Finding Cliques In Simulated Social Networks Using Graph Coloring Technique

اكتشاف المجموعات في الشبكات الاجتماعية الافتراضية باستخدام تقنية تلوين المخططات

Prepared By

Mohammad Hussein Alomari

Supervisors

Dr. Oleg Viktorov

Dr. Mohammad Malkawi

Thesis Submitted in Partial Fulfillment of the requirements for the

Degree of Master of Computer Science

Department Of Computer Science

Faculty of Information Technology

Middle East University

January, 2017

Authorization statement

I, Mohammad Hussein Ahmad Alomari, authorize the Middle East University to provide hard copies or soft copies of my Thesis to libraries, institutions or individuals upon their request.

Name: Mohammad Hussein Ahmad Alomari

Date: 21 /01 /2017

Signature:

اقرار تفويض

انا محمد حسين أحمد العمري افوض جامعة الشرق الاوسط بتزويد نسخ من رسالتي للمكتبات او المؤسسات او الهيئات او الافراد عند طلبها.

الاسم محمد حسين أحمد العمري

التاريخ : 2017/ 01/ 21

التوقيع :

Middle East University

Examination Committee Decision

This is to certify that the thesis entitled "Finding Cliques in Simulated Social Networks Using Graph Coloring Technique" was successfully defended and approved in 21 / 01 / 2017.

Examination Committee Members | **Signature**

Dr. Sharefa Murad (Internal Member & Chairman)

Assistant Professor, Department of Computer Science
Middle East University (MEU)

3/5/2017.

Dr. Faiz Al-Shrouf (Advisor)

Associate Professor, Department of Computer Science
Middle East University (MEU)
On Behalf Of Dr. Oleg Victorov

Dr. Mohammad Malkawi (Co Advisor)

Associate Professor, Department of Software Engineering
Jordanian University of Science and Technology (JUST)

Dr. Rizik Al-Sayyed (External Member)

Associate Professor, King Abdullah II School of Information Technology, Department of Business Information Technology
University of Jordanian ` (UJ)

ACKNOWLEDGEMENT

I would like to thank my God firstly then to thank the Middle East University which is anchored in its mission to the" knowledge power" and preparing it's students to be leaders and to be qualified whatever position they occupied.

My thesis advisors Dr Fayez Alshrouf at Middle East University, Dr Mohammad Malkawi at Jordan University of. Science and Technology, Dr Oleg Vectorov at Middle East University. To them I owe a great debt of thanks for their patience, motivation and sportiness, their offices always open whenever I ran into a trouble spot or had a question about my research or writing. They steered me in the right the direction whenever they thought I needed it.

To my examination committee, the chairman Dr. Sharefa Murad and, the member Dr. Rizik Al-Sayyed**,** thank you for your constructive remarks and support.

I would like to extend my gratitude to all of the staff in the faculty of information technology on their extensive efforts in the amelioration of the college and students

Finally, special recognition goes out to my parents, my family and my friends to support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them. Thank you.

Mohammad Alomari

DEDICATION

I dedicate my dissertation work to my family. A special feeling of gratitude to my loving parents, brothers and sistersI also dedicate this dissertation to my relatives and friends

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

GCP	Graph Coloring Problem
LDF	Largest Degree First i\Coloring
DIMACS	Discrete Mathematics and Theoretical Computer Science
CCI	Clique Conformance Measure
LDC	Largest Degree algorithm

Finding Cliques in Simulated Social Networks Using Graph Coloring Techniques

Prepared by Mohammad Hussein Alomari

Superviser Dr. Oleg Viktorov

Co Advisor Dr. Mohammad Malkawi

Abstract

In this thesis, the problem of finding cliques has addressed in large graphs such as social networks. The problem of finding all cliques in a graph is known to be NP problem. A new heuristic approach for finding cliques has provided. The new approach represents a graph coloring technique which is based on the using of Largest Degree Coloring algorithm, which colors a graph starting with the largest degree node in a particular graph. This proposed approach follows the largest cliques in a graph and moves to finding smaller cliques. The proposed approach with the algorithm for finding cliques in heuristic method and the exhaustive search algorithm has used as a reference.

A lot of experiments has been conducted in this study on different data sets, 72 experiments of them by applying the three algorithms on variants data sets generated by our java program, and five experiments data set taken from standard DIMACS benchmark and last data set consist of simulated social networks, in range between 10000 nodes and 50000 nodes. hopefully to apply the proposed approach in real social networks,

knowing that the exhaustive search may take relatively large time. the new approach based on graph coloring has achieved better complexity and detectability of cliques. However, the results of a proposed approach have shown that there is enhancement in time complexity than other two approaches that used in comparison to proposed algorithm and the number of detected cliques.

Keywords: Graph coloring problem, clique, clustering, social networks,

اكتشاف المجموعات في الشبكات الاجتماعية الافتراضية باستخدام تقنية تلوين المخططات

اعداد: محمد حسين العمري

المشرف: أد. أوليج فيكتوروف

المشرف المساعد: أد. محمد الملكاوي

الملخص

في هذه الأطروحة، وقد تم تناول مشكلة العثور على المجموعات المخطوطات الكبيرة مثل الشبكات الاجتماعية. ومن المعروف أن مشكلة العثور على جميع المجموعات في المخططات هو مسألة كثيرة حدود غير قطعية كاملة. وقد قدم نهجا جديدا يعتمد على الاسترشاد لإيجاد المجموعات. يمثل النهج الجديد تقنية تلوين المخططات الذي يقوم على استخدام خوارزمية البدء بالتلوين عند الراس ذو أكبر درجة، الذي ألوان الرسم البياني بدءا من عقدة أكبر درجة في رسم بياني معين. يتبع هذا النهج المقترح أكبر المجموعات في المخططات ويتحرك لإيجاد المجموعات الصغيرة. وقد استخدم النهج المقترح مع خوارزمية للعثور على المجموعات في الطريقة الاستدلالية وخوارزمية البحث شاملة كمرجع. وقد أجريت الكثير من التجارب في هذه الدراسة على مجموعات بيانات مختلفة و72 تجربة منها بتطبيق الخوارزميات الثلاثة على مخططات متباينة الحجم والكثافة الناتجة عن برنامجنا المكتوب بلغة جافا وخمسة مخططات من مجموعة البيانات المأخوذة من معيار ديماكس القياسي ومجموعة البيانات الأخيرة تتكون من شبكات اجتماعية افتراضية، تتراوح بين 10000 رأس و50000 رأس، ونأمل أن تطبيق النهج المقترح في الشبكات الاجتماعية الحقيقية، مع العلم أن البحث الشامل قد يستغرق وقتا كبيرا نسبيا. وقد حقق النهج الجديد القائم على تلوين المخططات أفضل وفت والكشف عن المجموعات. ومع ذلك، أظهرت نتائج النهج المقترح أن هناك زيادة في زمن التنفيذ من النهجين الآخرين التي استخدمت بالمقارنة مع النهج المقترحة وعدد المجموعات التي تم اكتشافها.

الكلمات المفتاحية: مسألة تلوين المخططات، المجموعة، إيجاد المجموعات(التجميع)، الشبكات الاجتماعية.

CHAPTER ONE

INTRODUCTION

1.1. Introduction

Graph theory concerns in studying graphs to formulate the real-life problems, the origin of graphs theory returns to mathematicians but after the existence of computers the graph theory becomes magnificent and core topic in computer science and increasingly it's significant as it is applied in a lot of computer fields and other areas of diverse sciences, where can be used for modeling many real-life problems.

Graph theory studies many topics such as paths and cycles, trees, Mazes and labyrinths and many others but graph coloring is considered as one of the best known, popular and extensively researched subject in the field of graph theory, due to having many applications, many fields and conjectures, which are still open and huge of studies and researches are underway by various computer researchers.

This research represents an analysis and utilization of what had been proposed by Alzubi, &Malkawi, (2015). In that paper, Alzubi et al (2015) proposed a new algorithm for solving the graph coloring problem and they used the Maximum Clique Conformance Measure (CCI) `for coloring graphs with the minimum number of colors to provide a solution for the exam scheduling problem. However, in this thesis, a new utilization of this algorithm has used for coloring graphs with minimum number of possible colors in a graph, also an analysis for the utilized algorithm has provided in terms of how we can detect cliques in undirected graph by using graph coloring technique and apply this new approach on simulated social networks.

Social network can be represented as a graph that consists of edges (friendships between two or more people) and nodes (personal profiles). A graph G is a pair (*N*, *E*) of a set of nodes *N* and a set of edges *E*. The edges are unordered pairs of the form $\{i, j\}$ where $i, j \in N$. Two nodes i and j are said to be adjacent if and only if $\{i, j\} \in E$, the degree of a node n is the number of nodes adjacent to *n* and is denoted by *deg*(*n*). Graphs can be classified depending on some criteria, these criteria related to the edges but not nodes, also these criteria and concepts include exitance or absent of edge, directed or undirected edges, weighted or un-weighted edges, next paragraphs hold the importance of this research.

Studying graphs acquired a special importance in computer science because of its ability to model many problems not just in computer science but beyond to many fields such as social sciences, math, medical sciences and. hard sciences. The philosophy of graphs states that there are many nodes and there are relations between them. A lot of applications with distributed elements and relations between these elements can be modeled as a graph, such as cities and roads, nodes represents a cities and roads represents by edges, another example in social networks, people represents by nodes and relationships among them represents edges.

the tremendous importance of this research due to widely speeding of social networks and how the people stay in connection with these networks, subsequently the social networks considered as repository of information about people, for example people with common interests and people who has friendship relations with each other, these groups is very important for administration of social networks, because the social networks is very tremendous network , consist of hundreds of millions participants with variant languages, ages, interests, regions, religions, hobbies and so many different of other attributes or follow some pages, comment on some topics, or repeats some words in his comments, these huge diversity of characteristics and attributes of social network's participants requires from these networks administrators to deal with the huge variants participants in lower cost and higher efficiency, and that could be achieved by creating tools with ability to find cliques and groups in simulated social networks according to specific attributes and deal with it instead of dealing with whole participants in social networks although the target recipients not exceed 10%.of all participants, hence sending advertisements, invitations or business activity to specific cliques will save a lot of time, cost and overloads.

Although the importance of social networks in our life but there are many problems in these social networks, due to it's huge size there is no algorithm can solve the clique's problem in reasonable time, although a lot of methods have been presented but no one has optimal solution, this research aims to present proposed solution to this complex problem.

1.2. BCKGROUND

This section contains background research and definitions of main concepts to give an over view about essential subjects of this research

graph coloring

General graph coloring is the process of labeling the graph's elements with special labels (traditionally called colors) subject to certain constraints. Node coloring is the process of coloring the nodes such that no two adjacent nodes share the same color. The graph coloring problem(GCP) is then to find a vertex coloring for an any graph using the minimum number of colors possible.

GCP is very important because it has many applications, some of them are planning and scheduling problems timetabling, map coloring and many other applications. Since GCP is a NP-hard problem, until now there are not known deterministic methods that can solved it in a polynomial time. So, non-deterministic algorithms have been built to solve this complex problem; one of them is Numeration algorithm that has been used on find cliques successfully. However, Numeration algorithm is not a complete algorithm and it not always gets the optimal solution. So, new heuristic algorithms have developed and efforts has focused on utilizing existing algorithms, which usually produces and accomplishes the coloring for any graph in a reasonable amount of time.

The clique is basic and an important concept in graph theory, also, called as a complete graph, it is defined as a graph where every node is adjacent to every other node, while the maximum clique in graph is that one which has the maximum number of nodes. The importance of cliques in graphs due to the analysis of cliques in graphs give us the analysis and behavior of that graphs, on the other side, analyzing any part of clique could derive to predict the rest members of clique, because the clique members commonly sharing a lot of attributes.

There are three measures of complexity classes, P, NP and NP-Complete problems. The P, NP and NP-Complete classes depend on two key factors related on these complex classes time and space, time is considered the absolutely most important factor in measuring the complicity of problems, the time required to solve some problem is the number of steps does it take to solve the problem, the other factor which is considered as less important than the time factor is space or memory and the space that memory takes to solve the problem. But in this research, has concentrated on time factor since the huge progress in manufacturing storage devices overcome the memory space problem.

However, finding cliques in graph seems to be difficult since this problem is considered NP problem. Despite of the class complexity P seems to be an independent class but it is contained into the class NP which contains many other problems such as the sales man problem, the graph coloring problem and the maximum clique problem. The hardest problems in the NP class are called the NP-complete problems; these problems don't have algorithms that run in a polynomial time.

The coloring graph technique used in many applications but the most resent application used graph coloring problem is social networks which has been existed since more than one decade but it has revolutionized in m most real life fields, so that the most importance application in computer science which applied the coloring graph is in social networks, but social networks are not free of drawbacks, characterized in its huge size (number of subscribers) and its number of relations(friendships) subsequence the problem or deal with these huge networks is more complex, sand required strong algorithms and tools to manipulate with the massive social networks.

Social networks

Social networks became the most important media and widely used in the world, it is important and useful due to the wide range of using in most fields such as marketing, education, health, promotion, the fight against crime and smuggling and so many others, using social network can save efforts, money and time**.**

Social network is a very huge network and Increases dramatically, millions of millions of people are connected directly or indirectly, those people have interests, hobbies, opinions geographical areas and so many characteristics that could be common. People in social networks distributed as clusters in vary sizes, our idea is to find these independent clusters, finding clusters help us to deal with large and dynamic graph efficiently such as sending advertisements or invitations or solving some social problems like addictions in teenagers or to finding criminal people and following them. Finding clusters help us in making statistics and studies on people whom have specific characteristics.

Graphs can be used to represent different type of problems in social networks, also graph can be used to solve many kinds of problems; one of these problems is the social networks problems. Obviously, each node in graph G will present as a person profile and each edge connects two nodes if there is a friendship between two people presented by the nodes. The graph coloring problem is such a graph labeling that we use to color graph nodes under some conditions with minimal total number of colors (chromatic number). It colors graph nodes such that no two adjacent nodes sharing same color, which we call it nodes coloring. It can be deduced that if two nodes have the same color, then they have no friendship that connects them.

1.2.1 Clustering in social networks

Clustering is the process of finding the completely connected subgraphs in large graph such as social networks Chalupa, (2011). this process is very important in many different aspects such as financial, social, business, management, marketing, advertising, and others Yan, B., & Gregory, S. (2009), Clustering enables to deal with scattered individuals with common at least one attribute as one group Stix (2004)., the main benefits of clustering are saving time and reducing cost, increase productivity, reducing overload on networks, speed up the analysis and decisions making, clustering help us to detect the behaviors and interests of individuals, and also clustering in social networks allows to manage relationship, for example friends can exchange their interests by adding new friends having common interests, and find pages when share common followers.

1.3. Problem statement

Finding clusters in large networks such as social networks is increasingly gaining importance due to financial, social, and political reasons. However, finding clusters, which is equivalent to find cliques in graphs is known to be NP complete problem

Yan and Gregory(2009). Two techniques have been used to solve problem, the first one is exhaustive search technique (brute force or blind search) which is guaranteed finding all clusters but it exhausted time so its performance is very weak special that the size of social networks getting larger and larger , subsequent the brute force method is useless In response to this problem, another technique is considered more smarter which it uses a heuristic in searching process, various algorithms follow these techniques such as the Tabu algorithm, Bron-Kerbosch algorithm ,SCAN++ algorithm, and others. In this research, we will investigate a new algorithm based on graph coloring technique to solve the problem of finding clusters in reasonable time.

