledge and Data Engineering, 21(7), 945-958.

[14]. Bellman, R. (1962). Dynamic programming treatment of the travelling salesman problem. Journal of the ACM (JACM), 9(1), 61-63.

[15]. Levitin, A., & Papalaskari, M. A. (2002, February). Using puzzles in teaching algorithms. In ACM SIGCSE Bulletin (Vol. 34, No. 1, pp. 292-296). ACM.

[16]. Garey, M. R., Johnson, D. S., & Tarjan, R. E. (1976). The planar Hamiltonian circuit problem is NP-complete. SIAM Journal on Computing, 5(4), 704-714.

[17]. Gazit, H. (1986, October). An optimal randomized parallel algorithm for finding connected components in a graph. In Foundations of Computer Science, 1986., 27th Annual Symposium on (pp. 492-501). IEEE.

[18]. Greenberg, H. J. (1998). Greedy algorithms for minimum spanning tree. University of Colorado at Denver.

[19]. Kernighan, B. W., & Lin, S. (1970). An efficient heuristic procedure for partitioning graphs. The Bell system technical journal, 49(2), 291-307.

[20]. Amestoy, P. R., Davis, T. A., & Duff, I. S. (1996). An approximate minimum degree ordering algorithm. SIAM Journal on Matrix Analysis and Applications, 17(4), 886-905.

[21]. Bullnheimer, B., Hartl, R. F., & Strauss, C. (1999). An improved ant System algorithm for thevehicle Routing Problem. Annals of operations research, 89, 319-328.

[22]. Henzinger, M. R., Henzinger, T. A., & Kopke, P. W. (1995, October). Computing simulations on finite and infinite graphs. In Foundations of Computer Science, 1995. Proceedings., 36th Annual Symposium on (pp. 453-462). IEEE.

[23]. Garey, M. R., Johnson, D. S., & Stockmeyer, L. (1976). Some simplified NP-complete graph problems. Theoretical computer science, 1(3), 237-267.

[24]. Alekseev, V. E., Boliac, R., Korobitsyn, D. V., & Lozin, V. V. (2007). NP-hard graph problems and boundary classes of graphs. Theoretical Computer Science, 389(1-2), 219-236.

[25]. Mathon, R. (1979). A note on the graph isomorphism counting problem. Information Processing Letters, 8(3), 131-136.

[26]. Yuan, S. Y., & Kuo, S. Y. (1998). A new technique for optimization problems in graph theory. IEEE Transactions on Computers, 47(2), 190-196.

[27]. LEWIN, K (1936). Principles of Topological Psycology. Mc-Graw-Hill, New York.

[28]. Deo, N. (2004). "Graph theory with applications to engineering and computer science." PHI.

[29]. Rosen, K. H. (2007). Discrete mathematics and its applications. Amc, 10(12), 824.

[30]. Madkour, A., Aref, W. G., Rehman, F. U., Rahman, M. A., & Basalamah, S. (2017). A survey of shortest-path algorithms. arXiv preprint arXiv:1705.02044.

[31]. Radhakrishnan, S., Racherla, G., Sekharan, C. N., Rao, N. S., & Batsell, S. G. (1999). DST-a routing protocol for ad hoc networks using distributed spanning trees. In Wireless Communications and Networking Conference, 1999. WCNC. 1999 IEEE (Vol. 3, pp. 1543-1547). IEEE.

[32]. Sensarma, D., &Majumder, K. "A Comparative Analysis of the Ant Based Systems for QoS Routing in MANET." In Recent Trends in Computer Networks and Distributed Systems Security (pp. 485-496). Springer Berlin Heidelberg, 2012.

[33]. Sensarma, D., andMajumder, K."AMTR: The ANT Based QOS Aware Multipath Temporally Ordered Routing Algorithm for MANETs", AISC- 2013, CS & IT-CSCP 2013, pp. 389–396, 2014.

[34]. Sensarma, D., andMajumder, K. "An efficient ant based qos aware intelligent temporally ordered routing algorithm for manets." International Journal of Computer Networks & Communications (IJCNC), Vol. 5, No. 4, PP.189-203, Jul. 2013.

[35]. Sensarma, D., and Majumder, K. "HAQR: TheHierarchical ANT based QOS aware On-demandRoutingfor MANETS."WimoA- 2013,CS & IT-CSCP 2013, pp.193-202, 2013.

[36]. Sensarma, D., andMajumder, K. "A Novel Hierarchical Ant Based QoS aware Intelligent Routing Scheme for MANETs." International Journal of Computer Networks & Communications (IJCNC), Vol. 5, No. 6, PP.215-229, Nov. 2013.

