INVESTIGATION OF MODULARITY MAXIMIZATION USING MATHEMATICAL PROGRAMMING

BY

ZEAD HAMED SALEH

A Thesis Presented to the
DEANSHIP OF GRADUATE STUDIES

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS
DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

In

INDUSTRIAL & SYSTEMS ENGINEERING

DECEMBER 2019

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN- 31261, SAUDI ARABIA

DEANSHIP OF GRADUATE STUDIES

This thesis, written by **Zead Saleh** under the direction of his thesis advisor and approved by his thesis committee, has been presented and accepted by the Dean of Graduate Studies, in partial fulfillment of the requirements for the degree of **MASTER OF SCIENCE IN INDUSTRIAL & SYSTEMS ENGINEERING.**

Dr. Harun Pirim
(Advisor)

Dr. Hesham K. Al-Fares
Department Chairman

Dr. Hesham K. Al-Fares
(Member)

Dr. Salam A. Zummo
Dean of Graduate Studies

عمادة الدراسات العليا
DEANSHIP OF GRADUATE STUDIES

Dr. Abdul Qadar Kara
(Member)

23/12/19
Date

This work is dedicated to my loving parents, wife, my brother and my sisters without whom after god’s help this work wouldn’t have been possible.

ACKNOWLEDGMENTS

I would first like to thank my thesis advisor Dr. Harun Pirim. The door to Dr. Pirim office was always open whenever I ran into a trouble spot or had a question about my research or writing. He consistently allowed this paper to be my own work but steered me in the right direction whenever he thought I needed it.

Also, I must express my very profound gratitude to my parents, wife, brother, sisters, friends and colleges for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis.

I would also like to thank my thesis committee members: Dr. Hesham K. Al-Fares, Dr. Abdul Qadar Kara for their remarks that led to enhance the quality of this research.

And finally, I want to thank the department of industrial and system engineering at KFUPM for giving me the chance to continue my graduate studies and obtain my master’s degree.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

L_c	The longest shortest path in different clusters
d_{ij}	The distance between the nodes i and j
x_{ic}	Binary variable that will be equal to 1 if node i belongs to cluster c and 0 otherwise
x_{jc}	Binary variable that will be equal to 1 if node j belongs to cluster c and 0 otherwise
x_{jk}	Binary variable that will be equal to 1 if node j belongs to cluster k and 0 otherwise
m_c	The number of connections inside cluster c
n_c	The number of nodes inside cluster c
a_{ij}	The elements of adjacency matrix for graph G
y_{jc}	Binary variable equals to 1 if node j belongs to cluster c and 0 otherwise
y_{jc}	Binary variable equals to 1 if node i belongs to cluster c and 0 otherwise

$\|V\|$	Total number of nodes in graph G.
$\|c\|$	Total number of nodes inside cluster c.
w_{ijc}	Linearization variable
α_c	Objective function continuous variable of MILP
S_{ic}	Linearization variable
X_c	Objective function continuous variable of upper bound
U_α	Upper bound variable
L_α	Lower bound variable
DR_{inter}	Inter Density Ratio
DR_{intra}	Intra Density Ratio
e_i	Number of out connections for node i
a_i	Number of in connections for node i
$\|c_l\|$	Number of nodes in cluster l
$\|c_k\|$	Number of nodes in cluster k.
$\|E\|$	Total number of edges in a graph G.
I	A set of initial solutions for heuristic algorithm

i, j	Indices for nodes
j, k, l	Indices for clusters
$I(p)$	Specific initial solution

ABSTRACT

Full Name : [Zead Hamed Abdul Jaleel Saleh]

Thesis Title : [Investigation of Modularity Maximization Using Mathematical Programming]

Major Field : [Industrial Engineering]

Date of Degree : [December 2019]

Networks science is one of the most considerable research areas in social, natural and computer sciences as well as engineering. Most networks have vertices organized in groups called communities, modules or clusters. Communities are groups of vertices which probably share similar properties and/or play common roles within a graph. Modularity maximization is one of the most popular approaches in community detection. However, modularity maximization solution has practical problems such as resolution limit and degeneracy. Recently, an alternative clustering measure called modularity density has been proposed to overcome the resolution limit of modularity maximization.

Modularity Density Maximization (MDM) aims to reduce the out links between clusters. So, the less out connections are the better. In this research, the out connections are perceived as a distance. Thus, we propose a Modified Modularity Density Maximization (MMDM) as we consider minimizing the deep out connection instead of minimizing the out links. Modified Modularity Density Maximization (MMDM) is formulated as a Mixed Integer Linear Programming (MILP). The model is solved by GAMS software and the obtained results are compared with MDM using internal cluster validation approach.

A clustering heuristic algorithm named Density Radio Heuristic DR is proposed to solve larger data sets that cannot be solved by MILP or take very long time to be solved. The heuristic is applied on both MMDM and MDM approaches and the obtained results are compared using internal cluster validation approach.

ملخص الرسالة

الاسم الكامل: زياد حامد عبدالجليل صالح

عنوان الرسالة: استقصاء حول تحقيق أعلى قدر من النمطيه بواسطة استخدام البرمجه الرياضيه

التخصص: هندسة صناعية

تاريخ الدرجة العلمية: ديسمبر 2019

علوم الشبكات هي واحدة من أهم مجالات البحث في العلوم الاجتماعية والطبيعية وعلوم الكمبيوتر وكذلك الهندسة. تحتوي معظم الشبكات على رؤوس منظمة في مجموعات تسمى المجتمعات أو الوحدات أو المجموعات. المجتمعات هي مجموعات من القمم التي تشترك على الأرجح في خصائص متشابهة و تلعب أدوارًا مشتركة في الرسم البياني. تحقيق اعلى قدر من النمطيه هو احد أكثر الطرق شيوعا في اكتشاف المجموعات، لكن تحقيق اعلى قدر من النطيه لديك مشاكل في الحلول مثل قرار الحد والتنكس.

في الآونة الأخيرة ، تم اقتراح بديل لتجميع الوحدات في مجموعات بما يسمى كثافة الوحدات للتغلب على الحد الأقصى لقرار زيادة الحد الأقصى. سيحاول تعظيم كثافة الوحدات (MDM) تقليل الروابط الخارجية بين المجموعات. لذلك كلما قل عدد الروابط الخارجية بين المجموعات كلما كان أفضل. في هذا البحث يمكن اعتبار الروابط الخارجية بين المجموعات كنوع من المسافه. لذالك اقترحنا تعديل كثافة الوحدات (MDMM) عندما فكرنا بأن بتقليل الاتصال العميق بين المجموعات بدلاً من الروابط الخارجيه. تعديل تحقيق أعلى قدر من كثافة المجموعات (MDMM) تمت صياغته كبرمجة خطية عددية مختلطه. وتم حل النموذج الرياضي بإستخدام برنامج (GAMS) ، ثم تمت مقارنة النتائج مع (MDM) من حيث نهج التحقق من جودة تقسيم المجموعات.

بالإضافة إلى ذلك ، اقترحنا خوارزمية لاكتشاف المجموعات وسميناها خوارزمية نسبة الكثافه (DR) . لقد كان الغرض الرئيسي من هذه الخوارزمية هو حل مجموعات البيانات الكبيرة التي لا يمكن حلها بواسطة MILP أو يستغرق وقتًا طويلاً للغاية لحلها .تم تطبيق هذه الخوارزمية كل من النهج MMDM و MDM وتمت مقارنة النتائج التي تم الحصول عليها في مصطلح التحقق من جودة تقسيم المجموعات.

CHAPTER 1

INTRODUCTION

1.1 Background

Network science stands out as one of the most fruitful research fields which has applications in social, natural and computer sciences as well as engineering. Graphs, or networks, originally are a set of vertices connected by links called edges. Networks occur in a huge diversity of contexts. For example, Twitter, Facebook and Instagram are large social networks, where millions if not billions of people are connected through virtual acquaintanceships. Another example is the internet used in computers connected thorough cables or wireless signals. Many other applications are in physics biology, engineering, economics, ecology, computer science, marketing, political and social sciences, etc. Most networks have vertices organized in groups called communities, modules or clusters. Communities are groups of vertices which probably share similar properties and/or play common roles within a graph. For example, communities could represent proteins with a similar function in protein–protein interaction networks and groups of friends in social networks, websites on similar topics on the Web graph, and so on [1]. Identifying clusters in networks can give an indication on how they are organized. Clustering helps to group the nodes, support their role with regard to the communities they belong in an unsupervised learning fashion. Nevertheless, clustering or community detection in networks is a nebulous problem. There is no comprehensive definition or clear-cut guidelines on how to compare different community detection algorithms and judge their performance. On one hand, such opacity, open the freedom to propose varied approaches to the problem, which often depend on the specific research question and (or) the particular system at study. Modularity maximization is one of the most popular approaches in community detection. Modularity can be defined as a quality metric that measures the difference between the actual density of edges within the cluster and the density of the subgraph in a randomized graph with equivalent number of nodes and edges [12]. However, modularity maximization solution has practical problems such as resolution limit and degeneracy [1]. Recently, an alternative clustering measure called modularity density has been proposed to overcome the resolution limit of modularity maximization. So, modularity maximization and modularity density maximization, standing as one of the most reviewed clustering methods, will be the base of our work. We propose that there is a gap in operations research literature especially for modularity density maximization and conduct the problem statement in the next section.

1.2 Problem Statement

Given a graph $G = (V, E)$ where V is the set of vertices and E is the set of edges. Forming subgraphs having similar properties, based on particular criteria, is called community detection. One of the most popular criterion of community detection is via modularity maximization. However, modularity maximization has some issues such as resolution limit and degeneracy. Resolution limit is the possibility of not detecting communities that are smaller than a scale, which depends on the network size and interconnection between clusters, because they could be merged with other lager communities [59]. Modularity function Q is also suffering from degeneracies which is the difficulty of finding an optimal solution because of the existence of many near optimal solutions [60]. To overcome this issue a new measure called modularity density maximization is used. Much research has been done in this area. However, there are some gaps in operations research literature. Modularity Density Maximization (MDM) aims to reduce the out links between clusters. So, the less out connection the better it is. The out connections can also be perceived as a distance. Thus, in our model we consider minimizing the deep out connection instead of minimizing the out links. For example, we don't want someone to be deeply rooted to someone else in another cluster because we want them to be in separate clusters. Moreover, we observe in the literature that modularity density maximization solution can have negative values for some communities which indicates a weak definition of MDM. Thus, we notice that most papers on MDM talk about the scale and computational time, however MDM is still not proved to result in accurate community detection. Hence, the problem is that the community information being hidden inside the network representation and there is always a need to extract different communities which maybe closer to ground-truth ones. So, developing mathematical programming formalism to address community detection more accurately for some scenarios and also solution methodologies under the formalism will result in better outcomes for some data sets.

1.3 Research Objectives

The main objective of this work is to provide a Modified Modularity Density Maximization (MMDM) approach that can address community detection more accurately. In the first part we formulate the problem as MILP and solve the model optimally for some data sets. In the second part we propose a heuristic that can solve larger instances that cannot be solved by MILP or take a long time to solve. Lastly, we bench mark the obtained results of Modified Modularity Density Maximization (MMDM) With Modularity Density Maximization (MDM).

The milestones of this research are the following:

1. Reinvestigate current MDM formulations and column generation approaches to apply on different data sets.
2. Investigate MILP formulations of modularity density maximization to modify objective function and/or add new constraints that may reflect some facts in the used data sets.
3. Propose a heuristic to solve larger data set of the proposed model.
4. Validating the results of the new model by applying some clustering quality metrics.

1.4 Approach

To achieve our objectives, the work will be divided into steps as following:

1. Real data sets can be collected from many popular network data sets website such as SNAP. Moreover, we can simulate datasets.
2. Applying optimization solution techniques for ILP and MILP problems.
3. Use optimization solvers such as CPLEX and BONMIN to solve the mathematical formulation.
4. Use one of most common programing languages such as python to code our heuristic algorithm.
5. Comparison between modified modularity density maximization MMDM with modularity density maximization MDM in term of cluster validation.
6. Concluding and proposing future studies.

CHAPTER 2

LITERATURE REVIEW

The purpose of this chapter is to present the literature related to community detection in networks. First, we review older methods proposed to detect communities such as hierarchal, spectral, divisive and dynamic clustering. Then we cover the literature of detecting communities through modularity maximization which is one of the most popular methods. Finally, we introduce the literature that focuses on modularity density maximization which is one of the most recently studied community detections metric.