In this work, pruning algorithm and brute force algorithm have been chosen to conduct the experiments and compare result with proposed algorithm. The brute force algorithm had chosen because it has ability to find the all clusters in 100% accuracy, hence brute force method is considered as standard method. The most well-known Numeration search algorithm is also used in this research due to its balanced simplicity and performance, Numeration algorithm works in reasonable time and gives good success in find clusters, considering all possible solutions is impossible within a reasonable amount of time. It avoids being trapped at local optimum, it turns out that satisficing solutions are possible that can produce good enough solutions Taylor (2006), all previous characteristics were the motivation to adapt Tabu algorithm in this research.

In response to this problem, our study proposes to investigate method to find and identify the cliques in graphs in a reasonable time and in enough accuracy in order to save time and reduce cost, also to increase productivity, reduce overload on networks, speed up the analysis and decisions making. Furthermore, clustering help us to detect the group of similar interests of individuals and classify or group them into a set of categories or clusters, also clustering in graph allows to manage relationship between full connected nodes. Finally, there is a need to improve the accuracy and speed up the time of finding cliques in a graph by utilizing existed algorithm. Therefore, this research aims to identify the cliques in particular graph using graph coloring technique.

This research has considered the following questions to be answered in order to achieve the research aims:

1. Can graph coloring techniques be used to find cliques in large graphs such as social networks?
2. Does the use of graph coloring techniques improve the accuracy and the time of finding cliques problem in large graph?

1.4. Hypothesis:

In order to answer the research questions the following assumptions has considered to guide the flow of research:

H1: Graph coloring techniques can be used to find cliques in large graphs such as social networks.

H0: Graph coloring techniques cannot be used to find cliques in large graphs such as social networks.

H1: Using graph coloring techniques will increase the accuracy and time of finding cliques within a graph than other heuristic techniques algorithms.

H0: Using graph coloring techniques will reduce the accuracy and time of finding cliques in within a graph than other heuristic techniques algorithms

1.5. Objectives

This research aims to present new technique to solve clustering problem in graphs by finding to linked nodes which is called a clique. this problem is considered as a very complex problem, the proposed technique is used to represent vertices of the similar group's for individual nodes. Finding these relationship in large graph is a complex problem and need a lot of time and accuracy is not grant, and did not find any algorithm until now to solve clustering on large graphs with in optimal solution which resonate between time and accuracy, hence detecting and identifying cliques is large graphs in restorable time and with highest possible of accuracy are on the top priority and aimed by a lot of scientists and researcher whom concern in graph theory studies.

This work involves other researcher's attempts to solve this problem, however, a new utilization has conducted on their work to solve the problem.

The main objectives of this research are:

1. Finding the most proper, simplest and effective algorithm for coloring graphs which using the minimum number of colors in graph coloring process.
2. Using graph coloring techniques in detecting and identifying the cliques in large graphs.
3. Apply and run the proposed approach in java integrated development environment.
4. Measure the complexity in detecting cliques in graph using graph coloring technique.

5. Obtain numerical and graphical experimental results for the proposed approach and other algorithms, then analyze and compare the results of the proposed approach with the of other algorithms.
6. Simulating social networks on order to applying and testing the new approach on simulating social network hopefully applying the proposed approach in future.

1.6. Motivations

Current social networks dominated many aspects of human behaviors and relations. Social networks have impact on political life of people as was expected in during arab spring. Social networks have become vehicles for social and financial development, however, the size of social networks has become extremely large such that it has become so difficult to analyze and extract meaningful information.

All researches have been conducted on clustering in large graphs did not solve the problem in deterministic time and high accuracy, not also dynamic programming do that, and existing algorithms tried to solve clustering problems but it's solution often falls into trap of local optimal and therefore cannot converge to global optima solution

. One of the main properties of large social networks is clusters which represent a group of people who share one or more common attribute such friendship, geographic location, special interest in sports political and others. Therefore, I developed interest in detecting and classifying clusters in large social networks in reasonable time.

Having studied graphs and graph coloring solutions and the shortage in existed algorithms I have become motivated to use graph coloring technique as a method for solving the complex problem of clusters identifying and classifying in large social networks|

1.7. Contributions

The main aim of this thesis is to utilize Largest Degree First algorithm for graph coloring to detect clusters in large graphs such as simulated social network, and the others aims are presented as the following:

This thesis has presented a new method to find cliques in large networks such Facebook and LinkedIn.

1. Evaluate and compare between the proposed approach with brute force algorithm and pruning algorithm in terms of execution complexity and accuracy of detecting cliques with consideration of that LDC has applied on simulated social network and the other algorithms have applied on exams scheduling.
2. Applying the proposed approach in variant densities and graph sizes
3. Applying the proposed approach on simulated social networks in size of 50 000 nodes hopefully to be applied in real social networks in future.
4. Finding cliques in graphs by using a new fast and near optimal graph coloring technique

1.8. Thesis organization

This thesis is organized in five chapters and each chapter divided into sections and some sections divided into subsections, the five chapters are:

1. Chapter one, introduction: an introduction about the topic is given, also an overview about essential background subjects are provided in graph coloring, social network, clustering in social network.
2. Chapter two, literature review: represent similar conducted work in graph clustering and graph coloring and summary of articles that published by other researchers.
3. Chapter three methodology, provides conducted steps to implements the proposed approach and related example showing the sequence of algorithm implementation.
4. Chapter four conformance analysis and experiments results this chapter provides an. overview of the programming language that used for implementing the proposed approach, showing the results of experiments and evaluation
5. Chapter five. conclusion and future work

CHAPTER TWO

LITERATURE REVIEW

2.1. Background.

A lot of studies have been conducted by many scientist and researchers focused on approaches and methodologies that used to solve clustering in graphs, this problem is considered as NP problem, that mean there is no algorithm to find optimal solution in reasonable time, so the researchers exerting extensive efforts to finding near optimal solution in reasonable time.

This review provides a simplified version and comprehensive review on Maximum clique and maximal cliques problem. Though capturing the complete literature in this regard is beyond scope of the research, but it is tried to capture most of the representative paper from similar approaches

exact algorithms and heuristic algorithms. In this section, a review about what the researchers have contributed in the field of finding maximum clique problem (MCP), this review divided the researchers approaches of Algorithms for NP-hard problems into two main approaches, the exact algorithms approach and heuristic algorithms approach, for each approach I will review the both approaches showing a few algorithms.

2.2. Graph coloring problem:

There are two main approaches to solve problem of finding cliques in graphs, the first approach is brute force, a brute force is a technique study of all candidate's solutions so it always find all solutions, but checking all candidates solution will cost too much, and this cost is proportional to the number of candidate solutions, which in many practical problems will grow very quickly as the size of the problem increase. brute force algorithm is typically used when the size of problem is small or when we want find all solutions regardless of finding cliques time and cost, backtracking in graph coloring graphs is an example of brute force techniques, however, backtracking method is very complicated in coloring graph with minimum chromatic number, therefore, it has not been chosen to be the approach for this research.

The second approach is heuristic which use smart and intuition ways to found best solution among all possible one, there is no guarantees that the best solution will be found, the heuristic algorithms considered as approximately and inaccuracy algorithms to find solution close to best one in speedy and easily, but in some cases the heuristic algorithm can be accurate when ti find the best solution but the algorithm still called heuristic until this solution is proven that is the best one.

Pattillo and Butenko (2011). Introduced the closely related maximum clique, maximum independent set, graph coloring, and minimum clique partitioning problems. and survey includes some of the most important results concerning these problems, including their computational complexity, known bounds, mathematical programming formulations, and exact and heuristic algorithms to solve them, the results showed that the using heuristics for the maximum clique and graph coloring problems usually have little theoretical justification or performance bounds

2.3. Graph Clustering

Clustering is the process of finding the completely connected subgraphs in large graph such as social networks Chalupa, (2011). The main benefits of clustering are saving time and reducing cost, increase productivity, reducing overload on networks, speed up the analysis and decisions making, clustering help us to detect the behaviors and interests of individuals, and also clustering in social networks allows to manage relationship. However, there many studies have conducted to find cliques in social network, one of them is the effort of Azizifard.

Azizifard(2014) claimed that the analyzing of social network is very important and to get accurate analysis to social network you must looking to clusters, he proposed an algorithm which aims to finding communities in a way that modularity factor increases, to achieve this goal, random walks with random local search agent are combined. Experimental results show that the proposed method gives better modularity in comparison with other algorithms, the empirical results on two social network datasets showed that the proposed method gives better modularity in comparison with other approaches.

2.4. Social Networks

Clustering is the process of finding the completely connected subgraphs in large graph such as social networks Chalupa, (2011). The main benefits of clustering are saving time and reducing cost, increase productivity, reducing overload on networks, speed up the analysis and decisions making, clustering help us to detect the behaviors and interests of individuals, and also clustering in social networks allows to manage relationship. However, there many studies have conducted to find cliques in social network, one of them is the effort of Azizifard.

Azizifard(2014) claimed that the analyzing of social network is very important and to get accurate analysis to social network you must looking to clusters, he proposed an algorithm which aims to finding communities in a way that modularity factor increases, to achieve this goal, random walks with random local search agent are combined. Experimental results show that the proposed method gives better modularity in comparison with other algorithms, the empirical results on two social network datasets showed that the proposed method gives better modularity in comparison with other approaches.

2.5. Related works

Malkawi & Hassan (2008*)*.Presented An exam scheduling application based on a graph-coloring- algorithm, with the objective of achieving fairness, accuracy, and optimal exam on limited time period the number of available halls, and the availability of faculty to conduct the exams. Through the work, they considered few assumptions and constraints, closely related to the general exam scheduling problem, and mainly driven from accumulated experience at various universities. The performance of the algorithm was a major concern. After run of the scheduling algorithm with several values. the results indicated that the exam scheduling algorithm achieved near optimal performance (close to minimal number of colors) in polynomial time.

2.6. proposed approach

Graph coloring technique is the approach for this research because of its ability to color the nodes within a large graph which allows to detect cliques. Pruning algorithm and brute force algorithm have been chosen to conduct the experiments and compare result with proposed algorithm. The brute force algorithm had chosen because it has ability to find the all clusters in 100% accuracy, hence brute force method is considered as standard method. The most well-known Tabu search algorithm is also used in this research due to its balanced simplicity and performance, Tabu algorithm works in reasonable time and gives good success in find clusters, considering all possible solutions is impossible within a reasonable amount of time. It avoids being trapped at local optimum, it turns out that satisficing solutions are possible that can produce 'good enough' solutions Taylor (2006), all previous characteristics were the motivation to depend Tabu algorithm in this research. The research attempt to find new algorithm that has advantages of brute force, heuristic approaches and avoidance their disadvantages, the new algorithm will follow heuristic way and try to reach tall possible solutions in reasonable time. the expected results of using the proposed algorithm could get better accuracy than the other heuristic algorithm such as pruning Algorithm. By applying *(LDC)* algorithm on graphs the output will be the colored graphs consuming the minimum number of colors (chromatic number). Furthermore, a new function will be added into the previously existed Largest Degree Coloring (LDC) Algorithm order to create improved algorithm with high ability to detects cliques in a graph in reasonable runtime.

CAPTER TREE

Methodology

This chapter covers and clearly defines in detail the methodology of propped approach that has been adopted in the research of finding cliques in simulated social networks using graph coloring technique, the title of research refers that the study problem consists of two problems, first problem is detecting cliques in simulated social network which can be formulated as finding cliques in large graphs which considered as NP hard problem , and the second problem is coloring large graphs which considered NP complete problem, hence to solve this very complicated combinatorial optimization problem the new approach that has been adopted in this research was so carefully designed as to go well with this complex field

3.1. Methodology

A new heuristic method has presented to solve many problems that encounter students or researchers whom studying or depending in their studies on graph theory, graph theoryis considered as repository of unsolved problems, this study has deal with graphs as data structures which consists of a set of nodes and edges.

Our approach aims to utilizes the most proper and effective graph coloring technique to get colored graph with the minimum number of colors then finding clique in that graph during the coloring process by using the comparisons between nodes that have been colored just yet. the approach aims to find cliques in larges graph in better runtime than runtime of other heuristic approaches, The proposed approach works in innovation way such that works alternately between the applying *(LDC)* algorithm on graphs and process of detecting cliques by using the nodes that have been already colored

3.2. Proposed approach

The methodology of proposed algorithm consists of many steps; these steps are proposed to detect cliques in graphs. Following are the steps for proposed algorithm:

The methodology of proposed approach consists of many steps; these steps are proposed to detect cliques in graphs. The following are the steps for proposed approach:

1. Sort nodes based on degree in descending order, store it in list called MainList.
2. Select the first node in the Main list (which has the largest degree).
3. Color the node with smallest available color.
4. List the neighbors of the selected node and store them in a list call it SubList.
5. Sort the neighbors of the selected node (all nodes inSubList) in descending order based on degree, in SubList, if two or more nodes have the same degree, choose any node.
6. Color the nodes in SubList starting with largest degree node (as it sorted in SubList) by the smallest available color,
7. After Coloring the node in main list and all its neighbors store the node in mail list in Array called Clique.
8. compare each colored neighbor in SubList with every node stored in array Clique if the node in SubList has the same color of any node in Clique ignore the node and check

next node in SubList, but if node has adiffer color than all nodes in array Clique, add it in the clique array because it is part of clique.

9. When all nodes have been checked print out the elements of Clique array because these elements are the detected cliques

10. Return back to Main list and select the next node of the main list.

11. Go to step 3.

Stop when all nodes have been colored

3.3. Example on proposed approach

To clarify the methodology of proposed approach in detecting cliques in in simulated social networks using graph coloring technique, simple example will be presented and explained in details, showing the sequence of steps that the proposed approach follows during the process of running, until end of program (results appear).

To simplify, imagine and understood the methodology of proposed approach all stages during the running of proposed approach program to find out the cliques in this graph will be demonstrated and depicted in figures

The proposed approach works on graphs data set and returns the all cliques in that graph, subsequently the input of this example is arbitrary graph and the issue is to find cliques in this graph using the proposed approach.