[37]. Sensarma, D., andMajumder, K."IWDRA: An Intelligent Water Drop Based QoS-Aware Routing Algorithm for MANETs." Proceedings of the International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA) 2013. Springer International Publishing, 2014.

[38]. Rajaram, V., Kumatharan, N., & Srividhya, S. Study on Applications of Graph Theory in Wireless Sensor Networks.

[39]. Mears, C., De La Banda, M. G., & Wallace, M.,"On implementing symmetry detection." Constraints, 14(4), 2009, 443-477.

[40]. Albertson, Michael O., and Karen L. Collins. "Symmetry breaking in graphs." Electron. J. Combin 3, no. 1 (1996): R18.

[41]. Harary, F. R. A. N. K. "Methods of destroying the symmetries of a graph." Bull. Malasyan Math. Sc. Soc 24, no. 2 (2001): 183-191.

[42]. Sensarma, D., and Sarma, S. S. A Unified Framework for Security and Storage of Information. International Journal of Advance Engineering and Research Development, Vol. 2, Issue 1, PP.197-203, 2015.

[43]. Kahate, A. Cryptography and network security. Tata McGraw-Hill Education, 2013.

[44]. Ustimenko, V. (2001). CRYPTIM: Graphs as tools for symmetric encryption. In Applied Algebra, Algebraic Algorithms and Error-Correcting Codes. Springer Berlin Heidelberg, PP. 278-286, 2001.

[45]. Ustimenko, V. "On Graph-Based Cryptography and Symbolic Computations."Serdica Journal of Computing 1, no. 2 (2007): 131-156.

[46]. Sensarma, D., Banerjee, S., and Basuli, K. A New Scheme for Key Exchange. International Journal of Modern Engineering Research (IJMER), ISSN (Online): 2249-6645, 2(3), 2012.

[47]. Sensarma, D., and Sarma, S. S. GMDES: A GRAPH BASED MODIFIED DATA ENCRYPTION STANDARD ALGORITHM WITH ENHANCED SECURITY. International Journal of Research in Engineering and Technology, eISSN: 2319-1163, Vol. 3, Issue. 3, PP. 653-660, Mar-2014.

[48]. Shih, F. Y. Digital watermarking and steganography: fundamentals and techniques. CRC Press, 2007.

[49]. Hetzl, S., and Petra, M. "A graph–theoretic approach to steganography." In Communications and Multimedia Security, pp. 119-128. Springer Berlin Heidelberg, 2005.

[50]. Wu, H., Wang, H., Zhao, H., & Yu, X. (2015). Multi-layer assignment steganography using graph-theoretic approach. Multimedia Tools and Applications, 74(18), 8171-8196.

[51]. Venkatesan, R., Vazirani, V., and Sinha, S. "A graph theoretic approach to software watermarking." In Information Hiding, pp. 157-168. Springer Berlin Heidelberg, 2001.

[52]. Halder, R., Dasgupta,P.,Naskar,S., and Sarma., S. S. "An Internet-based IP Protection Scheme for Circuit Designs using Linear Feedback Shift Register-based Locking."Engineering Letters 19, no. 2 (2011): 84.

[53]. Qu, G., &Potkonjak, M.,"Hiding signatures in graph coloring solutions." In Information Hiding (pp. 348-367). Springer Berlin Heidelberg, (2000, January).