2.1 Traditional Methods

2.1.1 Graph Partitioning

Graph partitioning problem is dividing the vertices into k clusters of predefined size, in a way that the number of edges lying between the clusters is minimal. These number of edges connecting the different clusters are called cut size. Imposing a partition with the minimal cut size without specifying the number of clusters gives a trivial solution, so it is very important to constraint on the number of clusters. Most variants of the graph partitioning problem are NP-hard [2]. Nevertheless, there are many algorithms that can partition a network efficiently, even if their solutions are not optimal [3]. Most of these algorithms depend on algebraic, geometric and multilevel ideas. One of the earliest proposed algorithms is the Kernighan-Lin algorithm [4]. This algorithm is still used because of its low computational time. Kernighan-Lin algorithm is extended to get partitions for any number of parts [5]. However, the run-time and storage costs increase rapidly with the number of clusters. Maximum flows concept was used by Flake et al [6] to identify clusters. Although these partitioning algorithms are efficient, they have certain ambiguities in clustering such as requiring that the number of clusters and their sizes are known in advance.

2.1.2 Hierarchal Clustering

Hierarchical clustering is a widely used tool in data analysis. The idea of this method is to build a binary tree of data that combine similar groups of points [3]. As other clustering techniques, hierarchal clustering intents to find a group of vertices having high similarity. Hierarchal clustering can be classified in two categories [1]:

1. Agglomerative algorithms are bottom-up techniques, where at the beginning every object belongs to an individual cluster and then clusters are iteratively merged based on their high similarity until all objects form a single cluster;
2. Divisive algorithms, are top-down techniques where at the beginning all objects belong to a single cluster and then clusters are iteratively split by removing edges connecting vertices with low similarity [1] until every object belongs to an individual cluster.

2.1.3 Partitional Clustering

In partitional clustering the goal is to separate objects in to k communities such that a given cost function of distance from points to centroid or between points is maximized/minimized, as in reference [1]. The most popular partitional technique in the literature is k-means clustering [7]. The idea of this algorithm depends on starting with a k groups and each group consists of only one random point. Then iteratively assign each new data point to the cluster with the nearest mean and the mean of each new group is recalculated. This procedure is repeated until no points change cluster memberships. There are many variations of k-means algorithm such as fuzzy k-means [8].

2.1.4 Spectral Clustering

In 1973 Donath and Hoffmann [9] propose the first algorithm on spectral clustering. They use eigen vectors of the adjacency matrix to partition the graph. Spectral clustering makes use of eigen values of the similarity matrix of the data. The similarity matrix is provided as an input and consists of a quantitative assessment of the relative similarity of each pair of points in the dataset [3]. Andrew Y. Ng et. al [10] propose a particular manner to use the *k* eigenvectors simultaneously. They also present the conditions when the algorithm will perform efficiently.

2.2 Divisive Algorithms

Detecting edges which connect vertices of different clusters and remove them to disconnect clusters from each other is the essence of divisive algorithms [1]. Girvan and Newman [11], [12] proposed one of most popular algorithms using divisive approach. This algorithm initially removes edges from the network by using one of edge betweenness measures, such as standard shortest path betweenness of Freeman [1]. The second step is called recalculation step in which betweenness scores are re-evaluated after the removal of an edge [12].

2.3 Spectral Algorithms

Slanina and Zhang [13], through empirical study, have shown that if the graph has a clear community structure, then one can localize eigen vectors of the adjacency matrix. Capocci et al [14] use a spectral technique to detect communities in directed graphs. They have converted the directed graph into undirected weighted graph and performed the analysis. Alves [15] uses eigen values and eigen vectors of the Laplacian matrix to compute the effective conductance for pairs of nodes in a graph.

2.4 Dynamic Algorithms

2.4.1 Spin Models

Reichardt and Bornholdt [16] propose an algorithm by combining the idea of graph bi-partitioning by Fu and Anderson with a modified Ising Hamiltonian and Potts model clustering of multivariate data by Blatt et al. [17]. They alter a q-state Potts Hamiltonian by adding a global constraint that forces the spins into communities.

2.4.2 Random Walk

Hughes [18] shows that random walk can be useful to detect the clusters in a graph. However, if a graph consists of many communities, a random walker spends a long time inside a community due to the high intra-connections among all the vertices. One of the advantages of random walk algorithms is that it can be easily extended for weighted graphs. Zhou and Lipowsky [19] use biased random walkers, where the bias happens to the fact that walkers usually move towards the nodes sharing a large number of neighbors with the starting node in a graph. A proximity index is defined to show that how much a pair of nodes is closer to all other nodes in the graph. The procedure is called Net Walk and is used to detect the communities in a graph, where Net Walk is a hierarchical clustering method, and the proximity defines the similarity.

2.5 Modularity Maximization

Modularity (Q) maximization is one of the most popular community detection methods. Modularity is one of the quality metrics that measures the difference between the actual density of edges within the cluster and the density of the subgraph in a randomized graph with equivalent number of nodes and edges. Modularity is based on the idea that the actual subgraphs should have more links between themselves than a random one. Thus, when the value of Q is close to 1 it means the nodes in the community is highly connected. On the other hand, Q closing to 0 indicates that the fraction of edges inside communities is no better than the random case [20]. According to M. E. J. Newman and M. Girvan [12] modularity can be defined as $Q = \sum_i (e_{ii} - a_i^2)$ where $a_i = \sum_j e_{ij}$ represents the fraction of edges that connect to vertices in community i and $e_{ij} = a_i a_j$ is the fraction of all edges in the network that link vertices in community i to vertices in community j. So, modularity can be represented as

$$Q = \sum_{c_i \in C} \left[\frac{\left|E_{c_i}^{in}\right|}{|E|} - \left(\frac{2\left|E_{c_i}^{in}\right| + \left|E_{c_i}^{out}\right|}{2|E|} \right)^2 \right] \quad \text{(2-1),}$$

where C is the set of all the subgraphs, and |E| is the total number of edges in the network, c_i is a specific cluster in C, $E_{c_i}^{out}$ is the number of edges from the nodes in cluster c_i to the nodes outside c_i, $E_{c_i}^{in}$ is the number of edges between nodes within cluster c_i. Modularity also can be represented, according to M. E. J. Newman [21] , as

$$Q = \frac{1}{2|E|} \sum_{ij} \left[A_{ij} - \frac{k_i k_j}{2|E|} \right] \delta_{c_i, c_j} \quad \text{(2-2),}$$

where A_{ij} is an element belonging to row i column j of the adjacency matrix, k_i is the degree of node i, c_i is the label of the community to which node i is assigned δ_{c_i,c_j} is the Kronecker delta symbol where $\delta_{c_i,c_j} = 1$ if i and j are in the same cluster and $\delta_{c_i,c_j} = 0$ otherwise. To illustrate the concept of modularity we give a numerical example and show how we can calculate the Q value. Consider a network with 8 nodes and 19 edges as in figure 2-1. Using fast greedy approach in RStudio we get a graph partition of two communities as in figure 2-2. The set of the blue nodes represents community one and the orange represents community two. According to expression (2-1) we calculate the Q as: $Q = \sum_{c_i \in C} \left[\frac{|E_{c_i}^{in}|}{|E|} - \left(\frac{2|E_{c_i}^{in}| + |E_{c_i}^{out}|}{2|E|} \right)^2 \right] = \left[\left(\frac{3}{19} - \left(\frac{(2)(3)+8}{38} \right)^2 \right) + \left(\frac{8}{19} - \left(\frac{(2)(8)+8}{38} \right)^2 \right) \right] = 0.0221$ + 0.0222= 0.0443. For expression (2-2) $Q = \frac{1}{2|E|} \sum_{ij} \left[A_{ij} - \frac{k_i k_j}{2|E|} \right] \delta_{c_i,c_j}$ e.g. for $i = 1$ and $j = 3$, $\left[A_{13} - \frac{k_1 k_3}{2|E|} \right] \delta_{2,2,} = 1 - \frac{(6)(6)}{(2)(19)} (1) = \frac{1}{19}$. We do same procedure for all pairs of nodes and sum up all the results to get the value of Qwhich is equal to 0.0443 in this example.

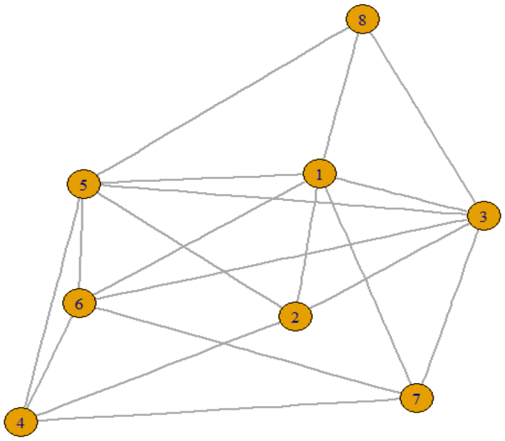


Figure 2-1 A Toy Graph of 8 Nodes and 19 Edges

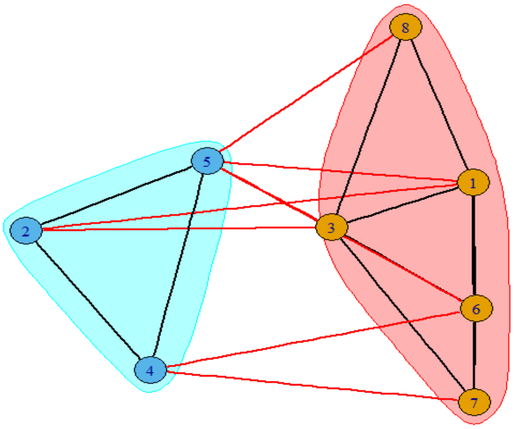


Figure 2-2 Community Detection via Fast Greedy Approach

The modularity formulations above are suitable only for unweighted and undirected graphs. modularity can be modified and applied for weighted and directed graphs. The modified definition of modularity for directed networks is as follows [22]: $Q = \frac{1}{|E|}\sum_{ij}\left[A_{ij} - \frac{k_i^{in}k_j^{out}}{|E|}\right]\delta_{c_i,c_j}$ (2-3) where and k_j^{out} and k_i^{in} are the out- and in- degrees. For weighted graphs someone can apply the same general techniques of un weighed graphs by mapping weighted networks onto multigraphs [23]. As mentioned earlier, high value of modularity (Q) indicates a good community structure so it is natural to optimize Q in a maximizing fashion. However, one of the disadvantages of this optimization is that it requires high computational time because of the large number of possible partitions specially in complex networks. Modularity optimization is shown to be NP hard [24]. Therefore, many heuristic methods are proposed to find high-modularity partitions in a reasonable time. These methods are as follows:

2.5.1 Greedy Algorithms

Newman [25] proposed the first greedy algorithm based on agglomerative hierarchical clustering method. Initially, every node is considered as a cluster, creating altogether $|V|$ clusters. Next, the algorithm merge pairs of clusters that form the largest modularity. Then continue repeating this step until all the nodes in the network are in a single community after $(|V| - 1)$ steps of merging. One of the advantages of Newman's algorithm [25] is that it is much faster than the algorithm of Newman and Girvan [12]. However, when the network is sparse its matrix will have a lot of zeros involving many unnecessary operations to update the adjacency matrix at each step has. Clauset et al. [26] introduce special data structures to perform better matrix updating for sparse matrices. However, greedy algorithm proposed by Clauset et al. is not applicable for networks larger than 500,000 nodes. Wakita and Tsurumi [27] discover that merging clusters in an unbalanced manner, which yields very unbalanced dendrograms, is the reason behind this limitation. The paper introduces three heuristics that attempt to balance the size of clusters being merged. It has successfully removed this size limitation and obtained community structures of a large network consists of more than 5,000,000 nodes. Another algorithm was introduced by Blondel et al. [28] which is divided into two phases that are repeated iteratively. Initially, every node is considered as a cluster itself, so there are $|V|$ clusters. In the first phase, every node is merged with neighboring cluster that forms the largest positive gain. The node stays in its cluster if all possible gains associated with the merging of this node are negative. This merging procedure repeats iteratively until there is no increase in the value of Q. The resulting Q from the first phase is considered as a local maximum. Then, the second phase make use of the results of the first phase to build a community network.

2.5.2 Spectral Algorithms

Spectral algorithms of modularity maximization can be categorized into two types: the first type is based on Laplacian matrix and the other is based on the modularity matrix of a network.