On order to understand the example two terms must be clarified before studying the example, the data set and table of colors, the following are beefily description for them:

3.3.1 Data set of the example.

The data set of this example is graph generated randomly by proposed program in size of 23 nodes, each node denoted by distinct English lower letter, and each node connected with each of its neighbors by edge, the aim is to detect the groups of nodes that have full connected with each other if exist by applying the proposed approach on this graph, the figure shows the graph that will be used as data set in this example, the graph

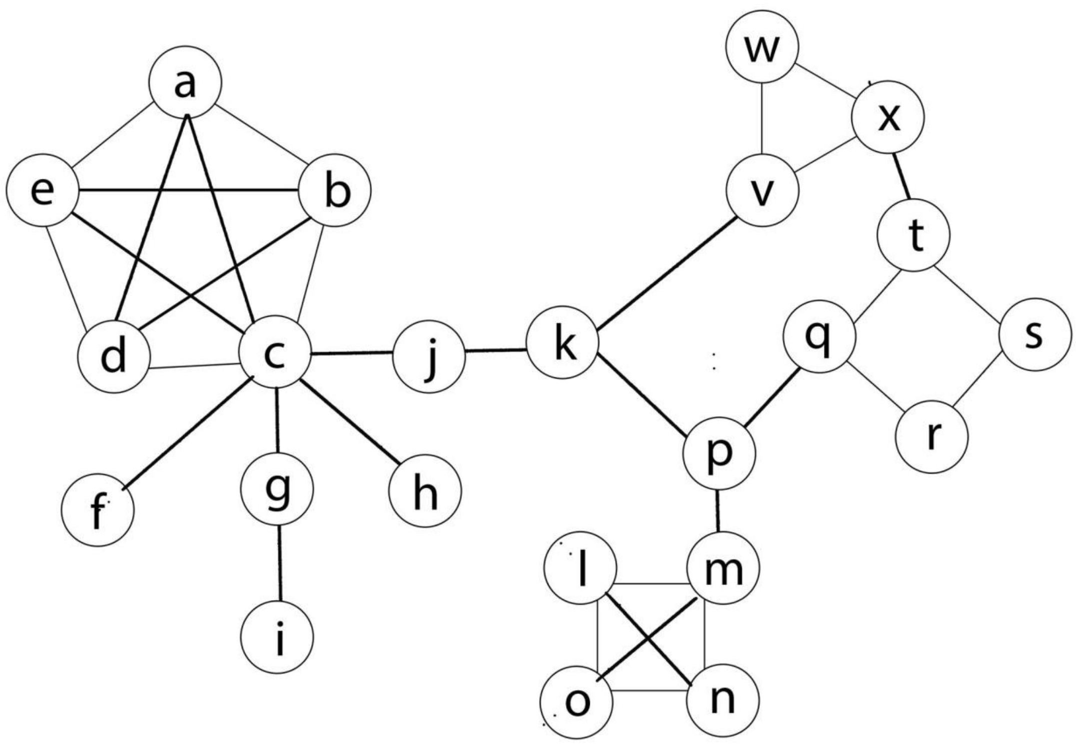


Figure 0-1 Graph Generated randomly

This form of graph just for but in computers its different, the graph is represented in computer as adjacent matrix, Table 0-1 shows how the graph in figure represented in computer.

Graphs can be represented in computer systems in many ways such as adjacency list, adjacency matrix; but the most used is adjacency matrix in this format, an [n, n] matrix is used to represent the graph, where n is the number of nodes. Matrix element [x, y] is one if and only if there is a relationship between node x and node y. and it is 0 otherwise.

Table 0-1 adjacency matrix

Nodes	a	b	c	d	e	f	g	h	i	j	k	L	m	n	o	P	q	r	s	t	u	v	w	x
a	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
b	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
c	1	1	0	1	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
d	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
e	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
f	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
g	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
h	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
i	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
j	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
k	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0
l	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0
m	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	0	0	0	0	0	0	0	0
n	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0	0
o	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0
p	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	0	0	0	0	0	0	0
q	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	0	0	0
r	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0
s	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0
t	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0
v	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	1
w	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

.

3.3.2 Table of colors

As mentioned in an introduction the coloring process is assignment of labels traditionally called "colors" to elements of a graph subject to certain constraints, these labels called colors, color term is added metaphorically hence it could be color or number.

in this study, the colors will be assigned as numbers, but in this example, we will use colors by its real meaning in order to draw a clear depiction of the coloring processand it is easier for the visual eye to distinguish colored nodes than nodes labeled by numbers.

The color which will be chosen for coloring such node is determined subject to certain constraints to choose from group of colors, these colors must be sorted in ascending order start from minimum color to maximum color as depicted in Figure 0-2 list of colors

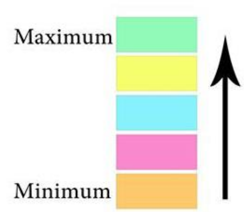


Figure 0-2 list of colors

To assign a color to any node the minimums color must checked firs, assign the minimum color to a node unless there is a conflict if no break the constrains assign it to node, but in the event of conflict go to next color under the graph coloring algorithm requirement and so on.

Here the process of applying the proposed approach on graph

To sort nodes of graph based on degrees number we must count the nodes that have and edge with that node, in other words find the degree of each node, the degree of each node can be found by accumulating the all neighbors of that node and as neighbors of any node are connected with that node by edge subsequently the to find the degree of node done the accumulating the edges that connected that node , and as mentioned before the edge represents of ones in adjacent matrix so by accumulating the one's of each row we get the degree of that node,

The table below shows the degree of each node in this example, after finding the degree number of each node we sort the nodes descending order to start coloring form the largest degree node, the figure 4 shows part of sorted matrix sorted according to largest number of nodes degrees, notes that the node c is the largest degree then node m …

After that the first node (**C**)is colored in this example with the minimum available color. And it stores the color and node letter in CliqueArray array.

Then we pick the node with largest degree (**C**) and check its neighbors, and sort them descending based on the number of its degrees, as shown in figure 5. Table 0-2The neighbor of nodes and its number of degree sorted descending

After sorting the nodes of graph in descending order and order the neighbors of each node descending too the first node is started coloring and all it's neighbors using graph coloring algorithms which constrains adding the minimum available color and no one of it's neighbors have the same color.

C	**b**	**a**	**d**	**e**	**f**	**g**	**h**	**j**

These nods will have appeared colored as **Error! Reference source not found.** illustrates

After applying the steps of the proposed algorithm, the first node of Main list (c) has been colored and all its neighbors have been colored also. as depicted in figure 6. after the process of checking these nodes does it form a clique or it does not form a clique.

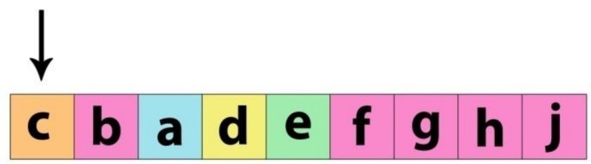


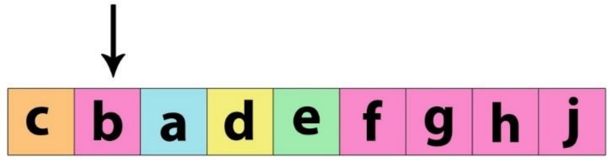
Figure 0-3 Main Node and its Neighbors after coloring

Here the process of checking has started if these colored nodes forms a clique or not by comparing it's colors.

First step store the node of Mail list int Clique1 array without checking because it's color will not be like it's nighbiurs colors.

Figure 0-4 first node stored in clique array

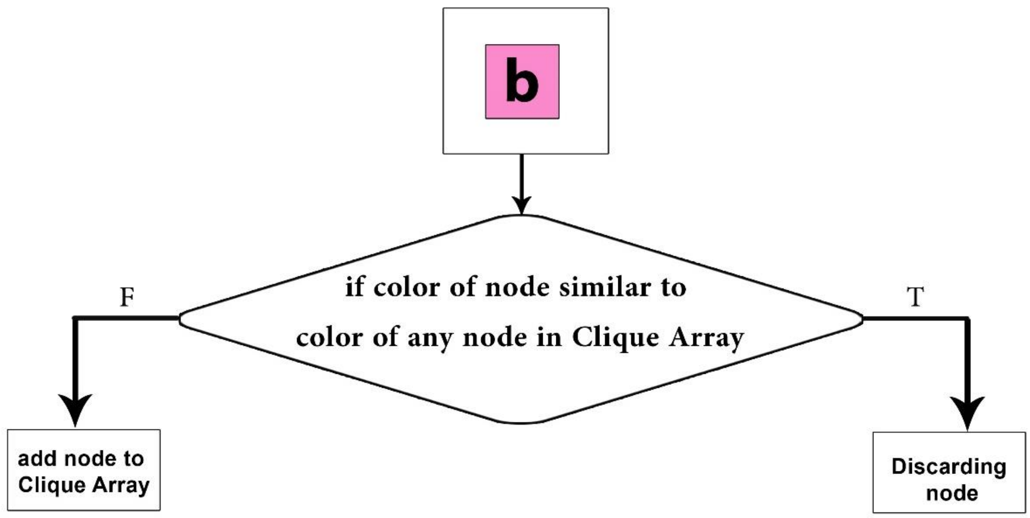
After storing node of main list in cluque1 array the algorithm starts to check every node of it's neighbors Here check the first node (largest degree) of it's neighbors (b) if his color similar to any color in Clique1 array? Here is not so the algorithm add it to Clique array



Does 

contains color of 

?



Since there is no pink color in array we add it Clique1 array

Figure 3.0-5the process of checking is elected node with cliques array element

Check the next neighbor (a) if has color similar any of node in Clique array

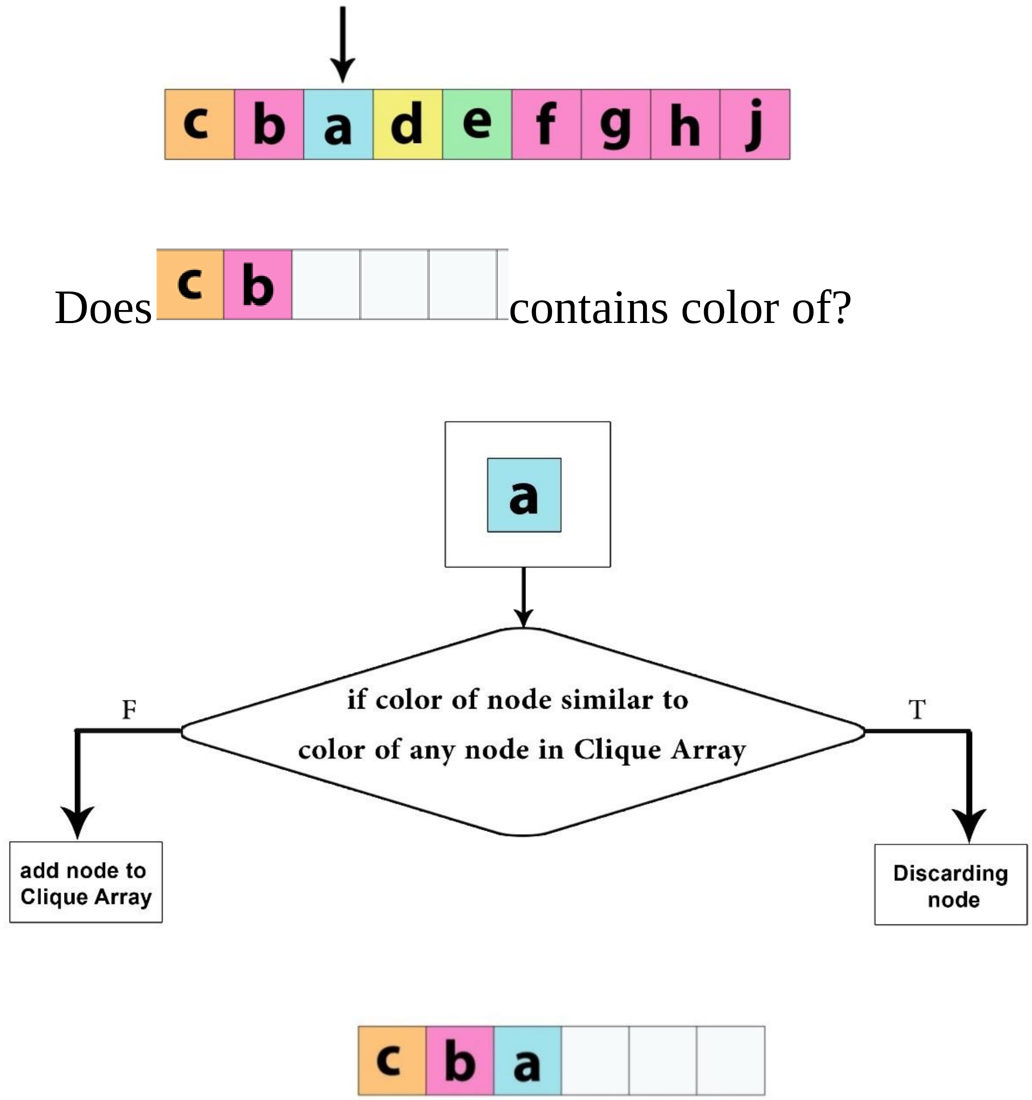
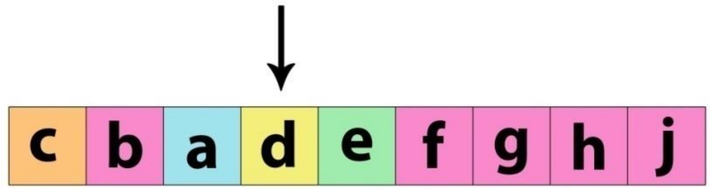
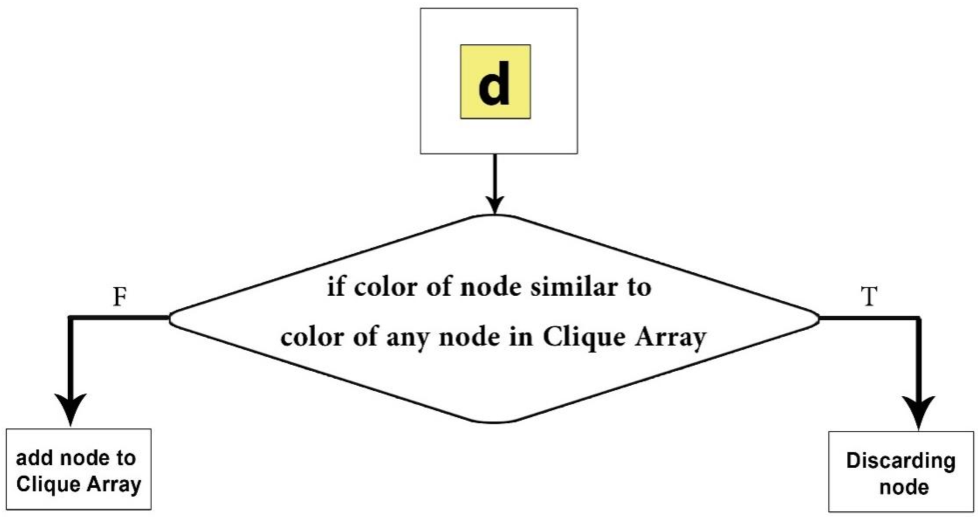


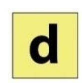
Figure 0-6 the process of checking is elected node with clique s array elements

Then goes to next neighbor (d) and checks if there is any node hold the same color in the cliuew array

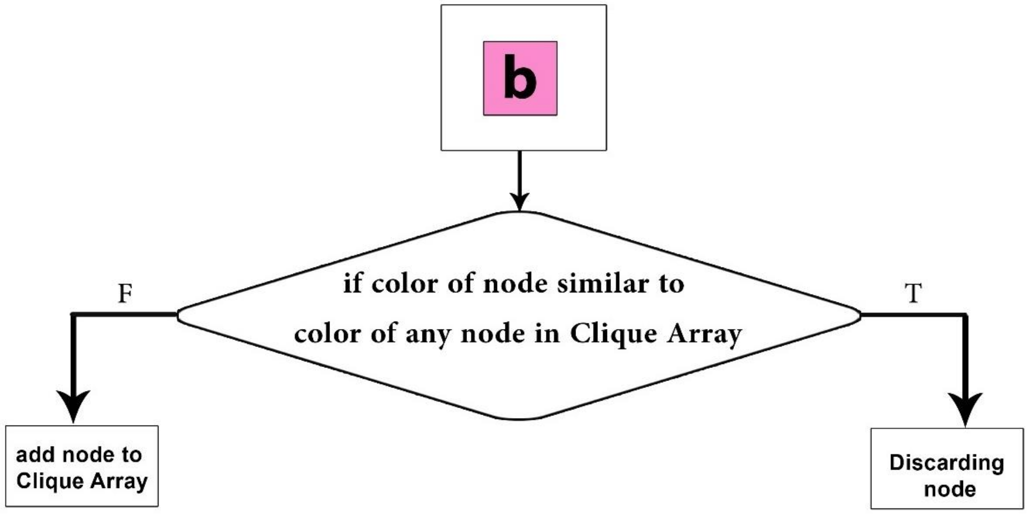




Does 

contains color of 

?