[54]. Kahng, A. B., Lach, J., Mangione-Smith, W. H., Mantik, S., Markov, I. L., Potkonjak, M., & Wolfe, G., "Constraint-based watermarking techniques for design IP protection."Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on, 20(10), 1236-1252, 2001.
[55]. Sensarma, D., & Sarma, S. S. (2015). Data Hiding using Graphical Code based Steganography Technique. arXiv preprint arXiv:1509.08743.
[56]. Collberg, Christian S., Thomborson, C. and Gregg M. Townsend. "Dynamic graph-based software fingerprinting." ACM Transactions on Programming Languages and Systems (TOPLAS) 29, no. 6 (2007): 35.
[57]. G. Brassard, C. Crepeau. "Non-Transitive Transfer of Confidence: A PerfectZeroKnowledge Interactive Protocol for SAT and Beyond." Proc. of the 27th Annual Symp. on Foundations of Computer Science: 188-195, 1986.
[58]. M. Blum. "How to Prove a Theorem So No One Else Can Claim It." Proceedings of the International Congress of Mathematicians: 1444-1451, 1986.
[59]. D. Lapidot, A. Shamir. "A one-round, two-prover, zero-knowledge protocol for NP." Combinatorica, 15(2): 203-214, June, 1995.
[60]. Horan, V., & Gudaitis, M. (2011). Investigation of Zero Knowledge Proof Approaches Based on Graph Theory. AIR FORCE RESEARCH LAB ROME NY INFORMATION DIRECTORATE.
[61]. Sensarma, D., & Sarma, S. S. (2015). A survey on different graph based anomaly detection techniques. Indian Journal of Science and Technology, 8(31).
[62]. Liu, K., and Terzi, E. "Towards identity anonymization on graphs." InProceedings of the 2008 ACM SIGMOD international conference on Management of data, pp. 93-106. ACM, 2008.
[63]. Zou, L., Chen, L., &Özsu, M. T. "K-automorphism: A general framework for privacy preserving network publication." Proceedings of the VLDB Endowment, 2(1), 946-957, (2009).
[64]. Moazzami, D. (2016). Towards a measure of vulnerability, tenacity of a Graph. Journal of Algorithms and Computation, 48(1), 149-154.
[65]. Li, F., & Li, X. (2004). On the integrity of graphs. In Parallel and Distributed Computing and Systems (Vol. 16, pp. 577-582).
[66]. Le-Phuoc, D., Quoc, H. N. M., Quoc, H. N., Nhat, T. T., & Hauswirth, M. (2016). The Graph of Things: A step towards the Live Knowledge Graph of connected things. Web Semantics: Science, Services and Agents on the World Wide Web, 37, 25-35.
[67]. Angles, R., & Gutierrez, C. (2008). Survey of graph database models. ACM Computing Surveys (CSUR), 40(1), 1.
[68]. Meyerhenke, H., & Gehweiler, J. (2010). On dynamic graph partitioning and graph clustering using diffusion. In Dagstuhl Seminar Proceedings. Schloss Dagstuhl-Leibniz-Zentrum fÃ¼r Informatik.
[69]. Zhou, F. (2015). Graph compression. Department of Computer Science and Helsinki Institute for Information Technology HIIT, 1-12.
[70]. Chierichetti, F., Kumar, R., Lattanzi, S., Mitzenmacher, M., Panconesi, A., & Raghavan, P. (2009, June). On compressing social networks. In Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 219-228). ACM.
[71]. Sensarma, D., & Sarma, S. S. (2017). A Graph Theoretic Approach for Minimizing Storage Space using Bin Packing Heuristics. INTERNATIONAL JOURNAL OF ADVANCED COMPUTER SCIENCE AND APPLICATIONS, 8(2), 29-39.
[72]. Sensarma, D., Banerjee, S., Basuli, K., Naskar, S., & Sarma, S. S. (2012). On an optimization technique using Binary Decision Diagram. arXiv preprint arXiv:1203.2505.
[73]. Hamming, R. W. Coding and information theory. Prentice-Hall, Inc., 1986.
[74]. Jungnickel, D., & Vanstone, S. A. (1997). Graphical codes revisited. IEEE Transactions on Information Theory, 43(1), 136-146.
[75]. Tanner, R. M., Sridhara, D., Sridharan, A., Fuja, T. E., & Costello, D. J. (2004). LDPC block and convolutional codes based on circulant matrices. IEEE Transactions on Information Theory, 50(12), 2966-2984.
[76]. Luby, M. G., Mitzenmacher, M., Shokrollahi, M. A., & Spielman, D. A. (2001). Improved low-density parity-check codes using irregular graphs. IEEE Transactions on Information Theory, 47(2), 585-598.
[77]. Gadouleau, M., & Riis, S. (2011). Graph-theoretical constructions for graph entropy and network coding based communications. IEEE Transactions on Information Theory, 57(10), 6703-6717.

Authors Profile

Mr. Debajit Sensarma has received his B.Sc degree in Computer Science in the year 2009 from university of Calcutta, Kolkata, India and M.Sc degree in computer Science in the year 2011 from West Bengal State University, Kolkata, India. He obtained his M.Tech. degree in Computer Science and engineering from West Bengal University of Technology, Kolkata, India in the year 2013. He has also worked as INSPIRE Fellow (Govt. of India) in the department of Computer Science & Engineering, University of Calcutta, India for three years. He is presently working as an Assistant Professor in the Dept. of Computer Science in Vivekananda Mission Mahavidyalaya, Vidyasagar University, Purba Medinipur, West Bengal, India. He has published several papers in International journals and conferences.