A. Modularity optimization using the eigenvalues and eigenvectors of the modularity matrix.

Modularity (Q) can be expressed as $Q = \frac{1}{4|E|}\sum_{ij}\left[A_{ij} - \frac{k_i k_j}{2|E|}\right] s_i s_j = \frac{1}{4|E|}\ s^T B s$ (2-4) where s is the column vector representing any division of the network into two groups, A_{ij} are the elements of adjacent matrix A, $s_i = -1$ if it belongs to the second group and $s_i = +1$ if node i belongs to the first group [21]. B is the modularity matrix with elements $B_{ij} =$

$A_{ij} - \frac{k_i k_j}{2|E|}$ (2-5). However, the spectral algorithm described above has two drawbacks. First, instead of getting all the clusters directly in a single step, it divides a network into more than two communities by repeated division. Second, only leading eigenvector of the modularity matrix is used, and the other eigenvectors are ignored. Using multiple leading eigenvectors, Newman [29], propose an algorithm to divide a network into a set of clusters C with $|C| \geq 2$ directly. Richardson et al. [30] presents a computationally-efficient method called spectral tri partitioning where they divide the plane of node vectors into three groups in a single partitioning step using the leading pair of eigenvectors of a modularity matrix.

***B*. Modularity optimization using the eigenvalues and eigenvectors of the Laplacian matrix**

Given a set of clusters C and the corresponding "cluster-assignment" matrix $S = (s_c)$, White and Smyth [31] rewrite modularity (Q) as follows: $Q = -Tr(S^T L_Q S)$ (2-6), where the matrix $L_Q = \breve{D} - W$ (2-7) is called the "Q-Laplacian". Finding the clusters assignment matrix S which maximizes Q above is NP-complete. However, by relaxing the discreteness constraints of the elements of S we can find a good approximation. However, because of running k-means partitioning the two algorithms, especially "Algorithm Spectral-1", is not efficient for large networks. Ruan and Zhang [32] proposed the *Kcut* algorithm which is efficient computationally and improves the quality of the identified clusters. At each recursive step, *Kcut* adopts a k-way partition ($k = 2, 3, \dots, l$) to the subnetwork induced by the nodes and edges in each community using "Algorithm Spectral-1" of White and Smyth [31]. Then, it selects the k that achieves the highest Q. Newman [33] shows that with hyper ellipsoid relaxation, the spectral modularity maximization method using the eigenvalues and eigenvectors of the modularity matrix can be formulated as the spectral algorithm that relies on the eigenvalues and eigenvectors of Laplacian matrix. He also shows that there is no difference in term of computational time between the spectral algorithms of modularity maximization, normalized cut graph partitioning and likelihood maximization.

2.5.3 Simulated Annealing Algorithms

Simulated annealing (SA) is a probabilistic procedure for solving unconstrained and bound-constrained optimization problems. This method was adopted for community detection problems in [34],[35],[36] and [37]to maximize modularity (Q). All the algorithms in [34],[35],[36] and [37]start with partitioning of nodes into clusters, even including $|V\ |$ clusters where each node belongs to its own cluster. A community c and a node i is chosen randomly in each iteration. This community could be a currently existing community, or an empty community introduced to increase the number of communities. Then, node i is moved from its original community to this new community c, which would change Q by ΔQ. If ΔQ is greater than zero, this update is accepted, otherwise it is accepted with probability $e^{\beta \Delta Q}$ where β in [34],[35],[36] and [37]represents the inverse of temperature T and β in [20] is the reciprocal of pseudo temperature τ. In addition, in [38], there is one more condition for the move of a node when c is not empty. Shifting node i to c is considered only if there are some edges between node i and the nodes in c.

2.5.4 Mathematical Modeling

Graph clustering can be defined as an optimization problem where there is a specific objective function to optimize and some constraints to satisfy [38]. Agarwal and Kempe [39] propose two novel algorithms for modularity maximization based on a linear programing (LP) relaxation of an integer programing(IP) and vector program (VP) relaxation of a quadratic program (QP). The integer programing formulation as follows:

$$\text{Max } \frac{1}{2|E|}\sum_{ij} B_{ij}(1 - x_{ij}) \qquad (2\text{-}8)$$

s.t

$$x_{ij} \leq x_{ij} + x_{jk} \ \forall\, i, j, k \qquad (2\text{-}9)$$

$$x_{ij} \in \{0, 1\} \forall\, i, j \qquad (2\text{-}10)$$

Where x_{ij} is a binary variable that $x_{ij} = 1$ if i and j belong to different communities and $x_{ij} =$ 0 if they are in the same community, and B= $a_{i,j} - \frac{d_i d_j}{2m}$ is the modularity matrix. Constraint (2-9) requires that i and k are in the same community if and only if $i, j,$ and k are in the same community. Agarwal and Kempe employ a linear programing, which can be solved in polynomial time, by replacing constrain (2-10) – that $x_{ij} \in \{0, 1\}$ – with $x_{ij} \in [0,1]$. However, if the solution is fractional, rounding of the LP is needed. The second algorithm is a vector program (VP) relaxation of a quadratic program (QP) by dividing a network into two communities which is similar to the approach proposed by Newman [29]. The quadratic program (QP) can be written as:

$$\text{Max } \frac{1}{4|E|}\sum_{ij} B_{ij}(1 + s_i s_j) \qquad (2\text{-}11)$$

s.t

$$s_i^2 = 1 \ \forall i \qquad (2\text{-}12),$$

where constraint $s_i^2 = 1$ ensures that $s_i = \pm 1$ which implies that node i belongs either to the first or the second community. Since quadratic programming is NP-complete, it has been relaxed to a vector program by standard technique of relaxing the QP. According to [40] modularity maximization was formulated as a mixed integer quadratic programming (MIQP) as follows :

$$\max Q = \sum_m \left[\frac{L_m}{L} - \left[\frac{D_m}{2L}\right]^2\right] \qquad (2\text{-}13)$$

s.t

$$L_m = \sum_l X_{lm} \quad \forall\, m \qquad (2\text{-}14)$$

$$D_m = \sum_n d_n Y_{nm} \quad \forall\, m \qquad (2\text{-}15)$$

$$E_m \le E_{m-1} \quad \forall\, m = 2, \ldots, M \qquad (2\text{-}16)$$

$$\sum_l X_{lm} \ge \alpha E_m \quad \forall\, m \qquad (2\text{-}17)$$

$$\sum_l X_{lm} \le \beta E_m \quad \forall\, m \qquad (2\text{-}18)$$

$$L_m - L_k \le \varepsilon + \beta(1 - E_k) \quad \forall\, m, k > m \qquad (2\text{-}19)$$

$$L_m - L_k \le \varepsilon + \beta(1 - E_k) \quad \forall\, m, k > m \qquad (2\text{-}20)$$

$$X_{lm} \le Y_{nm} \quad \forall\, l = \{n, e\}, m \in ML_l \qquad (2\text{-}21)$$

$$X_{lm} \le Y_{em} \quad \forall\, l = \{n, e\}, m \in ML_l \qquad (2\text{-}22)$$

$$X_{lm} = 0 \quad \forall\, l, m\; not \in ML_l \qquad (2\text{-}23)$$

$$\sum_{m \in AM_n} Y_{nm} = 1 \quad \forall\, n \in S \qquad (2\text{-}24)$$

$$\sum_m Y_{nm} = 1 \quad \forall\, n \in S \qquad (2\text{-}25)$$

$$Y_{nm} \le \sum_{e \in (B_n \cap Av_{m-1})} Y_{em-1} \qquad \forall n \in S, n \ge 3, m = 3\ldots, |AM_n| \qquad (2\text{-}26)$$

$$E_m, X_{lm}, Y_{nm} \in \{0,1\}\, \forall\, m, l, n \qquad (2\text{-}27)$$

$$L_m, D_m > 0 \quad \forall\, m \qquad (2\text{-}28),$$

where L_m is number of links among nodes within module m , L is the number of links and D_m is degree of module m. The proposed model results in an efficient computational performance, and this can be attributed to the special symmetry-breaking constraints. These constraints eliminate the equivalent solution resulting from renumbering the modules by allowing each node to be allocated to one of a particular set of modules. Moreover, modularity maximization can be reformulated as a clique partitioning problem [41]. D. Aloise introduces two types of formulations the row generation formulation and the column generation formulations. We will present the two formulations briefly and for more details you may refer to [41]

Row generation formulation

$$max \ \sum_{i,j \in V} w_{ij} x_{ij} - C \qquad (2\text{-}29)$$

s.t

$$x_{ij} + x_{jk} - x_{ik} \leq 1 \quad \forall\, 1 \leq i < j < k \leq n \qquad (2\text{-}30)$$

$$x_{ij} - x_{jk} + x_{ik} \leq 1 \quad \forall\, 1 \leq i < j < k \leq n \qquad (2\text{-}31)$$

$$-x_{ij} + x_{jk} + x_{ik} \leq 1 \quad \forall\, 1 \leq i < j < k \leq n \qquad (2\text{-}32)$$

$$x_{ij} \ \in \{0,1\} \quad \forall\, 1 \leq i < j \leq n \qquad (2\text{-}33)$$

Column generation formulation

$$\text{Max} \sum_{t \in T} c_t z_t - C \qquad (2\text{-}34)$$

St.

$$\sum_{t \in T} a_{it} z_t = 1 \ \ \forall\, i = 1 , \ldots , n \qquad (2\text{-}35)$$

$$z_t \ \in \{0,1\} \quad \forall\, t \in T \qquad (2\text{-}36)$$

In addition the author introduces a column generation reformulation of the (MIQP) [41] which appears to be the best choice because of its lower computational time and its ability to solve optimally network with 104 nodes.

2.5.5 Modularity Limits

Notwithstanding its *NP*-hardness, modularity maximization solution has practical problems such as *resolution limit* and *degeneracy*. Li and Zhang [42] propose an alternative clustering measure called modularity density which overcome the resolution limit of modularity maximization. The modularity density of a partition is defined as the sum of all average modularity degrees of $G_i\ for\ i = 1, \ldots, m$. Let D denote the modularity density called(the D value in this paper) of a partition of a network G into communities $G1 , \ldots, Gm$. Then, in contrast to Q, D can be calculated as $D = \sum_{i=1}^{m} d(G_i) = \sum_{i=1}^{m} \frac{L(V_i,V_i)-L(V_i,\overline{V_i})}{|V_i|}$ (2-37), where $L(V_i, V_i)$ is the sum of degrees of the nodes inside community V_i, $L(V_i, \overline{V_i})$ is the number of connections between nodes within the community $V_i$and the nodes outside the V_i community, $|V_i|$ is the number of nodes in community V_i. According to the clustering in figure 2, modularity density D can be calculated as $D = \left(\frac{16-8}{5}\right) + \left(\frac{6-8}{3}\right) = 0.933$. Thus, modularity density D metric has higher value compared to Q. Using this definition of modularity density function Li and Zhang [42] propose a nonlinear integer programming model for optimizing the D value as following:

$$max\ f = \sum_{l=1}^{k} \frac{\sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}x_{il}x_{jl} - \sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}x_{il}(1-x_{jl})}{\sum_{i=1}^{n} x_{il}} \qquad (2\text{-}38)$$

s.t

$$0 \le \sum_{i=1}^{n} x_{il} \le n \quad l = 1, \ldots, k \qquad (2\text{-}39)$$

$\sum_{l=1}^{k} x_{il} = 1 \; i = 1, \dots, n$ (2-40)

$x_{il} \in \{0,1\} \quad l = 1, \dots, k \quad$ and $\; i = 1, \dots, n$ (2-41)

Where k is the number of communities, a_{ij} is the elements of the adjacency matrix A, and x_{il} is a binary variable where $x_{il} = 1$ denotes that the node vi belongs to the l^{th} community. However, it seems that modularity density maximization (MDM) as a 0-1 NLP formulation is more complicated than modularity maximization (MM) because it is not straight forward to employ the clique partitioning formulation for MDM. In other words, given a clique with *n* nodes, maximizing modularity density does not divide it into two or more parts. This is proved by contradiction in [42]. Costa [43] reformulates the 0-1 NLP model as a mixed integer linear programing (MILP). He introduces four mathematical models, but we will briefly present the first model and you may refer to the paper for other models.