Since there is no yellow color in array we add it Clique1 array

.

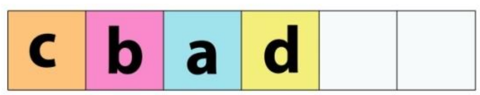
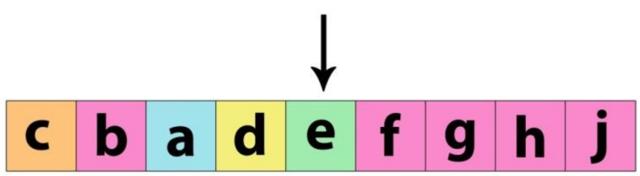
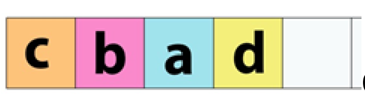


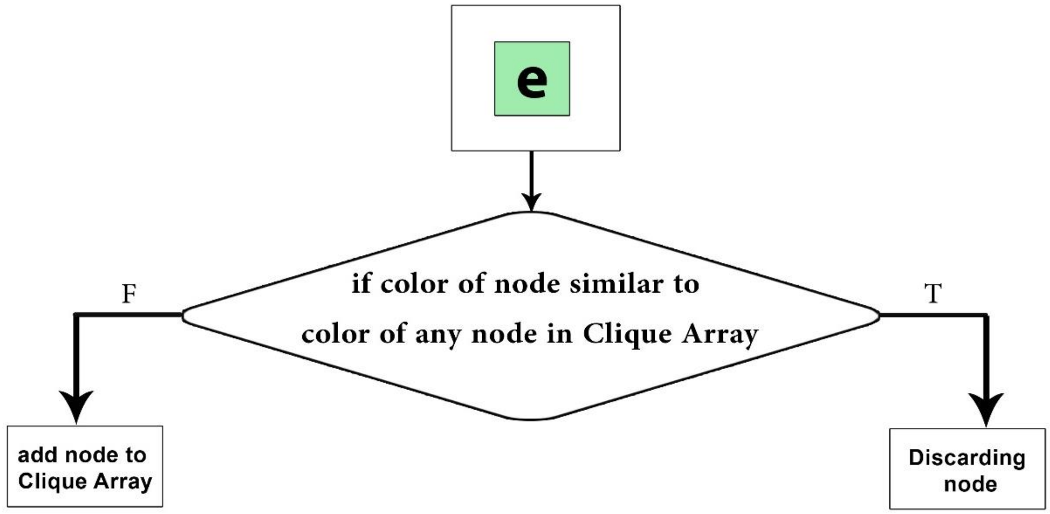
Figure 0-7 the process of checking is elected node with clique's array elements

Then goes to next neighbor (e) and checks if there is any node hold the same color in the cliuew array



Does 

contains color of? 



Since there is no green color in array we add it to Clique1 array

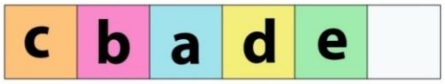
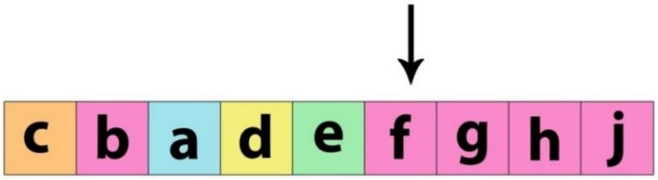


Figure 0-8 2the process of checking is elected node with clique's array elements

The algorithm checks the next neighbor (**f**) if has color similar any of node in Clique array



Does 

contains color of? 

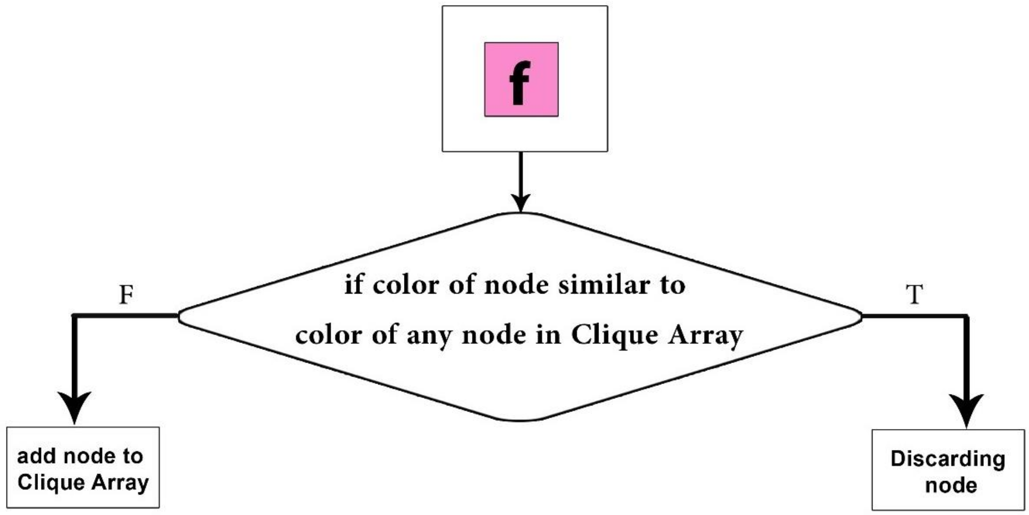


Figure 0-9 the process of checking is elected node with clique's array elements

There is a node (b) has the same color of (f) skip the node (f) and goes to next node in SubList

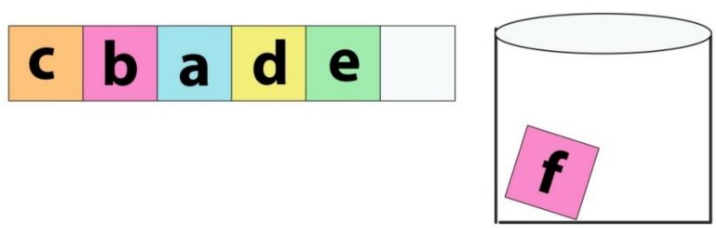
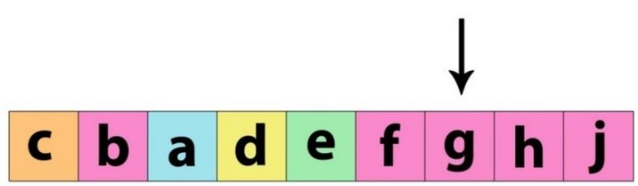


Figure 0-10process of checking is elected node with clique's array elements

The algorithm checks the next neighbor (g) if has color similar any of node in Clique array



Does 

contains color of 

?

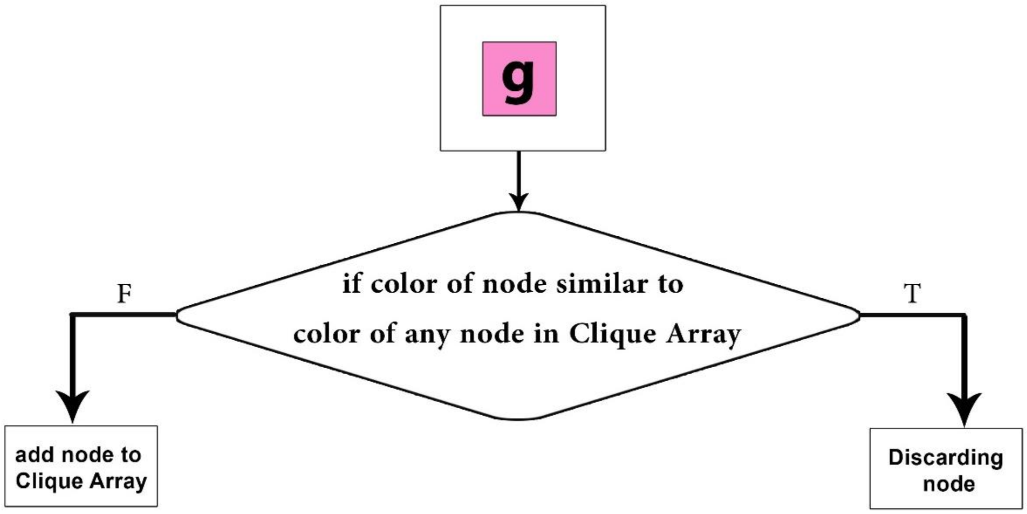
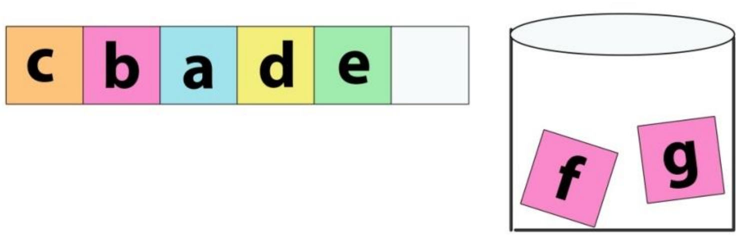
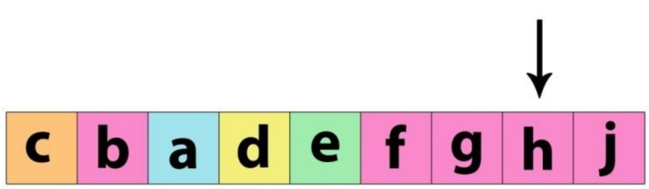


Figure 0-11 process of checking is elected node with clique's array elements

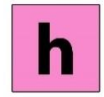
There is a node (b) has the same color of (g) skips the node (g) and goes to next node in SubList



The algorithm checks the next neighbor (h) if has color similar any of node in Clique array



Does 

contains color of 

?

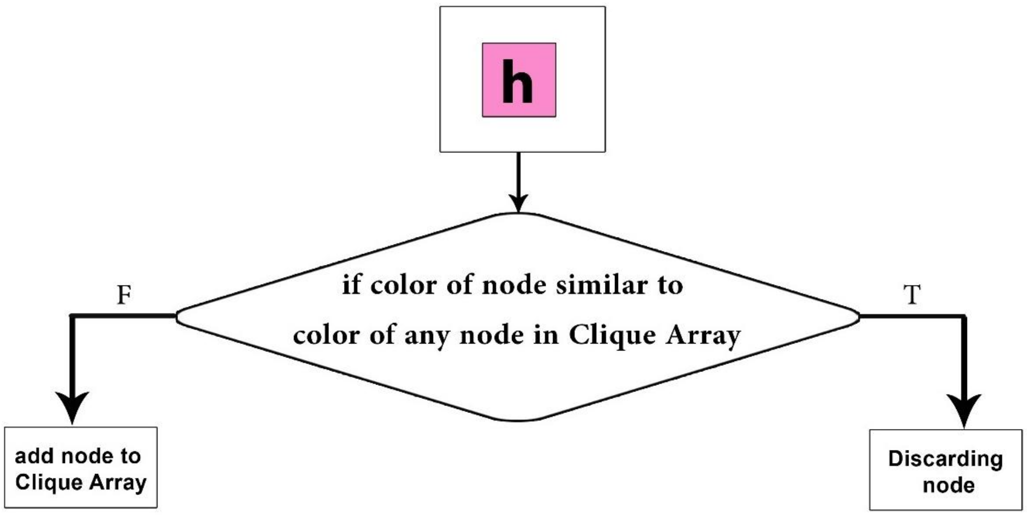
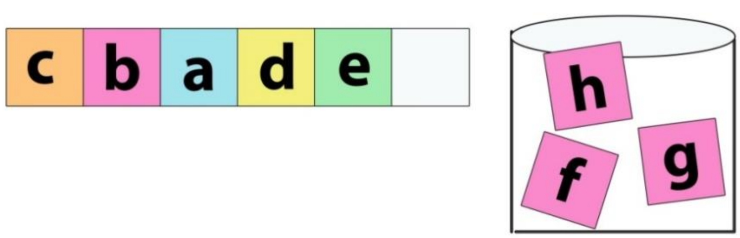


Figure 0-12process of checking if elected node with clique's array elements

Figure (3.4): There is a node (b) has the same color of (h) skip the node (h) and go to next node in SubList



The algorithm Checks the next neighbor in subList (j) if has color similar any of node in Clique array

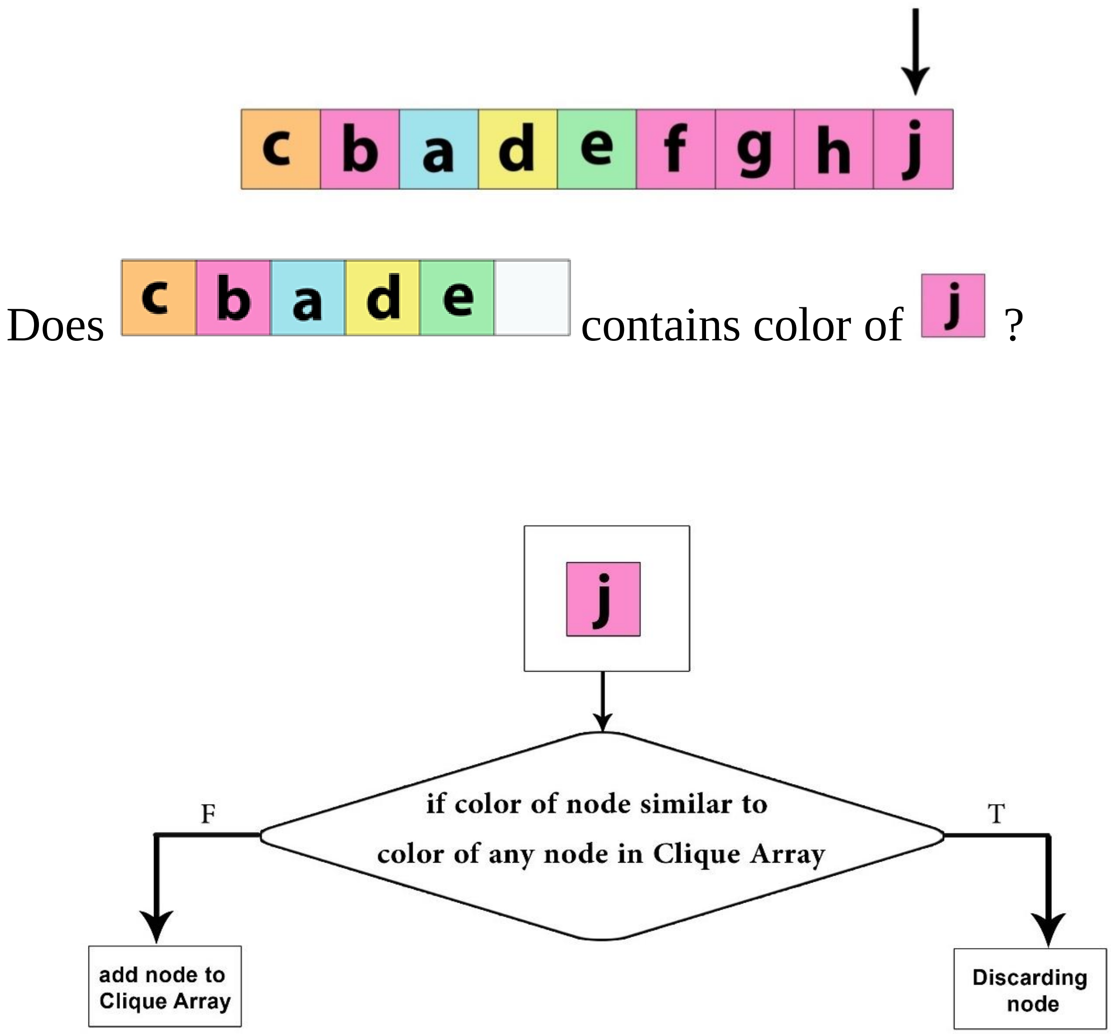
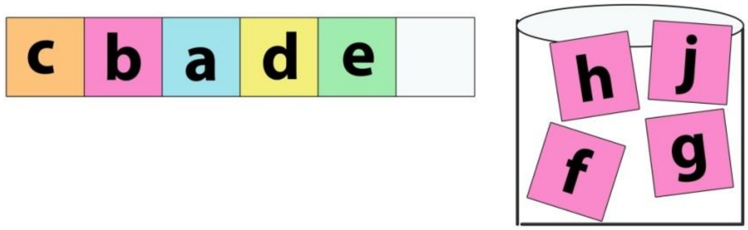


Figure 0-13 the process of checking is selected node with clique's array elements

There is a node (b) has the same color of (j) skips the node (j) and goes to next node in SubList



Her there is no element stays in SubList,

Then prints the elements of Clique1 array, this element is the first clique detected by using the proposed approach.