Max $\sum_{c \in C} \alpha_c$ (2-42)

s.t

$2 \leq \sum_{v_i \in V} Y_{i,c} \leq |V| - (|C| - 1) \quad \forall\, c \in C$ (2-43)

$\sum_{c \in C} Y_{i,c} = 1 \quad \forall v_i \in V$ (2-44)

$W_{i,j,c} \leq Y_{i,c} \quad \forall\, c \in C\,, \forall \{v_i, v_j\} \in E$ (2-45)

$W_{i,j,c} \leq Y_{j,c} \quad \forall\, c \in C, \forall \{v_i, v_j\} \in E$ (2-46)

$4 \sum_{\{v_i,v_j\} \in E} W_{i,j,c} - \sum_{v_i \in V} Y_{i,c} k_i > \sum_{v_i \in V} S_{i,c} \quad \forall\, c \in C$ (2-47)

$L_\alpha Y_{i,c} \leq S_{i,c} \leq U_\alpha Y_{i,c} \quad \forall\, c \in C\,,\ \forall v_i \in V$ (2-48)

$\alpha_c - U_\alpha (1 - Y_{i,c}) \leq S_{i,c} \leq \alpha_c - L_\alpha (1 - Y_{i,c}) \;\; \forall\, c \in C\,,\ \forall v_i \in V$ (2-49)

$W_{i,j,c} \in R \quad \forall \{v_i, v_j\} \in E, \forall\, c \in C$ (2-50)

$S_{i,c} \in R \quad \forall v_i \in V,\ \forall\, c \in C$ (2-51)

$$\alpha_c \in [L_\alpha, U_\alpha] \quad \forall\, c \in C \tag{2-52}$$

$$Y_{i,c} \in \{0,1\} \quad \forall\, c \in C,\ \forall v_i \in V \tag{2-53}$$

This model has an issue of solving the auxiliary binary Non-Linear Programs (NLPs) which is required as input for MILP. This issue will affect the computational time significantly when the network size becomes larger. Costa reformulates the auxiliary problem as a MILP to overcome this , so that MDM becomes a MILP [44]. This has been done by employing some reformulation techniques, e.g., linearization of bilinear terms [45], [46], [47], expansion of integers in power of two [48], and reformulation of fractional programs [49]. MILP in [43] has some challenges in which a 0–1 NLP auxiliary problem needs to be solved and the number of communities need to be fixed in advance. A variant of a semidefinite programming called 0-1SDP is proposed by [50]. He reformulates the MILP in [43] and show it is equivalent to semidefinite programming. The 0-1 SDP is given as

$$\text{maximize } \mathrm{T}r((2A - D)Z) \tag{2-54}$$

s.t

$$Z\, e_n = n \tag{2-55}$$

$$\mathrm{Tr}\ (Z) = t \tag{2-56}$$

$$Z^2 = Z \tag{2-57}$$

$$Z \in N_n \tag{2-58}$$

One of the advantages of this formulation is that the size of the problem is independent of the number of edges of the graph. In order to obtain an upper bound on the modularity density, they propose to relax 0-1SDP to a semidefinite programming problem with non-negative constraints. The relaxation problem obtained, can be solved in polynomial time, and also does not require the number of communities in contrast to MILP formulations. Moreover, they develop a method based on the combination of spectral heuristics and dynamic programming to construct a feasible solution from the solution obtained by the relaxation problem. In addition to its 0-1 NLP auxiliary problem difficulty, MILP in [45] can only solve up to 40 nodes. Thus, six column generation algorithms was proposed by [51] to find exact solution for MDM. The wonder of this algorithm is that it provides only integer solution, hence no need for further rounding procedure such as branch and price. The column generation formulation of MDM in [51] is derived from MM formulation in [41] is as follows:

$$\text{Max} \sum_{t\in T} c_t z_t \qquad (2\text{-}59)$$

St.

$$\sum_{t\in T} a_{vt} z_t = 1 \ \ \forall\, i\ v \in V \qquad (2\text{-}60)$$

$$z_t \geq 0 \qquad \forall\, t \in T \qquad (2\text{-}61)$$

$$c_t = \frac{4\sum_{u,v\in V:u<v} a_{ut}a_{vt}w_{uv} - \sum_{v\in V} d_v a_{vt}}{\sum_{v\in V} a_{vt}} \qquad (2\text{-}62)$$

Where $T = \{1, \dots, 2^{|v|}\}$ is all the possible clusters, and z_t is a binary variable as in MM but it was relaxed to obtain the dual problem. a_{vt} Variables are binary. If $a_{vt} = 1$, the node $v \in V$ belongs to the cluster t , and when a_{vt} = 0 , the node v does not belong to this cluster. For all u, $v \in V$, the constant $w_{uv} = 1$ if the u and v are adjacent and $w_{uv} = 0\{u, v\}$ otherwise.

Izunaga [52] proposes a branch-and-price frame work for MDM that is formulated as an ILP of a set covering problem. He proposes a column generation of a simple MILP combined with set-packing relaxation and the multiple-cutting-planes to solve the subproblem. The proposed algorithm is able to solve instances with over 100 vertices in a reasonable computational time. The previously mentioned algorithms find exact solution of MDM. However, there are many proposed heuristics that can give a good solution for MDM problem. One of the most famous heuristics is the hierarchical divisive heuristic by A. Costa [53]. The proposed heuristic mainly depends on the idea that firstly employed for modularity maximization by S. Cafieri [54], [55] and extended by A. Costa and P. Hansen [56], of starting by putting all vertices in one community and then recursively split it into two communities by maximizing the modularity density. To obtain the optimal splitting, the author derives four mathematical models (repetitive resolutions of an ILP or MILP) with two different symmetry breaking strategies. The hierarchical divisive heuristic provides a near optimal solution for MDM problem. Another heuristic for MDM is introduced by R. Santiago [57]. The author proposes seven scalable heuristics which are faster than many other heuristics such as BMD- λ in [50]. In addition, these heuristics can find the high objective value partitions for the largest instances up to 105 nodes. Although there are many algorithms to solve modularity density maximization problem (MDM), most of them are unable to handle large scale networks. Recently, Shang presents a new strategy based on pre-partitioning and optimizing an improved modularity density increment ΔD [58]. The method starts by searching for the core nodes and pre-partitions the network according to the node similarity. Secondly, they use the improved modularity density increment ΔD as an objective function to proceed clusters integration. According to [58] Improved modularity density increment ΔD is defined as

$\Delta D = \left[\frac{l_i(u)-l_0(u)-l_0(v)+3l_{uv}}{d_u+d_v}\right] - \left[\frac{l_i(u)-l_0(u)}{d_u} + \frac{l_i(v)-l_0(v)}{d_v}\right]$ (2-63) where u and v are the merged communities, $l_i(u)$ is the number of connections within the community u, $l_0(u)$ is the number of connections outside the community u, l_{uv} is the number of connections between the community u and v, d_u is the sum of node degree of the nodes within the community u.

CHAPTER 3

MODIFIED MODULARITY DENSITY MAXIMIZATION APPROACH

Modularity density maximization was proposed to overcome the issue of resolution and degeneracy limits appearing in modularity maximization. Much research has been done in this area. However, there are some gaps in operations research literature. Modularity Density Maximization (MDM) aims to reduce the out links between clusters. So, the less out connection is the better .The out connection can also be seen as a kind of distance. Thus, in the Modified Modularity Density Maximization (MMDM) we minimize the deep connection instead of the out links which works as out connection measure. For example, we don’t want someone in a group to be deeply connected (distant connection) to someone else in another group because we want them in separate clusters. In this chapter we modify modularity density maximization in this sense and propose a mixed integer linear programing (MILP) of MMDM then we discuss the optimal results obtained by solving the mathematical model.

3.1 Minimizing Longest Shortest Path Model (ML)

In this section, we propose ML model which minimizes the longest shortest path between each pair of nodes in different clusters. In community detection, clusters need to be separated from each other’s. This mathematical model will create kind of separation between clusters which will be used in the next section as a separation measure between clusters instead of number of out links between clusters. ML model can be formulated as following:

$$min \ \sum_c L_c \qquad (3\text{-}1)$$

s.t

$$L_c \geq \ d_{ij} x_{ic} x_{jc} \ \forall i, j, c, k \ \ , i \neq j, c \neq k \qquad (3\text{-}2)$$

$$\sum_c x_{ic} = 1 \qquad \forall i \qquad (3\text{-}3)$$

$$\sum_i x_{ic} \geq 2 \qquad \forall c \qquad (3\text{-}4)$$

$$x_{ic} \ \in \{0,1\} \qquad (3\text{-}5)$$

L_c is the longest shortest path between each pair of nodes in different clusters. Hence minimizing this variable will create kind of separation between clusters. d_{ij} is the distance matrix between the nodes of the whole graph and x_{ic} is a binary variable that will be equal to 1 if node *i* belongs to cluster *c* and 0 otherwise. Constraint (3-2) will check if node *i* and *j* are in different clusters then it will take the value of the distance between them otherwise it will be equal to 0. Thus, the constraint will keep updating the value of L_c and with objective function it will pick up the highest value. Constraint (3-3) is forcing each node to belong to only one cluster to avoid overlapping. Constraint (3-4) is assigns at least 2 nodes for each cluster. Nonlinear constraints (3-3) can be linearized as:

$$L_c \geq \ d_{ij}\left(x_{ic} + x_{jk} - 1\right) \forall i, j, c, k \ \ , i \neq j, c \neq k \qquad (3\text{-}5).$$

3.2 Modified Modularity Density Maximization (MMDM)

As it is known that modularity density maximization (MDM) will try to reduce the out links between clusters. So, the less out connection is the better. The out connection is also can be seen as a kind of distance. Thus, in our model we thought of having the paths instead of the out links which will work as out connection measure. The modified modularity density function is defined as $MD = \sum_{c \in C}(\frac{2m_c - L_c}{n_c})$ (3-6) where n_c is the number of nodes inside cluster c , m_c is the number of inside connection of each cluster, and L_c longest shortest path between each pair of nodes of c and different clusters. This expression can be written as $MD = \sum_{c \in C}\left(\frac{\sum_i \sum_j a_{ij} y_{ic} y_{jc} - L_c}{\sum_i y_{ic}}\right)$ (3-7), where y_{ic} is a binary variable equals to 1 if node i belongs to cluster c and 0 otherwise and a_{ij} is the adjacency matrix for graph G. A nonlinear formulation of maximizing the objective function in (3-7) is as following:

$$\max \sum_{c \in C}\left(\frac{\sum_i \sum_j a_{ij} y_{ic} y_{jc} - L_c}{\sum_i y_{ic}}\right) \qquad (3\text{-}8)$$

s.t

$$2 \leq \sum_{i \in V} Y_{i,c} \leq |V| - (|C| - 1) \qquad \forall\, c\ \in C \qquad (3\text{-}9)$$

$$\sum_{c \in C} Y_{i,c} = 1 \qquad \forall i \in V \qquad (3\text{-}10)$$

$$L_c \geq\ d_{ij}(y_{ic} + y_{jc} - 1)\ \forall i, j, c, k\ \ , i \neq j, c \neq k \qquad (3\text{-}11)$$

$$y_{ic}\ \in \{0,1\} \qquad (3\text{-}12)$$

Constraint (3-9) decides on the upper and lower size of each cluster. Constraint (3-10) is forcing each node to belong to only one cluster and preventing overlapping. Constraint (3-11) compute the largest distance between nodes of different clusters.