Now returns back to Main list and selects the second sorted node (m) and repeats all previous steps

Then continue until all node in MainList and their neighbors becomes colored.

Figure 0-14 shows the all nodes of graph has been assigned by proper color after running the proposed approach, all nodes on graph become colored, and due to using the largest degree coloring algorithm it’s supposed to be colored using the minimum number of colors

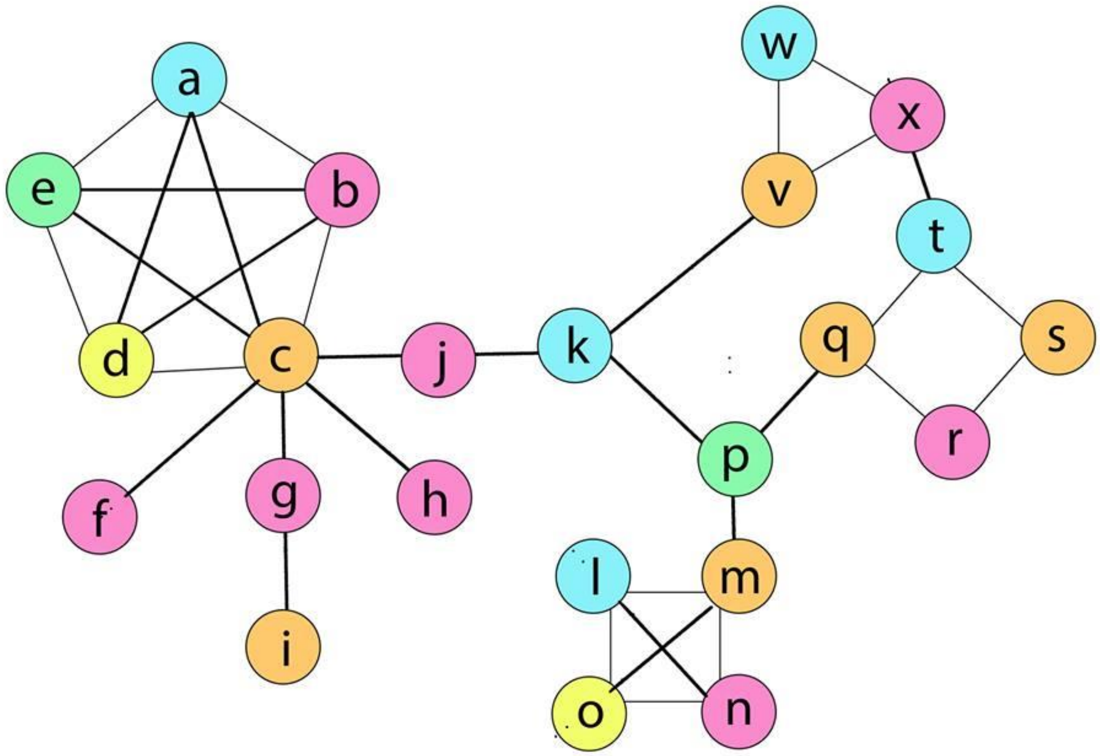


Figure 0-14 process of checking is elected node with clique's array elements

The **Error! Reference source not found.** shows the graph that has been colored and cliques that have been detected in graph using the proposed approach, each detected clique is surrounded by a colored circle to show all the nodes that formed each clique.

Error! Reference source not found. show the table of colors uses by graph coloring algorithm, the minimum number has the priority to be used, if there is conflict by coloring with first color go to next color and so on.

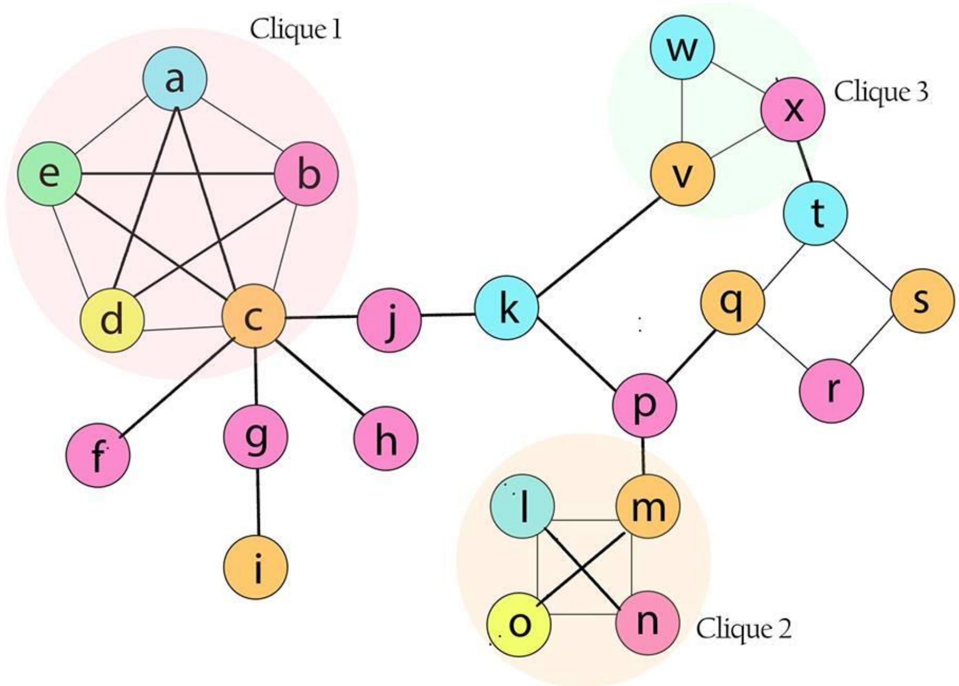
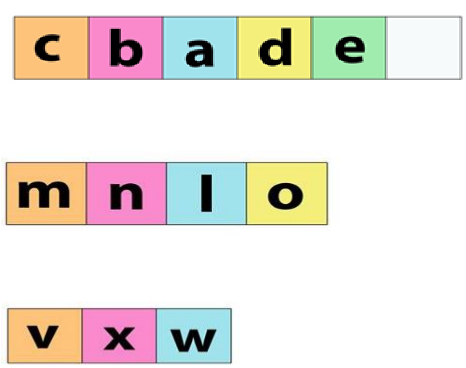


Figure 0-15 The detected cliques

By repeating the above steps until last node in mailList we will get other two cliques Clique 2 = { m, n, l, o } and Clique 3 = { v, x ,w }.**Error! Reference source not found.** shows all detected cliques in graph by using the proposed approach.



Clique 1 = {c, b, a, d, e}.

Clique 2 = {m, n, l, o}

Clique 3 = {v, x, w}

Figure 0-16 claques detected by proposed approach

CHAPTER FOUR

PEFORMANCE ANALYSIS AND RESULTS

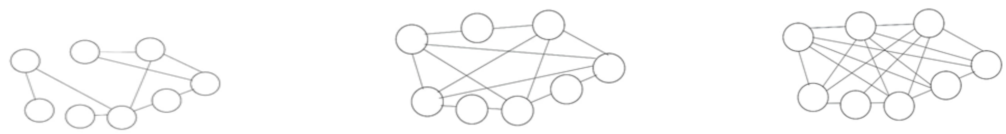
this chapter explains the data sets that has been used in this research, evaluation of experiments, compares results. Then explanation of complexity analysis of proposed algorithm, numeration algorithm and brute force algorithm followed the results of experiments of applying these three algorithms on a set of variant densities and sizes graphs

4.1. Data set

The data sets that have been used in this research can be categorized into three categories, these variant of data sets aims to examine the proposed approach and other two algorithms in different environments, the data sets of this research are differs in multi factors such as size of 0graph, densities of graph, it's source and method of generating graph, the following: are the categories and briefly description of each one and how it used in this research, these categories are : randomly generated graphs, benchmark graphs and simulated social network, following are briefly description for each one.:

4.1.1. Randomly generated graphs

These graphs are generated by program coded in java programming language, the user previously determines the. two main factors in creating graphs, the size and density of graph, size of graph is determined by an integer number refers to the number of nodes, number of nodes is an integer number from two to required size, in this program the user must enter the number of nodes and determine the density of graph that he want to create before running the program and also must be an integer for the graph size, the anther factor is density. here is defined as the probability of a pair of nodes being connected, in this study the density is divided into three types, Low Densit,regular density andheavy density



Low Density graph | Regular Desity Graph | Heavy Desity Graph

4-1 (A) | 4-2 (B) | 4-3 (C)

Figure 0-1 types of graphs densities

The Table 0-1 shows the three types of desires, notice that the number of edges in heavy density graph is more than edges in the both regular density graph and low density graph.

4.1.2. Benchmark graphs for detecting communities in graphs

These graphs gotten form DIMACS, it consists of variant numbers of nodes, in this research five graphs in sizes between 540 nodes to 1272 nodes, in evaluation section, the proposed approach will be run these data sets to evaluate its efficiency.

4.1.3. Simulated social network

Due to un availability of real social networks data set for security and privacy reasons, simulating social is a temporary alternative to real social networks, the base structure of simulating social networks is graph, graph consists of nodes and edges where the nodes represent the social network account and edges represent the friendship relations between the social networks accounts in this study, the real social networks will be simulated and utilized as data set in order to apply the proposed approach on it to detect groups of people who everyone is friend to rest of group, real social network account has a tremendous objects, attributes and functions but not all these features will be simulated, but just features related features to this research will be simulated in order to create simulated social network for finding cliques, the most important feature is a friendship relations, which can be represented by the edges between nodes in graph, if two persons are friends in such social network that means the relationship between them will be simulated as an edge connects two nodes in graph, anther attribute in social network the account has many attributes like ages and hobby, but in simulated social networks in this study needs for variable to hold the labels(colors), this attribute will be used in proses of coloring in simulated social networks,

4.2. Performance analysis

This research consist of utilizing the Largest Degree Algorithm for coloring graphs by adding functions in order to detect maximal cliques, maximum clique and intersects cliques in graphs during the coloring process, and to add credibility and trustiness to this research many experiments have conducted on variant types of graphs and many other experiments on same graphs have conducted using two other algorithms, these algorithms are brute force algorithm and Numeration algorithm, these two algorithms have chosen because to their ways of finding cliques, brute force algorithm search in graph exhaustively, this method finds all cliques definitely despite of the high cost of its search , so it has considered as standard to know all cliques without losing any one (because it will check all possibilities), but this algorithm consumes very long time regarding to the size of graph.

The other algorithm that has been chosen is Numeration algorithm, this algorithm characterized by the ability to find cliques in graph in heuristic way and follows intelligent technique, also it consumes less time but it does not take all the possible cliques within a graph.

4.3. complexity analysis

In the context of computational time complexity, the proposed algorithm is considered faster than the numeration algorithm, the proposed algorithm can be run in O (di^2) in worst case, while numeration algorithm needs O (n^3) in worst case.

4.3.1 complexity of the proposed approach

suppose that the largest degree in graph is deg = d1; and that node v1 has degree K1, the following steps are representing how to compute the complexity of the proposed algorithm:

The first step assigns the smallest color, called**col1** to the first node **n1**. The total number of steps required to color all the nodes in the neighbor list of n1 is 1+2+3+ … + d1 and the process of checking if the colored nods is form clique or not required the total number of neighbors and it equals to total number of neighbors (d1), this process in done linearly so its effect will ignore when computing the complexity of proposed algorithm, hence the equation which is form the mathematical series that is computed the total of elements summation by the mathematical equation (square of last element plus the last element divided by 2) plus the d1, ($d1^2$+ d1)/2 + d1 which represented in mathematical notation as O ($d1^2$) in the worst case, below is the final equation for the complexity computation of coloring first

$$1+2+3+\ldots+d1=(d1^2+d1)/2\ \ +d1=O(d1^2)\ \ldots\ldots\ldots\ldots\ldots\ldots\ \text{Equation 1}$$

Repeat the coloring procedure for the next node n2 with degree d2. The number of steps required to color all neighbors of node v2 is

$$(1+2+3+\ldots+d2)+d2=((d22+d2)/2)+d2=O(d22)\ 2\ldots\ldots\ldots\ldots\ \text{Equation 2}$$

In general, the number of steps required to color all the nodes in the neighbor list of any node ni with degree di and check if ni is port of clique or not is

$$((di^2+di)/2+di)=\ \ O(di^2)\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ \text{Equation 3}$$

Let the average degree of nodes be μ. Then the average number of steps required to color the neighbors of node *ni* with degree ρ is O (μ^2)

• Repeat the coloring procedure in steps 1 and 2 until all nodes are colored.

• Since each coloring step colors on the average ρ nodes, the coloring procedure will be repeated on the average (n/ρ), where n is the number of nodes.

• The total number of coloring steps required to color all nodes, on the average is

$$O((n/\mu).(\mu^2)=O(n.\mu).\ \ldots\ldots\ldots\ \text{Equation 4}$$

The complexity equation (above) can be mathematically expressed as

$$\sum = n\ i\ 1\ \mu\ ,\ \text{where}\ \mu=(\sum = n\ i\ i\ d\ 1\ )/n\ldots\ldots\ldots\ldots\ldots\ \text{Equation 5}$$

4.3.2. complexity of numeration algorithm

The major difference between proposed algorithm and numeration algorithm is that whenever a new node and all its neighbors are colored the algorithm will check the possibility of detection new clique, but in numeration algorithm the process of finding cliques in more complex such that the numeration algorithm works in stages before finding results these stages are add complexity to algorithm but adding also efficiency, these stages are branch and bound then uses backtracking to find solutions. the complexity of numeration algorithm is

$f(n)= O(n^3)$.

4.3.3. complexity of brute force algorithm

Brute force algorithm is differing than numeration algorithm and the proposed approach in using exhaustive search technique, this approach will look to all possible solution, suppose that we have graph with n number of nodes, then the probability of finding cliques is done by choose every node in graph and check the probability of adjacency to every other nodes in graph, subsequently the complexity of brute force algorithms equals factorial to size of graph $f(n)= O(n!)$.