3.3 Mixed Integer Linear Programing Formulation (MILP) of MMDM

MMDM (0-1NLP) formulation can be reformulated as MILP. Nonlinearity from the product of y binary variables in (3-8) can be linearized by introducing a new W_{ijc} variables using Fortet inequalities [61]. Thus, the term $y_{ic}y_{jc}$ is replaced by the variables W_{ijc} adding two sets of constraints

$$W_{i,j,c} \leq y_{i,c} \qquad \forall\, c \ \in C \qquad (3\text{-}13)$$

$$W_{i,j,c} \leq y_{j,c} \qquad \forall\, c \ \in C \qquad (3\text{-}14)$$

The second nonlinearity is due to the fraction, and this can be reformulated by introducing new variable called α_c which will be maximized. Thus, we can write (3-8) as

$\frac{\sum_i \sum_j a_{ij} W_{ijc} - L_c}{\sum_i y_{ic}} \geq \alpha_c \quad \forall\, c \ \in C$ which can be simplified as

$$\textstyle\sum_i \sum_j a_{ij} W_{ijc} - L_c \geq \sum_i y_{ic}\, \alpha_c \qquad \forall\, c \ \in C \qquad (3\text{-}15).$$

We still need to linearize the product $y_{ic}\alpha_c$ which can be done by introducing a new variable S_{ic} which will replace each $y_{ic}\alpha_c$ term and the McCormik inequalities will be added to the model [63].

$$L_\alpha y_{i,c} \leq S_{i,c} \qquad (3\text{-}16)$$

$$S_{i,c} \leq U_\alpha y_{i,c} \qquad (3\text{-}17)$$

$$\alpha_c - U_\alpha \left(1 - y_{i,c}\right) \leq S_{i,c} \qquad (3\text{-}18)$$

$$S_{i,c} \leq \alpha_c - L_\alpha(1 - y_{i,c}) \qquad (3\text{-}19)$$

We still need upper U_α and lower L_α bound on the variable α_c but this will be discussed later. The final MILP formulation of MMDM can be written as:

$$max \sum_{c \in C} \alpha_c \qquad (3\text{-}20)$$

s.t

$$2 \leq \sum_{i \in V} Y_{i,c} \leq |V| - (|C| - 1) \qquad \forall\, c \in C \qquad (3\text{-}21)$$

$$\sum_{c \in C} Y_{i,c} = 1 \qquad \forall\, i \in V \qquad (3\text{-}22)$$

$$L_c \geq d_{ij}(y_{ic} + y_{jc} - 1) \qquad \forall i, j, c, k \;, i \neq j, c \neq k \qquad (3\text{-}23)$$

$$W_{i,j,c} \leq Y_{i,c} \qquad \forall\, c \in C, \forall\, i, j \in V \,, a_{ij} = 1 \qquad (3\text{-}24)$$

$$W_{i,j,c} \leq Y_{j,c} \qquad \forall\, c \in C, \forall\, i, j \in V \,, a_{ij} = 1 \qquad (3\text{-}25)$$

$$\sum_{i \in V} \sum_{j \in V} a_{ij} W_{ijc} - L_c \geq \sum_{i \in V} S_{i,c} \qquad \forall\, c \in C \qquad (3\text{-}26)$$

$$L_\alpha Y_{i,c} \leq S_{i,c} \leq U_\alpha Y_{i,c} \qquad \forall\, c \in C \,, \; \forall\, i \in V \qquad (3\text{-}27)$$

$$\alpha_c - U_\alpha(1 - Y_{i,c}) \leq S_{i,c} \leq \alpha_c - L_\alpha(1 - Y_{i,c}) \;\; \forall\, c \in C \,, \; \forall\, i \in V \qquad (3\text{-}28)$$

$$W_{i,j,c} \in R \qquad \forall\, c \in C, \forall\, i, j \in V \,, a_{ij} = 1 \qquad (3\text{-}29)$$

$$S_{i,c} \in R \qquad \forall\, i \in V, \; \forall\, c \in C \qquad (3\text{-}30)$$

$$\alpha_c \in [L_\alpha, U_\alpha] \quad \forall\, c \in C \qquad (3\text{-}31)$$

$$Y_{i,c} \in \{0,1\} \qquad \forall\, c \in C, \; \forall\, i \in V \qquad (3\text{-}32).$$

3.4 Upper $\boldsymbol{U_\alpha}$ and Lower $\boldsymbol{L_\alpha}$ Bounds.

In the proposed MILP our objective variables α_c are constrained by upper U_α and lower L_α bounds. So, having a good upper and lower bound will affect the performance of the model. As it is stated in Costa[43], it is difficult to derive a tight upper and lower bound theoretically. A good upper bound can be found by maximizing the positive part of the objective function max $(\frac{\sum_i \sum_j a_{ij} y_{ic} y_{jc}}{\sum_i y_{ic}})$. Similarly, a good lower bound can be derived by minimizing the negative part of the objective function $min\ \sum_c L_c$.

<u>Upper Bound $\boldsymbol{U_\alpha}$</u>

$$max \sum_{c\in C} X_c \qquad (3\text{-}33)$$

s.t

$$X_c = \frac{\sum_i \sum_j a_{ij} y_{ic} y_{jc}}{\sum_i y_{ic}} \qquad \forall\ c\ \in C \qquad (3\text{-}34)$$

$$2 \le \sum_{i\in V} Y_{i,c} \le |V| - (|C| - 1) \qquad \forall\ c\ \in C \qquad (3\text{-}35)$$

$$\sum_{c\in C} Y_{i,c} = 1 \qquad \forall i \in V \qquad (3\text{-}36)$$

$$y_{ic}\ \in \{0,1\} \qquad \forall\ c\ \in C,\ \forall\ i \in V \qquad (3\text{-}37)$$

The upper bound will be the $U_\alpha = \max\{X_c\}$ (3-38)

<u>Lowe Bound L_α</u>

$$min\ \sum_c L_c \qquad (3\text{-}39)$$

s.t

$$L_c \geq\ d_{ij}x_{ic}x_{jc}\ \forall i,j,c,k\ \ , i \neq j, c \neq k \qquad (3\text{-}40)$$

$$\sum_c x_{ic} = 1 \qquad \forall i \qquad (3\text{-}41)$$

$$\sum_i x_{ic} \geq 2 \qquad \forall c \qquad (3\text{-}42)$$

$$x_{ic}\ \in \{0,1\} \qquad (3\text{-}43)$$

The lower bound will be $L_\alpha = \frac{-\max\{L_c\}}{2}$ (3-44) We divide by 2 because the minimum number of nodes in each cluster will be 2.

3.5 MMDM Results and Discussion

In this section we present the results obtained by solving the Mixed Integer Linear Programing of the Modified Modularity Density Maximization (MMDM) explained in the previous chapter. The experiment was performed on a PC with 8 gigabytes RAM and core i 5 processor. The mathematical models used in the paper are solved using CPLEX solver for both MILP model and ML model [62]. For the upper bound model, BONMIN solver was used. Both solvers are used in GAMS software [63], while all visualization are performed using R and igraph library[64]. The data sets was collected from Pajek website (http://vlado.fmf.uni-lj.si/pub/networks/data/). In order to obtain connected undirected and unweighted graphs some edges were removed, like in Korea2 data set, and the orientation of arcs were ignored. Strike data set refers to the employees in a wood-processing facility starting strike because of the new changes to their compensation package. The vertices representing the employees and the edges representing the frequently communication between employees to negotiate the administration about their statement. Karate is a well-known network which represent the friendship relationship between the members of Zachary karate club. In Korea 2 data set the vertices represent women and the edges are the discussion between them about the family planning. Mexico represents the Mexican political elites as the nodes and friendship, kinship or business ties as the connections between nodes. Chesapeake represent road-Chesapeake's link structure. Table 3-1 contains summary of network data sets providing number of nodes, number of edges. It also shows the optimal solution of Modified Modularity Density Maximization (in term of modified modularity density value MD and number of clusters $|C|$), the upper bound (U_α) and lower bound (L_α) used in MILP of MMDM and the computational time in seconds (t) to find the optimal solution. In addition, we represent the optimal solution of Modularity Density Maximization (MDM) (in term of modularity density value D and number of clusters $|C|$), since they are strongly related.

Table 3-1 The optimal solution of MMDM and MDM obtained by MILP in term of number of clusters, MD and D

			MMDM					MDM	
Data Set	V	E	U_α	L_α	$\|C\|$	MD	t[s]	$\|C\|$	D
Strike	24	38	3.11	-2	3	6.458	24.53	4	8.861
Karate	34	78	4.13	-4	2	7.411	86399.01	3	7.845
Korea 2	35	84	4.85	-3	5	9.448	86000.08	5	11.143
Mexico	35	117	5.33	-1.5	3	13.078	3597.00	3	8.718
Chesapeake	39	170	6.67	-1	3	15.100	86328.5	3	7.470

For illustration purposes and graph configuration figures 3-1 and 3-2 show the optimal solution of MMDM and MDM for Karate network where each cluster is identified by different color. As seen, the optimum value of MMDM occurs when Karate network is clustered into 2 groups. However, MDM optimum value was with 3 clusters. Similar graph visualization was made for the other networks in appendix A.

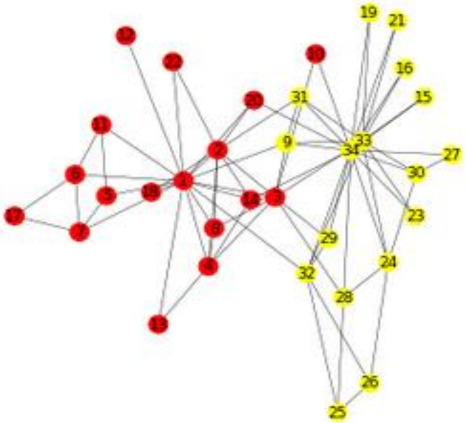


Figure 3-2 Optimal solution obtained by MILP of MMDM for Karate data set

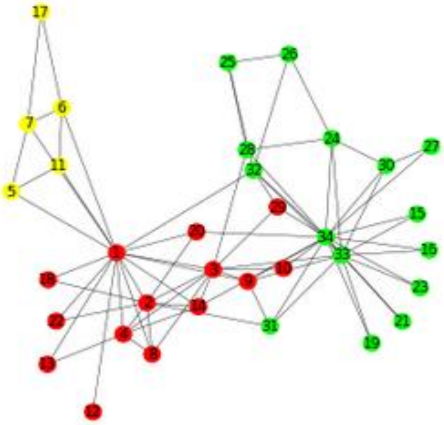


Figure 3-1 Optimal solution obtained by MILP of MDM for Karate data set

Since we have different objective function from modularity density maximization D it is not correct to compare the two objective functions as a numerical value. Thus, the comparison between the two clustering methods is done through cluster validation approach which will be clearly discussed in cluster validation section.

"Difficulty", as computational complexity of MILP, is affected by the number of constraints and number of variables. Table 3-2 compares the size of MMDM and MDM models in term of number of constraints and number of variables as equations where v is number of nodes, k is number of clusters and a_{ij} is the elements of adjacency matrix for graph G. As seen in table 3-2 MDM model has a smaller number of constraints and variables which reflects better computational time.

Table 3-2 MILP Computational Complexity

MMDM	Number of constraints	$\frac{kv(v-1)(k-1)}{4} + a_{ij}k(v-1)v + 4kv + v + 5k$
	Number of variables	$\frac{a_{ij}k(v-1)v}{2} + 2k(v+1)$
MDM	Number of constraints	$a_{ij}k(v-1)v + 4kv + v + 5k$
	Number of variables	$\frac{a_{ij}k(v-1)v}{2} + k(v+1)$