4.4. performance analysis

In this thesis three algorithms have been presented Numeration algorithm, Brute force algorithm and Proposed approach, these algorithms have been tested on three different categories of data sets which are randomly generated graphs, benchmark graphs and simulated social network, this section covers the experiments and results of these three algorithms on variant categories of data sets, the following are data sets and expedients that conducted in each categorify:

4.4.1. Randomly generated graphs

this category of data set consist of graphs in size of {5, 10, ….. 40 } and for each size of graph there are three deferent densities ,subsequently the data set consist of 24 graphs variant in size and density, each graph will be ran by all of the three algorithms in total of 72 experiments.

After conducting the experiments the result will be shown in tables and charts, the following are nine tables show the 72 experiments that have been conducted on different randomly generated graphs, and after each table there are chart or more to clarify and analyze the results of experiments in table.

Table 0-1 the result of applying Numeration algorithm under Low density and different sizes of graphs ranging from 5 to 40 nodes

Table 0-1 Numeration algorithm results on low density graph

Criterial		Numeration algorithm		
Density	Number of nodes	Time milliseconds	Number of cliques	Maximum clique
Low	5	3.3	0	0
Low	10	20	2	3
Low	15	28.8	5	3
Low	20	37.6	23	4
Low	25	56.1	32	4
Low	30	75.5	49	4
Low	35	110.2	125	5
Low	40	116.5	156	5

This diagram show the relation between time required by Numeration algorithm and number of nodes to find cliques, the first experiment started with 5 nodes graph size and took 3.3 milliseconds. The maximum number of nodes was 40 while it took 116.5 milliseconds.

Figure 0-2 clarify the relation between time required by Numeration algorithm and number of nodes to find cliques,

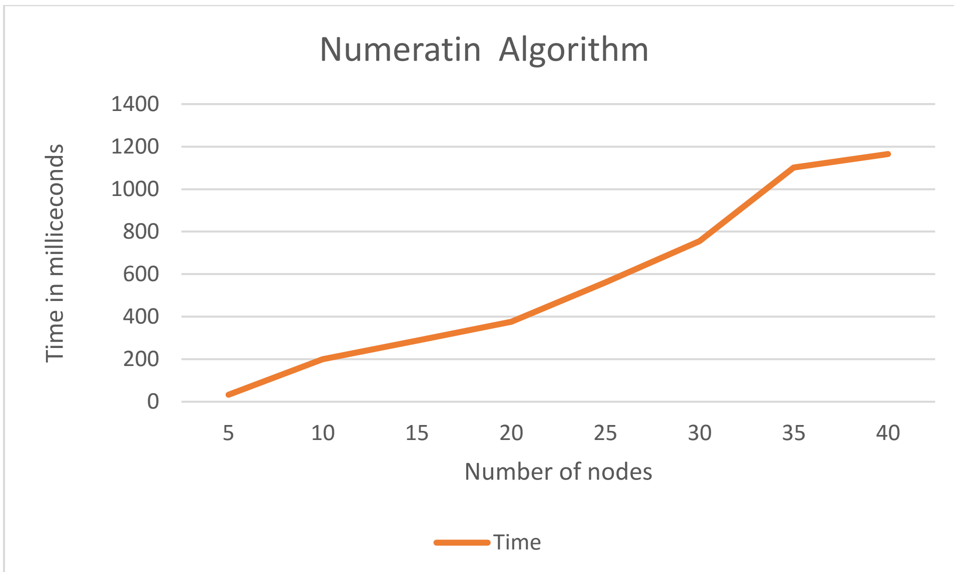


Figure 0-2 relation between size graph of and time needed to solve problem

Figure 0-3 clarifies the relation between size of graph and number of discovered cliques by Numeration algorithm, note that Figure 0-3

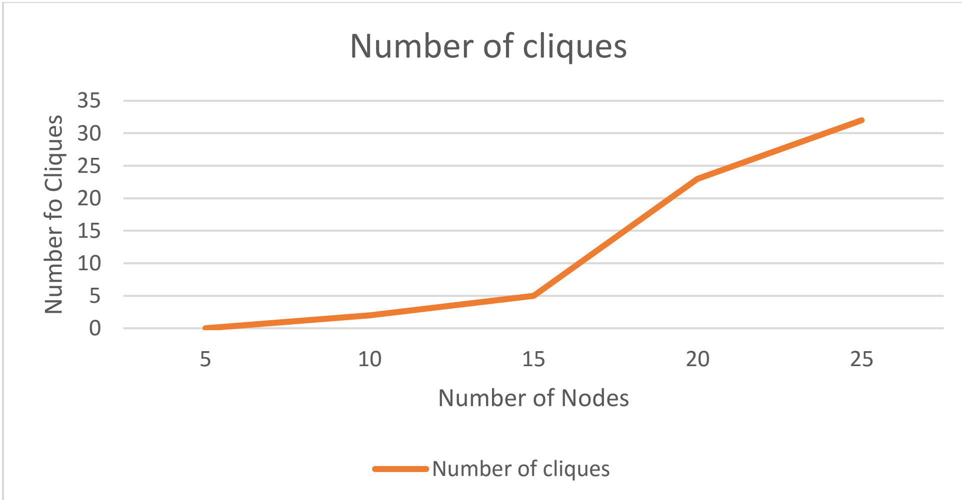


Figure 0-3 relation between size graph number of cliques discovered

Error! Reference source not found. shows the result using proposed algorithm under Low density graphs with different numbers of nodes.

Table 0-2 result of proposed algorithm in low density graphs

Criteria		Proposed algorithm			
Density	**Number of nodes**	**Time in ms**	**Number of cliques**	**Maximum clique**	**Colors used**
Low	5	15	0	0	4
Low	10	116	2	3	4
Low	15	167	4	3	4
Low	20	204	23	4	5
Low	25	351	32	4	5
Low	30	533	48	5	7
Low	35	617	128	5	7
Low	40	1002	159	6	8

Figure 0-4 clarifies the relation between size of graph and number of discovered cliques, using by proposed approach.

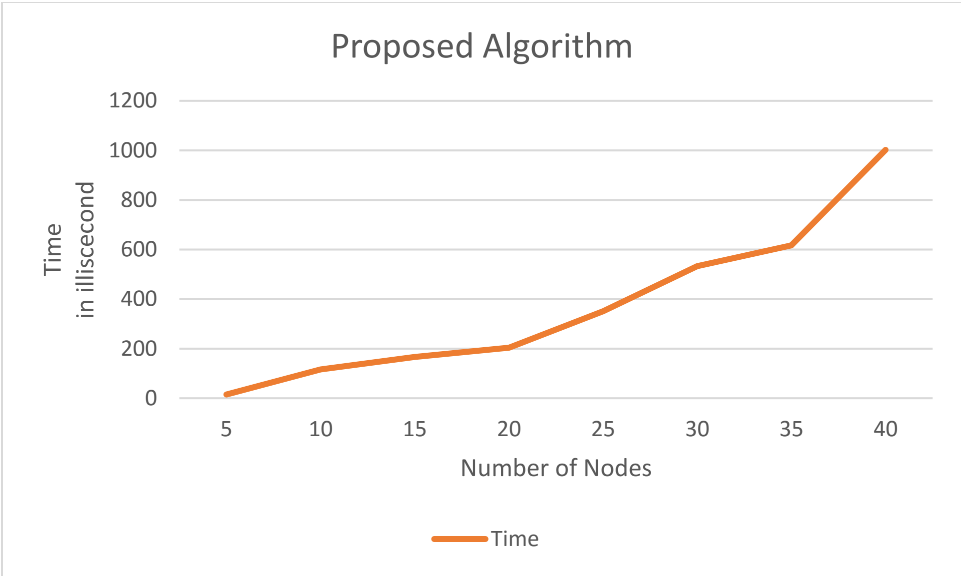


Figure 0-4 relation size of graph and required execution time

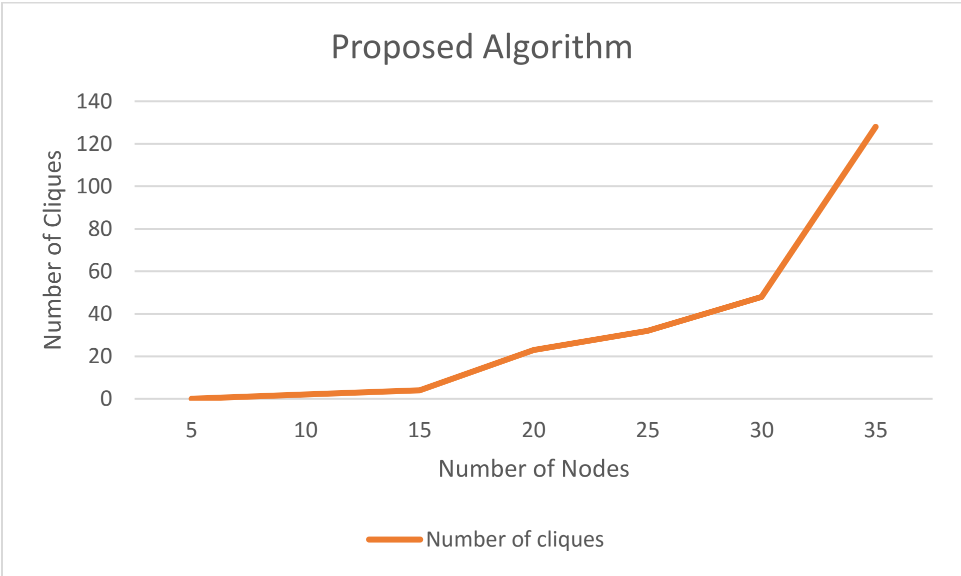


Figure 0-5 relation between number of nodes and time need to solve problem the following Table 0-3 shows the result using brute force algorithm under Low density with different numbers of nodes.

Table 0-3 the relation between the size of problem and time required to solve it by brute force

Criterial		**Brute force algorithm**		
Density	Number of nodes	Time in minutes	Number of cliques	Percentage of found cliques
Low	5	0.05488	0	100%
Low	10	0.12767	2	100%
Low	15	3.1	6	100%
Low	20	26	24	100%
Low	25	44	33	100%
Low	30	75	51	100%
Low	35	88.8	129	100%
Low	40	148.8	159	100%

In Figure 0-6 clarify the relation between by Numeration algorithm size of graph and number of discovered cliques, when the size of graphs is get bigger the of time to detect clique would be increased and also the number of discovered cliques

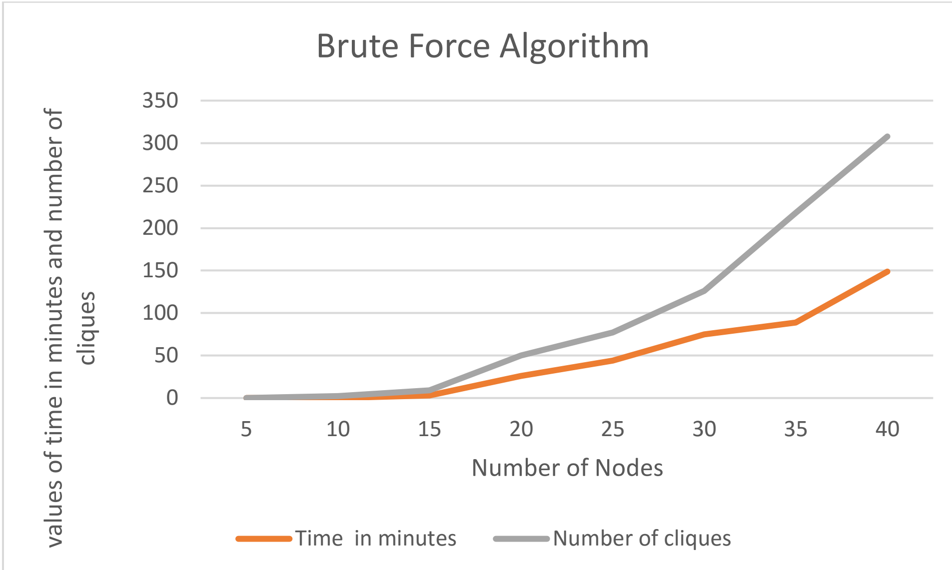


Figure 0-6 brute force algorithm performance on low density graphs

Table 0- show the result using Numeration algorithm under regular density graphs with different numbers of nodes

Table 0-4 numeration algorithm on regular graphs

Criterial		**Numeration algorithm**		
Density	Number of nodes	Unit Time	Number of cliques	Maximum clique
Regular	5	22	0	0
Regular	10	234	18	4
Regular	15	429	60	5
Regular	20	467	127	6
Regular	25	573	257	6
Regular	30	836	507	7
Regular	35	1335	798	7
Regular	40	1429	1272	7

On regular density data set we note that the when the size of graph is get larger time needed to find cliques is getting larger and size of clique also.

Figure 0-3 relation between size graph number of cliques discovered

Figure 0-7 clarifies the relation between by Numeration algorithm size of graph and number of discovered cliques,

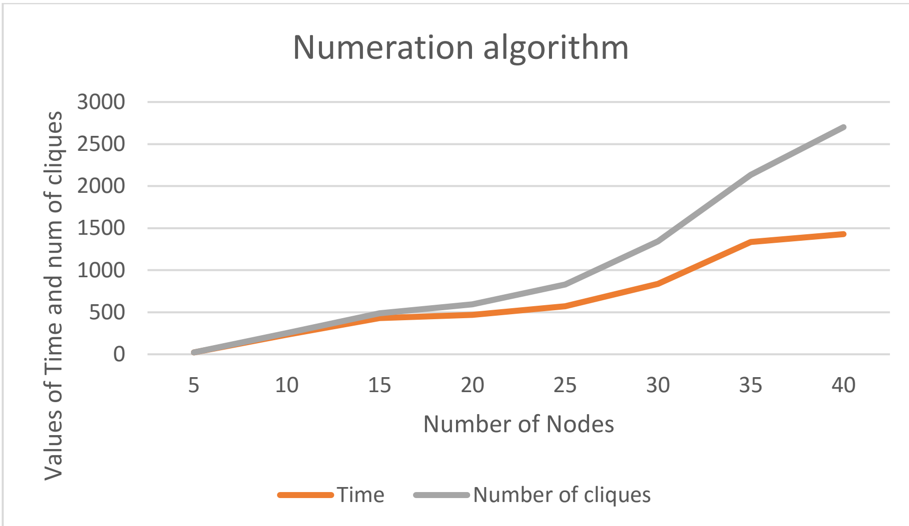


Figure 0-7 relation between size of graph and time required to solve the problem by numeration algorithm

the following Table 0- is show the result using proposed algorithm under regular density with different numbers of nodes.

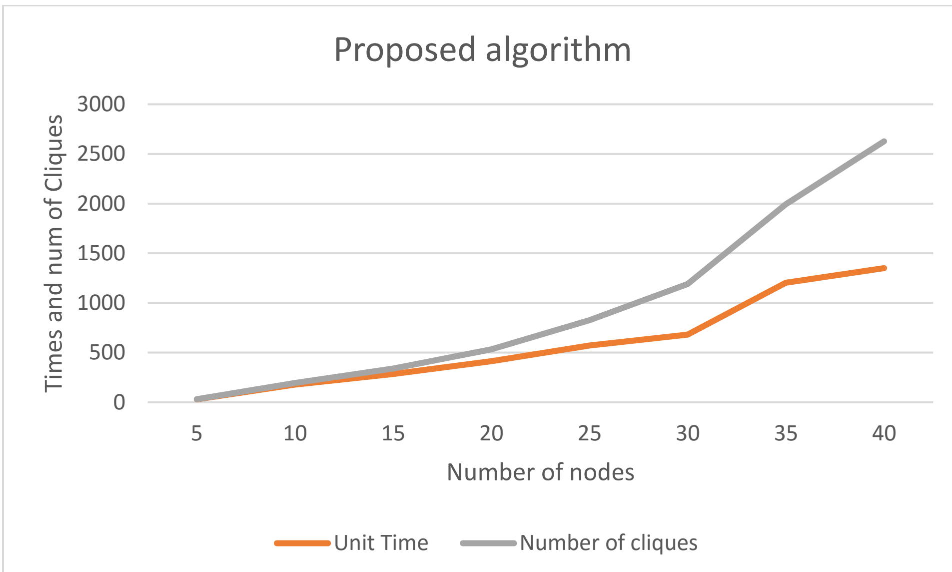
Table 0-5 6proposed approach results

Criterial		**Proposed approach**			
Density	Number of nodes	Unit Time	Number of cliques	Maximum Clique	Colored used
Regular	5	29	1	1	3
Regular	10	177	17	4	4
Regular	15	285	55	4	5
Regular	20	412	121	6	6
Regular	25	571	254	6	7
Regular	30	682	507	6	7
Regular	35	1204	790	6	9
Regular	40	1350	1277	8	14

the following table is show the result using brute force algorithm under regular density with different numbers of nodes

Figure 0-3 relation between size graph number of cliques discovered

clarify the relation between by Numeration algorithm size of graph and number of discovered cliques,



The Table 0-4 shows brute force algorithm results on regular density graphs the result shows that this algorithm has ability to find all cliques in graphs

Table 0-4 brute force algorithm running on regular graphs

Criterial		**Brute force algorithm**		
Ensity	Number of nodes	Time in Minutes	Number of cliques	Percentage of found cliques to lost ones
Regular	5	1	0	100%
Regular	10	1.2	19	100%
Regular	15	7	63	100%
Regular	20	37	128	100%
Regular	25	69.2	257	100%
Regular	30	133	510	100%
Regular	35	154.6	806	100%
Regular	40	204.7	1291	100%

The following table is show the result using Numeration algorithm under Heavy density with different numbers of nodes.

Figure 0-3 relation between size graph number of cliques discovered

clarify the relation between by Numeration algorithm size of graph and number of discovered cliques,

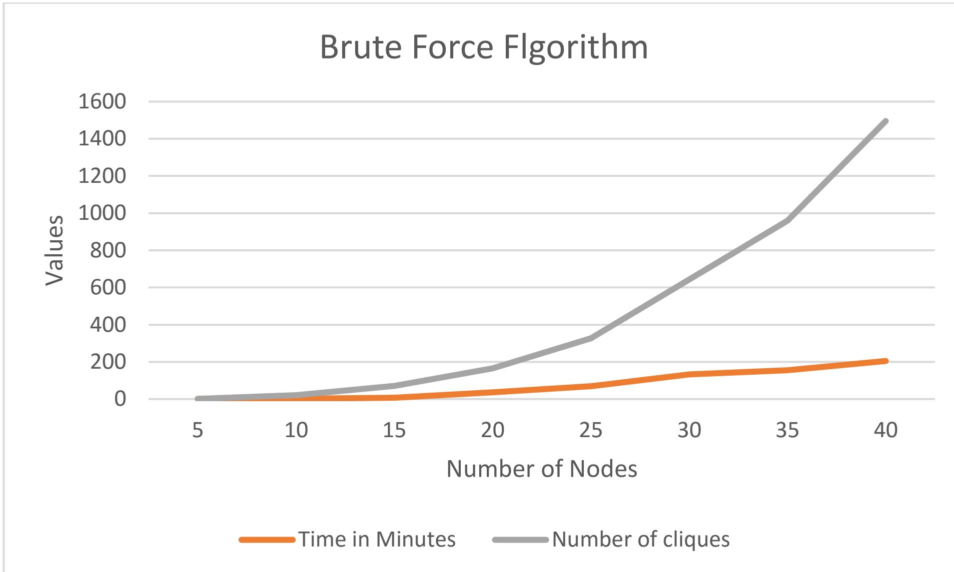


Figure 0-8 b

Table 0-5result of numeration algorithm on heavy density gash

Criteria		Numeration algorithm		
Density	Number of nodes	Time	Number of cliques	Maximum clique
Heavy	5	4	4	4
Heavy	10	19	56	6
Heavy	15	80	225	7
Heavy	20	504	341	9
Heavy	25	1132	967	11
Heavy	30	1997	2954	17
Heavy	35	2839	7552	22
Heavy	40	3673	9812	8

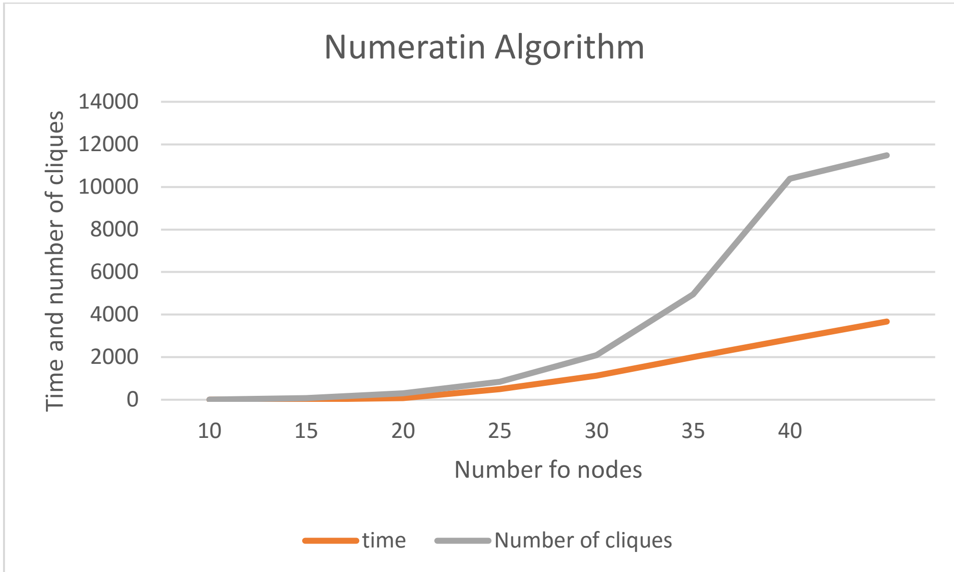
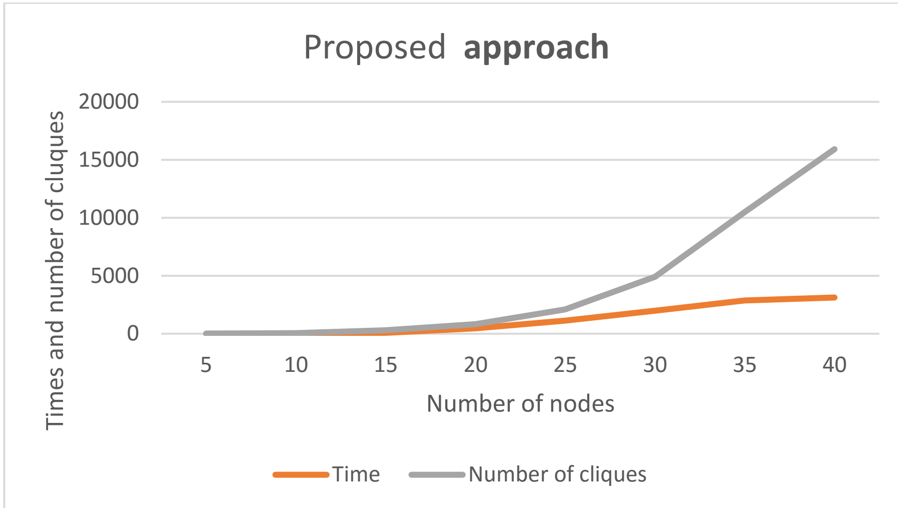


Figure 0-9 numeration algorithm works good in finding cliques in reasonable time

The following Table 0-6 shows the result using proposed approach under Heavy density with different numbers of nodes

Table 0-6 result of proposed approach on heavy density

criterial		**Proposed approach**			
Density	Number of nodes	Time	Number of cliques	Maximum clique	Cooled used
Heavy	5	4	4	4	3
Heavy	10	7	55	6	7
Heavy	15	66	224	7	9
Heavy	20	479	344	11	9
Heavy	25	1118	972	12	19
Heavy	30	1980	2942	21	25
Heavy	35	2860	7632	23	26
Heavy	40	3123	11790	25	26



.

Figure 0-10 proposed approach perforce on grapes

the following *Table 0-7* is show the result using brute force algorithm under Heavy density with different numbers of nodes

Table 0-7 8result using brute force algorithm under Heavy density

Criterial		**Brute Force Algorithm**		
Density	Number of nodes	Time in Minutes	Number of cliques	Percentage found cliques to lost ones
Heavy	5	3.3	4	100%
Heavy	10	9	56	100%
Heavy	15	18	226	100%
Heavy	20	66.7	351	100%
Heavy	25	134.2	981	100%
Heavy	30	204.1	2976	100%
Heavy	35	374	7604	100%
Heavy	40	509.3	12934	100%

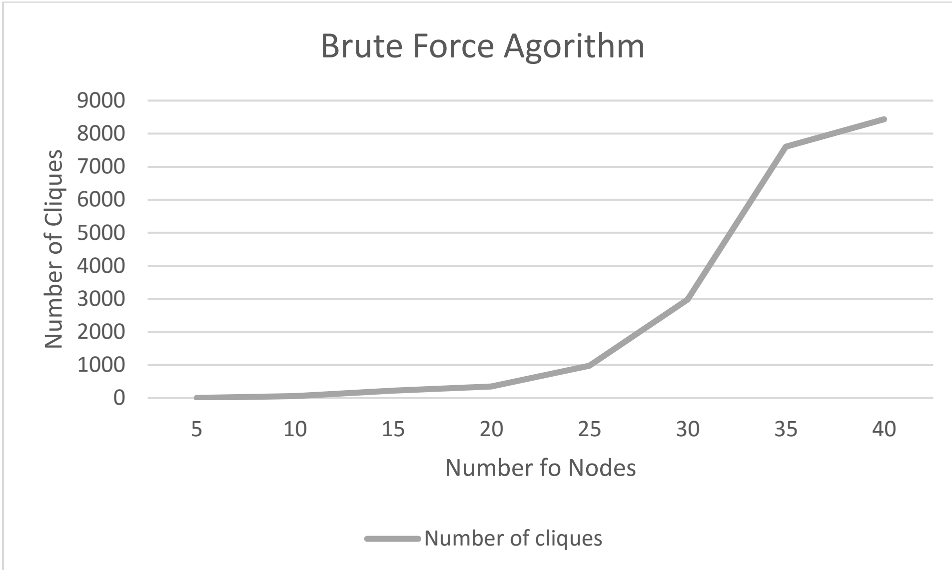


Figure 0-11 shows relation the size of graph and detected cliques by brute force algorithm

Table 0-8 comparison results of three algorithms on regular graphclarifies the relation between size of graph and time needed to find clique by brute force algorithm

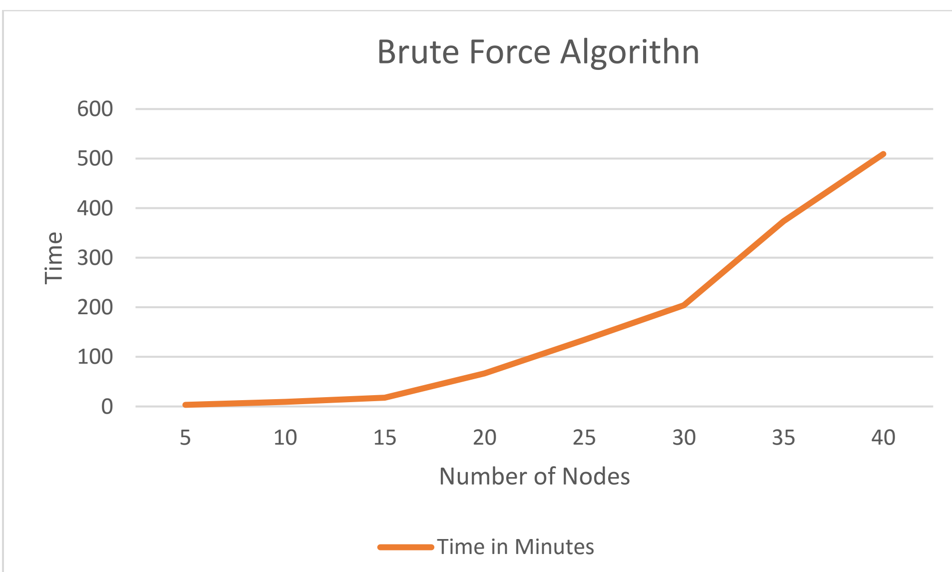


Figure 0-12 shows relation the size of graph and time needed to find clique by brute force algorithm

The nine tables show the results of experiments that have been conducted on different sizes and density graphs size of graphs start at 5 nodes and incremented by 5 nodes until reached 45 nodes graph size, and density is divided into three different densities Heavy, Regular and low, these variants of criterial enable us to examine the algorithms in variant cases.

Each of table shows the following parameters:

1. Density: (L, R, H): given as input to the program in the form of a probability.
2. The number of nodes: This is given as an input to the program.
3. Time required to finishing the process of finding cliques in each graph.
4. Number of detected cliques
5. Percentage of detected clique to lost ones
6. Number of colors used in coloring problem.

4.5. comparisons of algorithms results

After conducting the experiments of the three algorithms and many different graphs, a new comparison will be done according to time consumed to ruing the data set and accuracy of algorithm in finding cliques.

There are three table, each table include the results of three algorithms in same density and same size of graphs, the measure of each experiment is time consumed and number of detected cliques.

Error! Reference source not found. shows the comparison of results for the Numeration algorithm, brute force and proposed algorithm, on low density and 40 nodes graph, the comparison process done according to time consuming and number of detected cliques.

Table 0-9 results of comparison on heavy graph

Algorithm	Density	Number of nodes	Unit Time	Number of clique detected
Numeration	Low	40	2	156
Proposed	Low	40	1	159
Brute force	Low	40	148.8	159

Table 0-9 results of comparison on heavy graph show that the time complexity of brute force algorithm is very high comparing to the two other algorithms but the accuracy approximate equal.

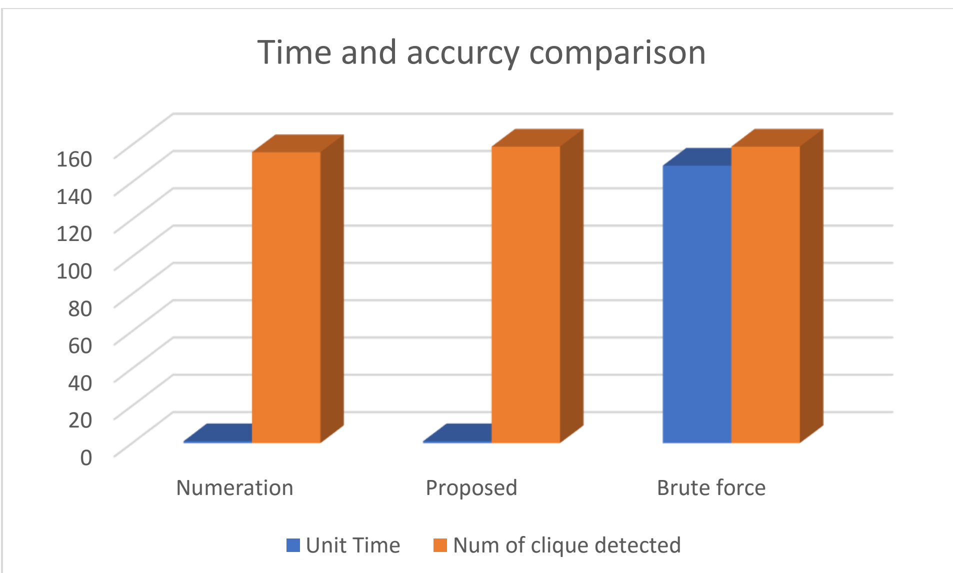


Figure 0-13 shows charts for result on low density graph

Table 0-8 shows the comparison of results for the Numeration algorithm, brute force and proposed approach, on regular density and 40 nodes graph size, the comparison process done according to time consuming and number of detected cliques.