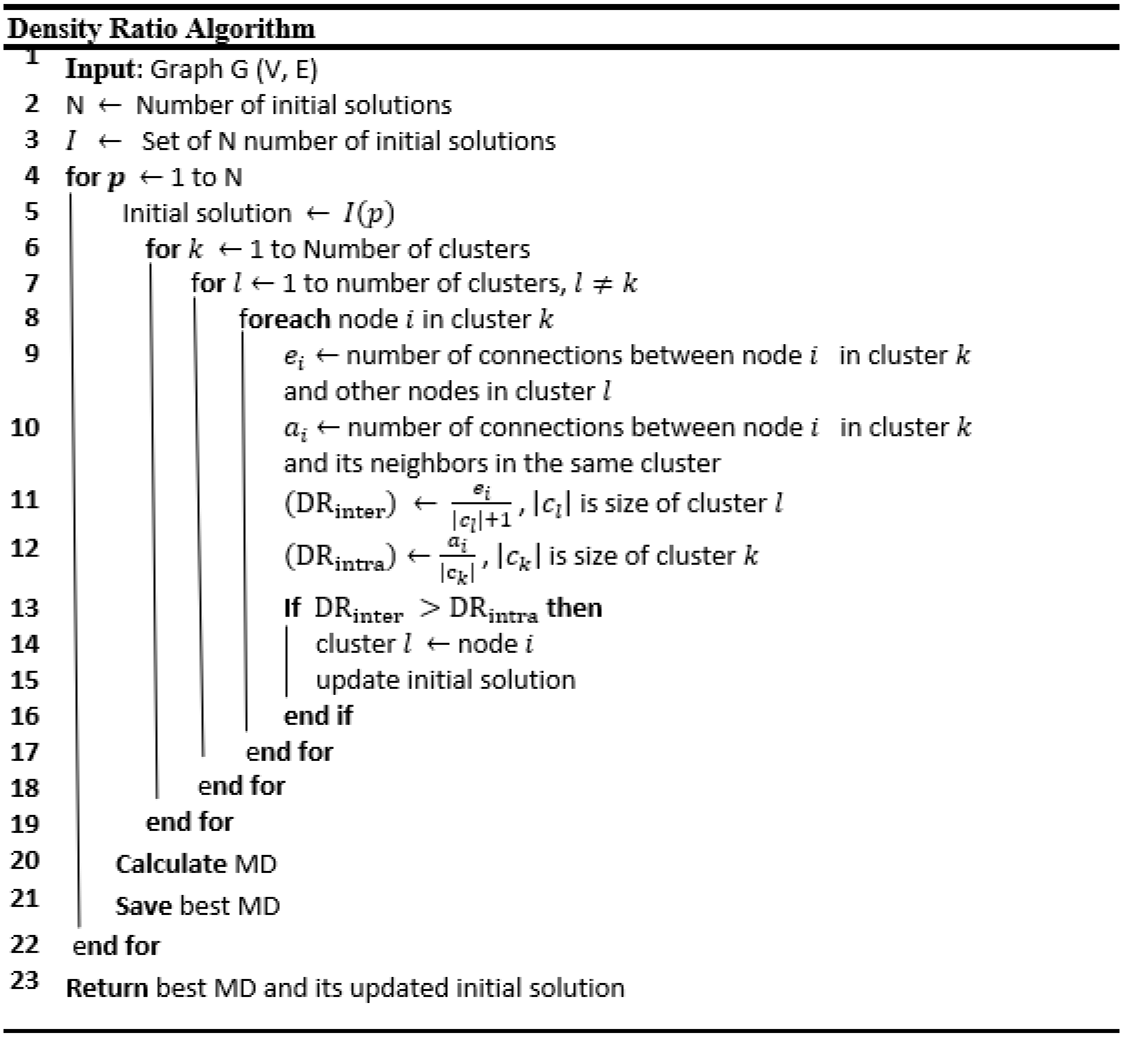
CHAPTER 4

DENSITY RATIO HEURISTIC ALGORITHM

Modularity density maximization was proposed as a binary nonlinear programing (0-1 NLP) which is very difficult for many solvers. Recently, Modularity density maximization (MDM) was formulated as mixed integer linear programing (MILP) [43]. However, the proposed model can solve optimally only up to 40 instances. Thus, many heuristics has been developed to solve larger data sets. Santiago [51] proposed six exact algorithms for MDM using column generation methods. The algorithms provide only integer solution thus, branch and price are no longer needed for further procedures. The algorithms were able to provide solutions up to 105 nodes only. Another exact heuristic proposed a branch-and-price framework to solve the ILP of MDM [52]. The heuristic is deterministic and can solve cases over 100 nodes. In addition to the exact algorithms, a hierarchical divisive heuristic that works by splitting recursively a community into two new communities by maximizing the modularity density, was introduced by Costa [53]. Recently, Santiago presents seven saleable heuristics that can solve up to hundred thousand instances such as “Stanford Large Network Dataset Collection” [57]. In this Chapter we are going to propose a new heuristic algorithm named **Density Ratio Heuristic** that can solve larger instances for Modified Modularity Density Maximization.

4.1 Density Ratio Heuristic (DRH)

Given a graph G (V, E) where V is the set of nodes and E is the set of edges. The algorithm starts by finding a set of initial solutions I where every initial solution has different number of clusters. K-means algorithm was a very good choose to give initial for our case since it needs few parameters to set such as distance matrix between nodes in the graph and number of clusters. Then for each node i in cluster k, where k is from 1 to number of clusters, we compare two density ratios, Inter Density Ratio and Intra Density Ratio. Inter Density Ratio $(DR_{inter}) = \frac{e_i}{|c_l|+1}$ where e_i is number of connections between node i in cluster k and other nodes in other cluster l and $|c_l|$ the number of nodes in cluster *l*. Intra Density Ratio $(DR_{intra}) = \frac{a_i}{|c_k|}$ where a_i is number of connections between node i in cluster k and its neighbor nodes in the same cluster and $|c_k|$ the number of nodes in cluster k. If Inter Density Ratio (DR_{inter}) is greater than Intra Density Ratio (DR_{intra}), we move the node i from its current cluster *k* to the new cluster *l* and update the initial solution I(P). We repeat this iteration for all nodes in each cluster k and calculate the objective function value (MD) of MMDM. Then we repeat for all initial solutions and return best MD value along with its updated initial solution. The algorithm was also applied to evaluate the objective function value D of MDM. Figure 4-1 shows the heuristic algorithm pseudo code as applied to evaluate the objective function MD of MMDM. The pseudo code of the heuristic will be the same for MDM with evaluating D instead of MD.



```
Density Ratio Algorithm
Input: Graph G (V, E)
N ← Number of initial solutions
$I$ ← Set of N number of initial solutions
for $p$ ← 1 to N
    Initial solution ← $I(p)$
    for $k$ ← 1 to Number of clusters
        for $l$ ← 1 to number of clusters, $l \neq k$
            foreach node $i$ in cluster $k$
                $e_i$ ← number of connections between node $i$ in cluster $k$ and other nodes in cluster $l$
                $a_i$ ← number of connections between node $i$ in cluster $k$ and its neighbors in the same cluster
                $(\mathrm{DR}_{\mathbf{inter}}) \leftarrow \frac{e_i}{|c_l|+1}$, $|c_l|$ is size of cluster $l$
                $(\mathrm{DR}_{\mathbf{intra}}) \leftarrow \frac{a_i}{|c_k|}$, $|c_k|$ is size of cluster $k$
                If $\mathrm{DR}_{\mathbf{inter}} > \mathrm{DR}_{\mathbf{intra}}$ then
                    cluster $l$ ← node $i$
                    update initial solution
                end if
            end for
        end for
    end for
    Calculate MD
    Save best MD
end for
Return best MD and its updated initial solution
```

Figure 4-1 Density Ratio Heuristic pseudo code

4.2 Density Ratio Heuristic Results and Discussion

In this section we present the results obtained by solving the Density Ratio heuristic explained in the previous chapter. The code was run on google colaboratory https://colab.research.google.com/notebooks/intro.ipynb#scrollTo=bJyydlZ5lZe3 with given 12 gigabyte RAM. The heuristic was solved by python using Networkx package, and Matplot library was used for graph visualization[65]. We applied our heuristic on 3 different data sets with variant size. Football data set was collected from M. Girvan and M. E. J. Newman paper [11]. Email data set and Facebook data set was collected from "Stanford Large Network Dataset Collection"[66]. Football data contains the network of American football games between Division IA colleges during regular season Fall 2000. Two edges were erroneously duplicated in this data set and have been removed. Facebook data set consists of circles (or friends lists) from Facebook. The data was collected from survey participants using a Facebook app. Table 4-1 contains networks data set details such as number of nodes, number of edges. It also shows the value of MD obtained by our heuristic as we run the heuristic with changing number of clusters starting from 2 clusters up to 20 clusters. We also applied our heuristic to evaluate D as objective function instead of MD. As seen the best values of MD and D is highlighted in bold. Big O notation is used in Computer Science to describe the performance or complexity of an algorithm. Big O specifically describes the worst-case scenario and can be used to describe the execution time required or the space used (e.g. in memory or on disk) by an algorithm. In worst case scenario DR heuristic will have number of clusters k equals number of nodes N. So, the complexity of our proposed heuristic is given by $O(N^2)$ where N is the total number of nodes in the given graph G.

Table 4-1 The obtained results of Density Ratio heuristic as applied to MMDM and MDM

Data set	**Football**				**Email**				**Facebook**			
C	N	E	MD	D	N	E	MD	D	N	E	MD	D
2	115	613	19.065	17.083	1133	5451	13.495	10.997	4039	88234	89.797	88.655
3			26.667	22.038			15.694	8.989			127.167	125.635
4			33.851	26.397			23.591	14.176			177.442	176.688
5			40.307	28.913			24.420	10.225			197.397	196.682
6			47.235	33.076			32.059	16.843			195.349	180.493
7			52.362	33.468			36.481	17.325			231.887	196.642
8			57.323	34.876			41.231	**19.395**			262.218	245.297
9			65.140	40.130			40.819	11.348			261.021	245.240
10			72.186	44.340			47.492	15.573			328.723	310.623
11			**76.447**	**44.388**			50.710	12.899			327.142	262.984
12			73.290	30.781			51.303	5.217			409.405	345.366
13			72.625	21.114			54.200	8.774			474.864	389.984
14			71.950	11.406			59.406	11.364			442.211	344.403
15			74.400	9.228			63.624	10.849			591.872	**495.944**
16			73.003	-2.209			62.868	2.505			542.103	443.452
17			74.890	-6.026			73.802	10.139			**614.246**	332.325
18			70.223	-6.456			72.947	1.984			498.416	327.782
19			69.333	-6.994			72.329	5.351			567.228	322.354
20			66.324	-7.044			**76.009**	-0.348			524.662	261.470

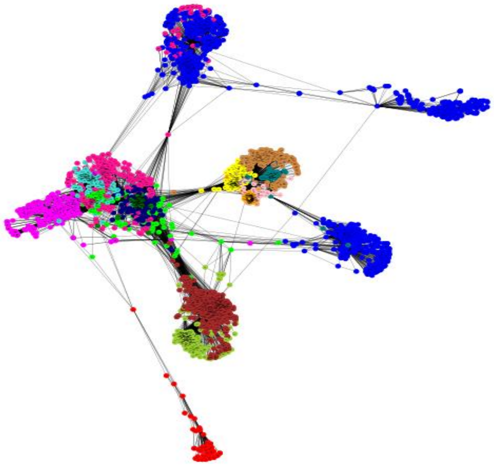


Figure 4-3 Best value obtained by applying Density Ratio heuristic on MMDM for Facebook data set

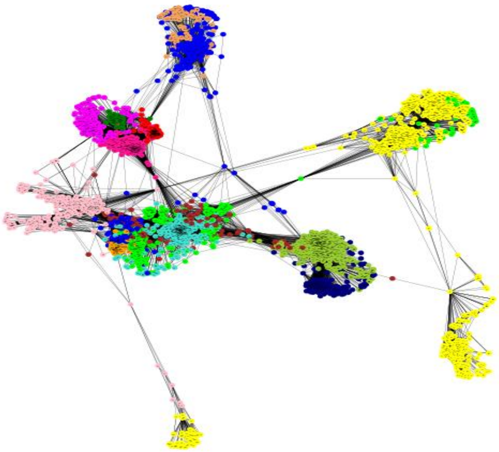


Figure 4-2 Best value obtained by applying Density Ratio heuristic on MDM for Facebook data set

For illustration purposes and graph configuration figure 4-2 and 4-3 show the best solution given by Density ratio heuristic of MMDM and MDM for Facebook network where each cluster is identified by different color. As seen, the best value of MMDM occurs when Facebook network is clustered into 17 groups. However, MDM best value was with 15 clusters. Similar graph visualization was made for the other networks in appendix A.

Since different community detection algorithms will exhibit different results that is affected by the features of the data sets and their predefined groups. It is very challenging task to judge on the performance community detection algorithm. Evaluating the results of the proposed community detection algorithm will be done by cluster validation approach explained in the next chapter.

CHAPTER 5

CLUSTER VALIDATION

In data mining, different community detection algorithms will exhibit different results that is affected by the features of the data sets and their predefined groups. So, it is very challenging task to judge on how community detection algorithm is good. Evaluating the results of community detection algorithms and goodness judgment is called cluster validation[67]. There are three types of cluster validation named as internal validation, external validation, and relative validation[68]. The internal validation cares about the internal information of the clustering process regardless of any external clustering information. It can be also used for determining the right number of clusters. The external validation is comparing the clustering of the proposed community detection algorithm with the real partitioning of the datasets. The relative validation is about evaluating clustering structure by changing different parameter values such as varying the number of clusters c for the same community detection algorithm. Since external validation requires a ground truth which is not provided in most data sets and relative cluster validation is not commonly used in the literature, we are going to use internal cluster validation to evaluate our proposed model and heuristic. Internal cluster validation is much more realistic and efficient in many real-world scenarios as it does not refer to any assumed references from outside which is not always feasible to obtain. Particularly, with the huge increase of the data size, one can hardly claim that a complete knowledge of the ground truth is available or always valid.

5.1 Internal Cluster Validation

Internal validation measure always refers to separation and compactness of the detected communities. Compactness of a cluster is a measure of how the instances inside the cluster are close to each other [69]. Many validation indices consider the distance measures, such as the variance between instances, as a closeness measure. Thus, lower variation between the instances belongs to the same cluster represents good compactness. However, separation of a clusters points how different clusters are separated from each other. In many validation indices they reflect this separation by conducting the distance between cluster's centroids or the minimum distance between each pairs of instances belong to different clusters. In this paper we will use Dunn index, which is based on diameter and distances, and Silhouette index which depends on node's neighborhood. Generally, most indices of cluster internal validation represented as $\frac{\alpha \text{ x } separation}{\beta \text{ x } compactness}$ where α and β are weights.