Table 0-8 comparison results of three algorithms on regular graph

Algorithm	Density	Number of nodes	Unit Time	Number of detected clique
Numeration	Regular	40	4.4	1272
Proposed	Regular	40	3.8	1277
Brute force	Regular	40	2047	1291

Error! Reference source not found. show that the time complexity of brute force algorithm is very high comparing the two other algorithms but the accuracy approximate equal

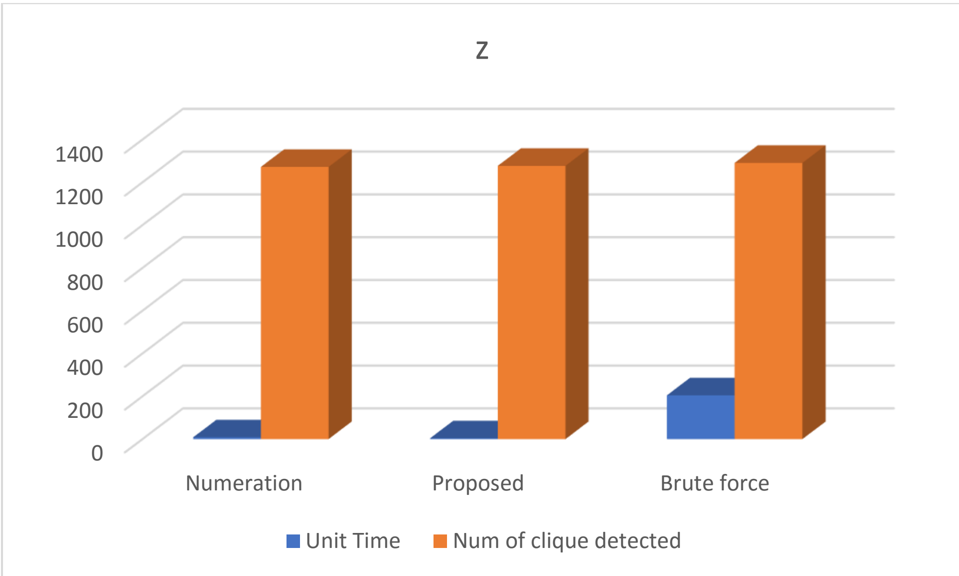


Table 0-9 results of comparison on heavy graph shows the comparison of results for the Numeration algorithm, brute force and proposed algorithm, on heavy density and 40 nodes graph, the comparison process done according to time consuming and number of detected cliques.

Table 0-9 results of comparison on heavy graph

Algorithm	Density	Number of nodes	Unit Time in minutes	Number of detected clique
Numeration	Heavy	40	14.9	9812
Proposed	Heavy	40	7.5	11790
Brute force	Heavy	40	509.3	12934

Figure 0-14 shows that the time complexity of brute force algorithm is very high comparing to the two other algorithms but proposed algorithm scored less complexity time on heavy graph and higher accuracy than numeration algorithm .

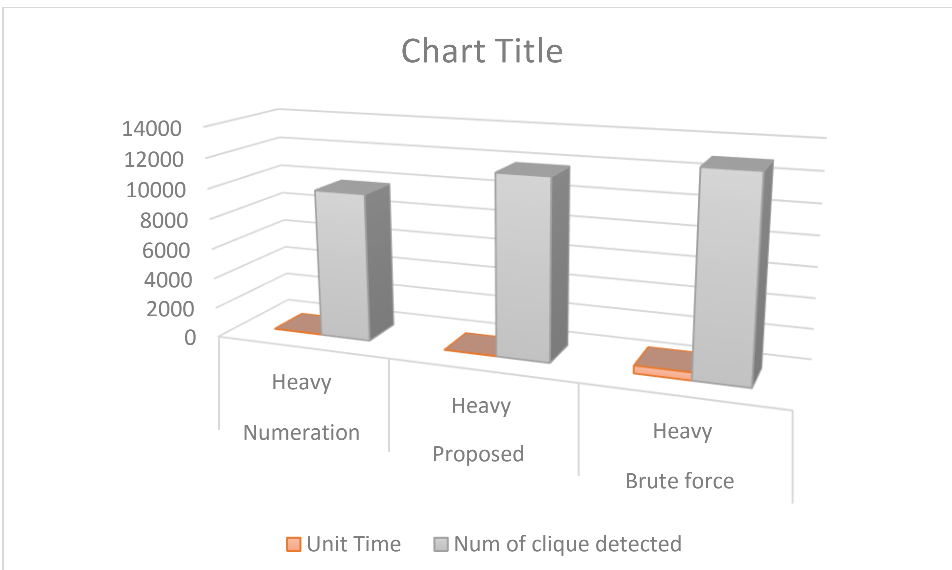


Figure 0-14 result of comparison on heavy graph

Chart shoes the three algorithms complexity, the proposed algorithm running in less time that the enumeration and brute force algorithms

In this section contains the results of applying the proposedapproach on benchmark dataset consist of five graphs variants in sizes,

4.6. Applying proposed approach on benchmark data set.

This category of data set considered as standard to prove the best performance and efficiency being achieved, a group of graphs will be used as data set to apply the proposed approach,

Five graphs gotten from benchmark on order to conduct the experiments by the proposed approach to find cliques, Table 0-10 shows the number of colors, number of cliques and execution time for the applying the proposed approach on five benchmark graphs

Table 0-10 results of running proposed approach on benchmark graphs

Number of nodes	Number of colors	Time in minutes	Number of cliques
450	88	0.13	1.1^{6}
595	101	0.16	1.5^{6}
760	167	0.22	2.1^{6}
1150	211	1..52	5.5^{6}
1272	279	2.3	7.1^{6}

4.7. Applying proposed approach on simulated social net works

Social networks differ than other problems that can be formulated as graph in that the social networks are very huge subsequently it needs a super powerful tool to deal with it, the main objective of this research to find smart approach to find cliques in simulated social networks, the following are five experiments of applying the proposed approach to finding cliques in simulated social networks of sizes between 10 000 profiles and 50 000 profiles, the results shows relatively promising to the huge data. Table 0-11 shows the results of applying the proposed approach on simulated social networks, the results shows that when the size of social networks gets bigger the number of cliques increases in exponentially, subsequently the time required to find cliques will increase in huge proportional and number of colors needed to color the social networks in increased in high range

Table 0-11 running proposed approach n simulated social networks

Number of accounts	Number of friendship relation	Number of colors used	Time in hour	Number of cliques
10000	22 675	508	0.18	5.8^{8}
20000	69712	2741	0.49	1.4^{9}
30000	193 975	5088	2.13	2.2^{10}
40000	311186	11490	3.51	7.8^{10}
50000	903102	17113	5.40	1.3^{11}

CHAPTER FIVE

CONCLUSIONS AND FUTURE WORKS

5.1. Conclusion

This research represents an attempt to utilize LDC algorithm to find cliques in large graphs such as social network by using graph coloring technique. The new approach represents a graph coloring using an algorithm, which colors a graph starting with the largest degree node in a particular graph

This research build on utilizing on what the researcher have been reached to enhance and support researchers around the world to find near optimal solution to this kind of finding cliques in large graphs.

This small sample of problems in our real life require huge graph to represent each one, and to deal with this graph we need a powerful algorithm, so that this thesis is introduced to add some contribution in solving cliques detecting in graphs.

This research consists of adding features to previous achieved work, our feature is to detect cliques using graph coloring algorithm, this new method showed promising results comparing to other brute force and numeration algorithm, the findings proved the performance of improved algorithm and running time is better than the other algorithms in the terms of time complexity. The results of a proposed approach have shown that there is enhancement in time complexity than other two approaches that used in comparison to proposed algorithm and the number of detected cliques.

5.2. Future work

1. applying suggested approach in real social networks, and use it as tools to analyze the behavior of social networks subscribers by data mining of subscriber's and activates such as analyzing his comments, pages subscribed to, grouping connected to common attributes.
2. Detecting and following the criminal people.
3. applying the new method in security analysis. Such as organized criminals1 `
4. Improving the new algorithm on order to be used as search engines by grouping similar objects in one cluster and the dissimilar objects far from each other.

REFERENCES

Almara'beh, H. & Suleiman, A. (2012).Heuristic algorithm for graph coloring based on maximum independent set. *Journal of Applied Computer Science & Mathematics*, *13*(6).

Akbulut, A., &Yılmaz, G. (2013).University Exam Scheduling System Using Graph Coloring Algorithm and RFID Technology. *International Journal of Innovation, Management and Technology*, *4*(1), 66.

Artacho, F. J. A., &Campoy, R. (2016). Solving graph coloring problems with the Douglas-Rachford algorithm. *arXiv preprint arXiv:1612.05026*.

Arya, S., & Dixit, M. A.(2014).A clustered based novel approach of graph coloring. *IOSR Journal of Computer Engineering (IOSR-JCE), 16(6),14-17*

Azizifard, N. (2013).Social Network Clustering. *International Journal of Information Technology and Computer Science (IJITCS)*, *6*(1), 76.

Alzubi, A., Al-Haj, M., &Malkawi, M. (2015). Maximum Clique Conformance Measure for Graph Coloring Algorithms. *International Journal of Computer Applications*, *127*(11), 32-37.

Carraghan, R., &Pardalos, P. M. (1990). An exact algorithm for the maximum clique problem. *Operations Research Letters*, *9*(6), 375-382.

Koyuncu, B., &Seçir, M. (2007). Student Time Table By Using Graph Coloring Algorithm. In *5th International Conference on Electrical and Electronics Engineering–ELECO*.

Jeff Pattillo, J. &Butenko, S. (2011).Maximum Clique, Maximum Independent Set, and Graph Coloring Problems. wiley encyclopedia of operations research and management science. doi: 10.1002/9780470400531.eorms0146.

Barnier, N., &Brisset, P. (2004). Graph coloring for air traffic flow management. Annals of operations research, 130(1-4), 163-178.

Caramia, M., &Dell'Olmo, P. (2001). Iterative coloring extension of a maximum clique. Naval Research Logistics, 48(6), 518-550.

Cauvery, N. K. (2011). Timetable scheduling using graph coloring. *International Journal of P2P Network Trends and Technology*, *1*(2), 57-62.

Chalupa, D. (2011). On the ability of graph coloring heuristics to find substructures in social networks. *Information Sciences and Technologies, Bulletin of ACM Slovakia*, *3*(2), 51-54.

Demange, M., Ekim, T., Ries, B., &Tanasescu, C. (2015). On some applications of the selective graph coloring problem. European Journal of Operational Research, 240(2), 307-314.

Donderia, V., & Jana, P. K. (2012).A novel scheme for graph coloring. Procedia Technology, 4, 261-266.

Hindi, M. M., &Yampolskiy, R. V.(2012, April).Genetic algorithm applied to the graph coloring problem. In *Proc. 23rd Midwest Artificial Intelligence and Cognitive Science Conf* (pp. 61-66).

Jin, Y., & Hao, J. K. (2016). Hybrid evolutionary search for the minimum sum coloring problem of graphs. *Information Sciences*, *352*, 15-34.

Kokosi´nski, K.,Domagała, P.(2014). Solving graph coloring problem with parallel evolutionary algorithms in a mesh model. *Federated Conference on Computer Science and Information Systems*, DOI: 10.15439/2014F391.

Koyuncu, B., &Seçir, M. (2007). Student Time Table By Using Graph Coloring Algorithm. In *5th International Conference on Electrical and Electronics Engineering–ELEC*.

Kroc, L., Sabharwal, A., & Selman, B. (2008). Counting solution clusters in graph coloring problems using belief propagation. In *Advances in Neural Information Processing Systems* (pp. 873-880).

Lai, X., Hao, J. K., & Glover, F. (2015). Backtracking based iterated tabu search for equitable coloring. Engineering Applications of Artificial Intelligence, 46, 269-278.

Lim, A., Zhu, Y., Lou, Q., & Rodrigues, B. (2005, March). Heuristic methods for graph coloring problems. In *Proceedings of the 2005 ACM symposium on Applied computing* (pp. 933-939). ACM,

Lü, Z., & Hao, J. K. (2010). A memetic algorithm for graph coloring. *European Journal of Operational Research*, *203*(1), 241-250.

Moalic, L., &Gondran, A. (2015).The New Memetic Algorithm HEAD for Graph Coloring: An Easy Way for Managing Diversity.In *Evolutionary Computation in Combinatorial Optimization* (pp. 173-183). Springer International Publishing.

Malkawi, M.,Hassan, M. A. H., & Hassan, O.A.H.(2008). A New exam scheduling algorithm using graph coloring, *International Arab Journal of Information Technology*,*5*(1), 80-86.

Mittal, *A.* ,Nogja, *A.(2014).*Analysis and Application of Genetic Algorithm in context of Graph Coloring. I*nternational Journal of Scientific Research and Engineering Studies (IJSRES).1(1).*

Mokken, R. J.(1979).Cliques, Clubs and Clans. *Elsevier Scientific Publishing Company*, 13(1979) 161-173.

Gondran, A. (2014).Variations on Memetic Algorithms for Graph Coloring Problems., Vehicular Technology, IEEE Transactions .52(4),

Montemanni, R., Moon, J. N., & Smith, D. H. (2003).An improved tabu search algorithm for the fixed-spectrum frequency-assignment problem. Vehicular Technology, IEEE Transactions on, 52(4), 891-901.

Pattillo, J., &Butenko, S. (2011).Clique, independent set, and graph coloring. *Encyclopedia of Operations Research and Management Science*, 3150-3163.

Stix, V. (2004).Finding all maximal cliques in dynamic graphs. *Computational Optimization and applications*, *27*(2), 173-186.

Shen, J. W. (2003). Solving the graph coloring problem using genetic programming. *Genetic Algorithms and Genetic Programming at Stanford*, 187-196.

Shiokawa, H., Fujiwara, Y., &Onizuka, M. (2015). SCAN++: efficient algorithm for finding clusters, hubs and outliers on large-scale graphs. *International Journal of Information Processing & Management*;*8*(11), 1178-1189.

Taylor, D.,Weber,B. &Bojduj, B.(2006).A tabu search framework for dynamic combinatorial optimization problems. *Proceedings, Integrated Design and Process Technology, IDPT-2006*, 663-668.

Yan, B.,& Gregory, S. (2009, November).Detecting communities in networks by merging cliques. In *Intelligent Computing and Intelligent Systems, 2009. ICIS 2009. IEEE International Conference on* (Vol. 1, pp. 832-836). IEEE.

Yang, J., McAuley, J., &Leskovec, J. (2013, December). Community detection in networks with node attributes. In *Data Mining (ICDM), 2013 IEEE 13th international conference on* (pp. 1151-1156). IEEE.

Wang, J., & Xia, Y. (2012). Fast graph construction using auction algorithm.arXiv preprint arXiv:1210.491