5.1.1 Dunn Index

Dunn index is one the most common used cluster internal validation indices. It is defined as the ratio between the minimum distance between clusters(min.separation) to the maximum diameter (intra-compactness) as following $\boldsymbol{D = \frac{min.separation}{max.diameter}}$**.** If the dataset has well-separated and compact clusters, the distance between the clusters is expected to be large and the diameter of the clusters is expected to be small [70]. Thus, based on the Dunn's index definition, large values of the index represents well-separated and compact clusters. On the other hand, bad separated and less compact clusters will give low index value close to 0. In table 1, the obtained results from MMDM is compared with the results provided by MDM model from the literature in terms of Dunn Index. Table 5-1 shows that

MMDM method has higher Dunn index value than MDM for Karate network as well as Korea network. This indicates that our MMDM method detected communities that are more compact and well separated than what MDM detected. For Mexico and Chesapeake data sets MMDM and MDM give the same value of Dunn index. For Strike data sets, MDM gives higher Dunn index value

Table 5-1 Comparison between MMDM and MDM in term of Dunn index

Data sets	MMDM		MDM	
	$\|C\|$	**Dunn index**	$\|C\|$	**Dunn index**
Strike	3	0.167	4	0.333
Karate	2	**0.250**	3	0.091
Mexico	3	0.250	3	0.250
Korea2	5	**0.250**	5	0.167
Chesapeake	3	0.333	3	0.333

5.1.2 Silhouette index

Silhouette index is also considered as one of the most famous internal validation indices, that estimates the average distance between clusters. Silhouette width of the instance *i is given as* $S_i = \frac{b_i - a_i}{max(b_i, a_i)}$ where b_i is the inter-dissimilarity and a_i is the intra-dissimilarity of the instance *i* [71]. *T*he value of Silhouette index is between -1 and 1. The larger S_i (value closer to 1) means the instance is very well clustered and the negative value of S_i indicates wrong cluster for that instance. In table 5-2 we compare the results of our model (MMDM) and the results of the MDM model from the literature in terms of Silhouette index. Table 2 shows that MMDM method has higher Silhouette index value than MDM for Karate, Mexico and Korea 2 data sets. For Strike and **Chesapeake** data sets MDM has slightly higher Silhouette index value than MMDM.

Table 5-2 Comparison between MMDM and MDM in term of Silhouette index

Data sets	MMDM		MDM	
	$\|C\|$	**Silhouette index**	$\|C\|$	**Silhouette index**
Strike	3	0.400	4	0.413
Karate	2	**0.329**	3	0.140
Mexico	3	**0.220**	3	0.210
Korea2	5	**0.291**	5	0.247
Chesapeake	3	0.120	3	0.210

5.1.3 Modularity Index

Modularity (Q) is one of the quality metrics that measures the difference between the actual density of edges within the cluster and the density of the subgraph in a randomized graph with equivalent number of nodes and edges. Thus, when the value of Q is close to 1 it means the nodes in the community is highly connected. On the other hand, Q begins close to 0 indicates that the fraction of edges inside communities is no better than the random case. In table 5-3 we compare the results of our model (MMDM) and the results of the MDM model from the literature in terms of modularity. Table 5-3 shows that MMDM method has higher modularity value than MDM for Mexico and Chesapeake data set. It also shows that MMDM has lower modularity value than MDM for Strike**,** Karate and Korea2 data sets.

Table 5-3 Comparison between MMDM and MDM in term of Modularity index

Data sets	MMDM		MDM	
	$\lvert C \rvert$	Modularity index	$\lvert C \rvert$	Modularity index
Strike	3	0.521	4	0.561
Karate	2	0.372	3	0.402
Mexico	3	**0.359**	3	0.354
Korea2	5	0.425	5	0.439
Chesapeake	3	**0.266**	3	0.229

5.2 Heuristic Validation

The main purpose of introducing the heuristic approach was to solve large data sets that can't be solved optimally or take very long time to solve. After we stablish the heuristic idea, we tested it on the small data sets that we solved optimally. Table 5-4 shows a comparison of MMDM value achieved optimally and by the heuristic. Compare function in R software was used to check for the equality. The function gives a ratio between 0 and 1, where the value close to 1 means the tow values are similar while value close to 0 means the tow object are different from each other. As it is shown in table 4 most of the ratios are close to 1 which means the values are almost similar which indicates that our heuristic is working properly.

Table 5-4 Comparison of MMDM value achieved optimally and by the heuristic

Data sets	$\lvert \boldsymbol{C} \rvert$	MMDM	Heuristic	Compare Ratio
Strike	3	8.073	6.35	0.87
Karate	2	7.411	7.411	1.00
Korea 2	5	9.448	9.448	1.00
Mexican	3	13.078	13.01	0.94
Chesapeake	3	15.010	14.828	0.94

For large data set, we present the best results obtained by Density Ratio Heuristic applied to MMDM and MDM and compare them in term of internal cluster validation. Table 5-5 shows Dunn Index, Silhouette Index and modularity for the results obtained by Density Ratio Heuristic.

Table 5-5 Density Ratio Heuristic applied to MMDM and MDM and compare them in term of internal cluster validation

	MMDM						MDM					
Data set	$\|C\|$	MD	t(s)	Dunn index	Silhouette index	Modularity	$\|C\|$	D	t(s)	Dunn index	Silhouette index	Modularity
Football	11	76.447	2.74	0.333	0.341	0.603	11	44.388	2.71	0.333	0.341	0.603
Email	20	76.009	9.8	0.143	0.022	0.448	8	19.395	3.8	0.143	0.031	0.455
Facebook	17	614.246	**285.8**	**0.25**	**0.136**	0.712	15	495.944	10.5	0.125	0.095	0.778

As seen in table 5-5, for football network, applying Density Ratio Heuristic to MMDM gives almost similar values to MDM in term of Dunn, silhouette and modularity indices. Similarly, for email network the results were very close to each other with slight priority of MDM in Silhouette and modularity indices. For Facebook network, applying the heuristic to MMDM gives better solution than applying it to MDM in terms of Dunn index and Silhouette indices and lower value in term of modularity index. Table 5-5 also shows that the computational time of applying Density Ratio heuristic on both MMDM and MDM. As seen, our proposed heuristic, beside it solves large networks, it has very low computational time compared to the computational time of solving the problem optimally specially when we apply it to MDM.

CHAPTER 6

CONCLUSION AND RECOMMENDATIONS

In this work, we proposed a modified modularity density maximization clustering method. We consider minimizing the deep furthest connection instead of the total out links which works as an out-connection measure. A MILP formalism was introduced to maximize the modified modularity density value MD and we obtain optimal value for instances up to 39. The obtained results were compared with MDM results in term of internal clustering validation approach. Unfortunately, large data sets cannot be solved by this MILP, so we proposed what we called Density Ratio heuristic. The main idea of the heuristic was to move each node, that has Intra Density Ratio (DR_{intra}) less than Inter Density Ratio (DR_{inter}) , from its current cluster to the other cluster. Density Ratio heuristic was applied to both Modified Modularity Density Maximization (MMDM) and Modularity Density Maximization MDM approaches. The obtained results were compared using internal cluster validation approach.

We proposed a Mixed Integer Linear Programing (MILP) for Modified Modularity Density Maximization MMDM. The mathematical model was applied to data sets from the literature and solved by GAMS software. Our model was able to solve, optimally, instances up to 40 nodes. In the proposed MILP our objective variable α_c is constrained by upper U_α and lower L_α bound. So, having a good upper and lower bound will make the model performance very good. However, it is very difficult to derive an upper and lower bound theoretically as stated in the literature. MMDM obtained results were compared with MDM in term of cluster validation approach. It was found that in most data sets MMDM got higher Dunn, silhouette and modularity indices than MDM got. This indicates that Modified modularity density maximization was able to detect communities that are more compact and well separated from each other than what MDM detected.

The proposed Mixed Integer Linear Programming (MILP) of Modified Modularity Density Maximization (MMDM) was able to solve, optimally, networks with up to 40 instances. Thus, Density Ratio heuristic was proposed to solve large data sets that can't be solved optimally or take very long time to solve. The main idea of the heuristic was to move each node, that has Intra density ratio (DR_{intra}) less than Inter density ratio (DR_{inter}) , from its current cluster to the other cluster. Density Ratio heuristic was applied to both Modified Modularity Density Maximization (MMDM) and Modularity Density Maximization MDM approaches. The obtained results were compared using internal cluster validation approach. It has been found that applying Density Ratio heuristic on MMDM approaches can find better Dunn and Silhouette values than applying Density Ratio heuristic on MDM for some data sets and similar for some other data sets. For modularity, applying the heuristic to MDM is giving higher values than applying it to MMDM.

Some of the future work would be to introduce a pre-partitioning method before applying the MILP of MMDM. This could help in improving the ability of MILP to solve larger data sets. Also, more enhancement could be done to the MILP of MMDM by applying constrained clustering approach. In addition, improving the upper and lower bound for MILP of MMDM could be a significant future contribution. For Density Ratio heuristic, finding a good initial solution could improve the results significantly. Someone could run the heuristic with different initial solutions from Lagrangian relaxation, hierarchical clustering or any similar approaches.

.

References

[1] S. Fortunato, “Community detection in graphs,” *Phys. Rep.*, vol. 486, no. 3–5, pp. 75–174, 2010.

[2] M. Wang, C. Wang, J. X. Yu, and J. Zhang, “Community detection in social networks,” *Proc. VLDB Endow.*, vol. 8, no. 10, pp. 998–1009, 2015.

[3] A Pothen, “GRAPH PARTITIONING ALGORITHMS 1 Introduction,” *ieeexplore.ieee.org*, 1997.

[4] B. W. kernighan and S. Lin, “[4].pdf.” 1970.

[5] S. Cell and D. Environment, “An Algorithm for Quadrisection and Its Application to Standard Cell Placement,” *ieee Trans. circuits Syst.*, vol. 35, no. 3, pp. 294–303, 1988.

[6] G. W. Flake, S. Lawrence, G. W. Flake, S. Lawrence, C. L. Giles, and F. M. Coetzee, “Self-Organization and Identification of Web Communities Self-Organization of the Web and Identification of Communities,” no. March, 2014.

[7] J. Macqueen, “SOME METHODS FOR CLASSIFICATION AND ANALYSIS OF MULTIVARIATE OBSERVATIONS,” *Proc. fifth berkeley Symp. Math. Stat. Probab.*, vol. 233, no. 233, pp. 281–297, 1967.

[8] James C. Bezdek, *PATTERN RECOGNITION WITH FUZZY OBJECTIVE FUNCTION ALGORITHMS*. 1981.

[9] A. H. WE Donath, “Lower Bounds for the Partitioning of Graphs.” 1973.

[10] A. Y. Ng and M. I. Jordan, “On Spectral Clustering : Analysis and an algorithm,” *Adv. Neural Inf. Process. Syst.*, 2002.

[11] M. Girvan and M. E. J. Newman, “Community structure in social and biological networks,” *Proc. Natl. Acad. Sci.*, vol. 99, no. 12, pp. 7821–7826, 2002.

[12] M. E. J. Newman and M. Girvan, “Finding and evaluating community structure in networks,” *Phys. Rev. E*, pp. 1–16, 2003.

[13] F. Slanina and Y. C. Zhang, “Referee networks and their spectral properties,” *Acta Phys. Pol. B*, vol. 36, no. 9, pp. 2797–2804, 2005.

[14] A. Capocci, V. D. P. Servedio, G. Caldarelli, and F. Colaiori, “Detecting communities in large networks,” *Phys. A Stat. Mech. its Appl.*, no. m, pp. 1–4, 2004.

[15] N. A. Alves, “Unveiling community structures in weighted networks,” *Phys. Rev. E - Stat. Nonlinear, Soft Matter Phys.*, vol. 76, no. 3, 2007.

[16] J. Reichardt and S. Bornholdt, “Detecting fuzzy community structures in complex networks with a potts model,” *Phys. Rev. Lett.*, vol. 93, no. 21, pp. 2–5, 2004.

[17] C. Systems, “Super-paramagnetic clustering of data,” 1997.

[18] A. Js, C. Sb, and T. I. Society, “References 1.,” vol. 5, no. 1993, pp. 235–246, 1995.

[19] H. Zhou and R. Lipowsky, “Dynamic pattern evolution on scale-free networks,” *Proc. Natl. Acad. Sci. United States Am. Sci.*, vol. 102, no. 29, pp. 10052–10057,

2005.

[20] M. Chen, K. Kuzmin, and B. K. Szymanski, "Community detection via maximization of modularity and its variants," *IEEE Trans. Comput. Soc. Syst.*, vol. 1, no. 1, pp. 46–65, 2014.

[21] M. E. J. Newman, "Modularity and community structure in networks," *Proc. Natl. Acad. Sci.*, vol. 103, no. 23, pp. 8577–8582, 2006.

[22] E. A. Leicht and M. E. J. Newman, "Community structure in directed networks," *Phys. Rev. Lett.*, pp. 1–5, 2007.

[23] M. E. J. Newman, "Analysis of weighted networks," *Phys. Rev. E - Stat. Physics, Plasmas, Fluids, Relat. Interdiscip. Top.*, vol. 70, no. 5, p. 9, 2004.

[24] U. Brandes *et al.*, "On modularity clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 20, no. 2, pp. 172–188, 2008.

[25] M. E. J. Newman, "Fast algorithm for detecting community structure in networks," *Phys. Rev. E*, no. 2, pp. 1–5, 2003.

[26] A. Clauset, M. E. J. Newman, and C. Moore, "Finding community structure in very large networks," *Phys. Rev. E*, pp. 1–6, 2004.

[27] K. Wakita, "Finding Community Structure in Mega-scale Social Networks," *Proc. 16th Int. Conf. World Wide Web*, 2007.

[28] V. D. Blondel, J. Guillaume, and E. Lefebvre, "Fast unfolding of communities in large networks," *J. Stat. Mech.*, pp. 1–12, 2008.

[29] M. E. J. Newman, “Finding community structure in networks using the eigenvectors of matrices,” *Phys. Rev. E*, 2006.

[30] T. Richardson, P. J. Mucha, and M. A. Porter, “Spectral Tripartitioning of Networks,” *Phys. Rev. E*, pp. 1–12, 2009.

[31] S. White and P. Smyth, “A Spectral Clustering Approach To Finding Communities in Graphs,” *Proc. 2005 SIAM Int. Conf. Data Min.*, 2005.

[32] C. Paper and S. Antonio, “An Efficient Spectral Algorithm for Network Community Discovery and Its Applications to Biological and Social Net .... An Efficient Spectral Algorithm for Network Community Discovery,” *ieeexplore.ieee.org*, no. November 2007, 2014.

[33] M. E. J. Newman, “Spectral methods for network community detection and graph partitioning,” *pdfs.semanticscholar.org*, 2013.

[34] C. P. Massen and J. P. K. Doye, “Identifying ‘communities’ within energy landscapes,” *Phys. Rev. E*, pp. 1–13, 2005.

[35] A. Manuscript, “NIH Public Access,” vol. 2005, pp. 1–17, 2007.

[36] A. Manuscript, “Functional cartography of complex metabolic networks,” *Nature*, vol. 433, no. 7028, pp. 895–900, 2008.

[37] C. O. Ã. Dorso, “via global optimization $,” vol. 358, pp. 593–604, 2005.

[38] H. Pirim, “A Comparison of Heuristics with Modularity Maximization Objective using Biological Data Sets,” *MATEC Web Conf.*, vol. 1, pp. 1–5, 2016.

[39] G. Agarwal and D. Kempe, “P HYSICAL J OURNAL B Modularity-maximizing graph communities via mathematical programming,” *Eur. Phys. J. B,* vol. 418, pp. 409–418, 2008.

[40] G. Xu, S. Tsoka, and L. G. Papageorgiou, “P HYSICAL J OURNAL B Finding community structures in complex networks using mixed ineger optimization,” *Eur. Phys. J. B,* vol. 239, pp. 231–239, 2007.

[41] D. Aloise and L. Liberti, “Column generation algorithms for exact modularity maximization in networks,” *Phys. Rev. E,* pp. 1–9, 2010.

[42] Z. Li, S. Zhang, R. Wang, X. Zhang, and L. Chen, “Quantitative function for community detection,” *Phys. Rev. E,* pp. 1–9, 2008.

[43] A. Costa, “Discrete Optimization MILP formulations for the modularity density maximization problem,” *Eur. J. Oper. Res.,* vol. 245, no. 1, pp. 14–21, 2015.

[44] A. Costa, T. Sheng, and L. Xuan, “Discrete Optimization Complete mixed integer linear programming formulations for modularity density based clustering,” *Discret. Optim.,* vol. 25, pp. 141–158, 2017.

[45] A. D. Rikun, “A Convex Envelope Formula for Multilinear Functions,” *J. Glob. Optim.,* vol. 10, pp. 425–437, 1997.

[46] F. Tardella, “Existence and sum decomposition of vertex polyhedral convex envelopes,” *Optim. Lett.,* vol. 2, no. 3, pp. 363–375, 2008.

[47] G. P. McCormick, “Computability of global solutions to factorable nonconvex programs: Part I - Convex underestimating problems,” *Math. Program.,* vol. 10,

no. 1, pp. 147–175, 1976.

[48] A. Costa, P. Hansen, and L. Liberti, “On the impact of symmetry-breaking constraints on spatial Branch-and-Bound for circle packing in a square,” *Discret. Appl. Math.*, vol. 161, no. 1–2, pp. 96–106, 2013.

[49] W. Dinkelbach, “On Nonlinear Fractional Programming,” *Manage. Sci.*, vol. 13, no. 7, pp. 492–498, 1967.

[50] A. Y. Y. YOICHI IZUNAGA, TOMOMI MATSUI, “A doubly nonnegative relaxation for modularity density maximization,” pp. 84–97, 2016.

[51] R. De Santiago and L. C. Lamb, “Computers and Operations Research Exact computational solution of Modularity Density Maximization by effective column generation,” *Comput. Oper. Res.*, vol. 86, pp. 18–29, 2017.

[52] K. Sato and Y. Izunaga, “A BRANCH-AND-PRICE APPROACH WITH MILP FORMULATION TO MODULARITY DENSITY MAXIMIZATION ON GRAPHS,” *arXiv Prepr. arXiv*, pp. 1–25, 2017.

[53] A. Costa, S. Kushnarev, L. Liberti, and Z. Sun, “Computers & Operations Research Divisive heuristic for modularity density maximization,” *Comput. Oper. Res.*, vol. 71, pp. 100–109, 2016.

[54] S. Cafieri, P. Hansen, L. Liberti, S. Cafieri, P. Hansen, and L. Liberti, “Locally optimal heuristic for modularity maximization of networks,” *Phys. Rev. E*, 2014.

[55] S. Cafieri, A. Costa, and P. Hansen, “Reformulation of a model for hierarchical divisive graph modularity maximization,” *Ann. Oper. Res.*, pp. 213–226, 2014.

[56] A. Costa and P. Hansen, “A locally optimal hierarchical divisive heuristic for bipartite modularity maximization,” *Optim. Lett.*, pp. 903–917, 2014.

[57] R. Santiago and L. C. Lamb, “Efficient modularity density heuristics for large graphs,” *Eur. J. Oper. Res.*, vol. 258, no. 3, pp. 844–865, 2017.

[58] R. Shang, W. Zhang, L. Jiao, R. Stolkin, and Y. Xue, “A community integration strategy based on an improved modularity density increment for large-scale networks,” *Phys. A Stat. Mech. its Appl.*, vol. 469, pp. 471–485, 2017.

[59] S. Fortunato and M. Barthe, “Resolution limit in community detection ´,” *Proc. Natl. Acad. Sci.*, vol. 104, no. 1, 2007.

[60] B. H. Good, Y. De Montjoye, and A. Clauset, “The performance of modularity maximization in practical contexts,” *Phys. Rev. E*, pp. 1–20, 2010.

[61] R. Fortet, “L’algebre de boole et ses applications en recherche operationnelle (*),” *Springer*, no. 1, p. 8, 1960.

[62] “CPLEX Solver (Version 12.8) Retrieved from.” [Online]. Available: www.cplex.com.

[63] R. E. Rosenthal, “A GAMS Tutorial.” [Online]. Available: https://www.gams.com/latest/docs/UG_Tutorial.html.

[64] “igraph – The network analysis package.” [Online]. Available: https://igraph.org/r/.

[65] “Find, install and publish Python packages with the Python Package Index.”

[Online]. Available: https://pypi.org/.

[66] J. L. and A. Krevl, “SNAP Datasets: Stanford Large Network Dataset Collection.” [Online]. Available: http://snap.stanford.edu/data.

[67] M. Halkidi, Y. Batistakis, and M. Vazirgiannis, “Cluster validity methods: Part I,” *SIGMOD Rec.*, vol. 31, no. 2, pp. 40–45, 2002.

[68] F. Kovács, C. Legány, and A. Babos, “Cluster Validity Measurement Techniques,” *Proc. 6th Int. Symp. Hungarian Res. Comput. Intell.*, pp. 1–11, 2005.

[69] L. Jegatha Deborah, R. Baskaran, and A. Kannan, “A Survey on Internal Validity Measure for Cluster Validation,” *Int. J. Comput. Sci. Eng. Surv.*, vol. 1, no. 2, pp. 85–102, 2010.

[70] F. Ghshqgv *et al.*, “&oxvwhu 9dolglw\ ,qglfhv iru *udsk 3duwlwlrqlqj.”

[71] P. J. Rousseeuw, “Silhouettes: A graphical aid to the interpretation and validation of cluster analysis,” *J. Comput. Appl. Math.*, vol. 20, no. C, pp. 53–65, 1987.

APPENDIX A

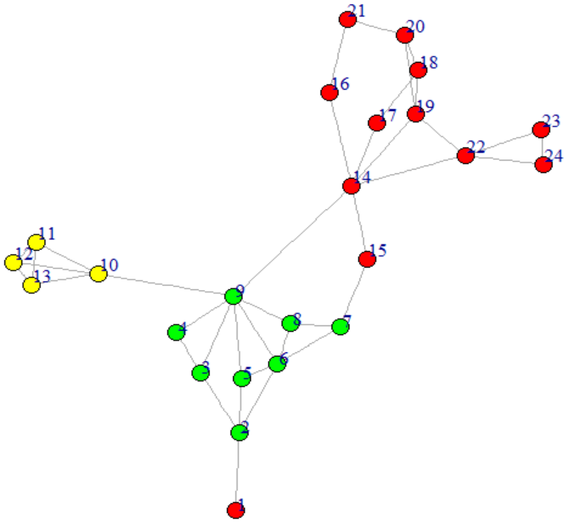


Figure A-1 Optimal solution obtained by MILP of MMDM for Strike data set

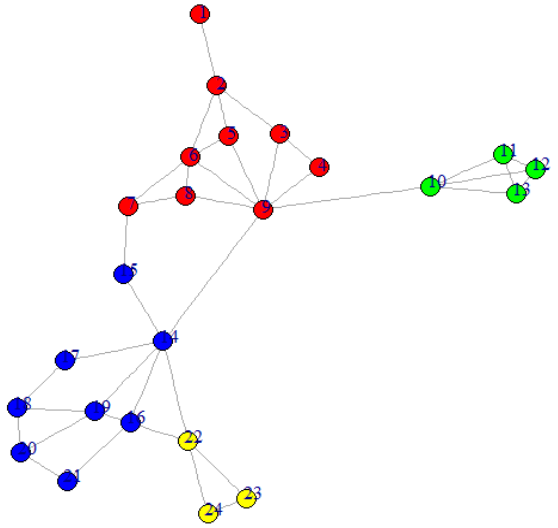


Figure A-2 Optimal solution obtained by MILP of MDM for Strike data set

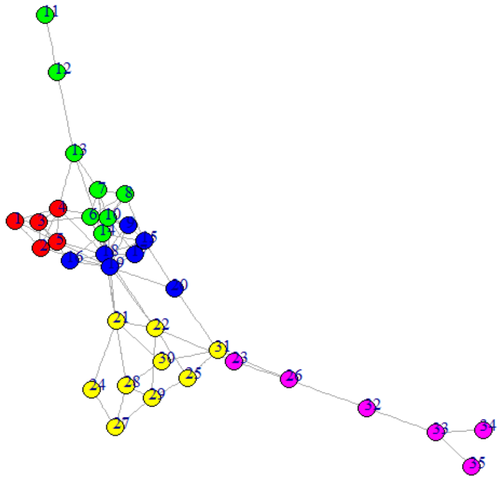


Figure A-3 Optimal solution obtained by MILP of MMDM for Korea2 data set

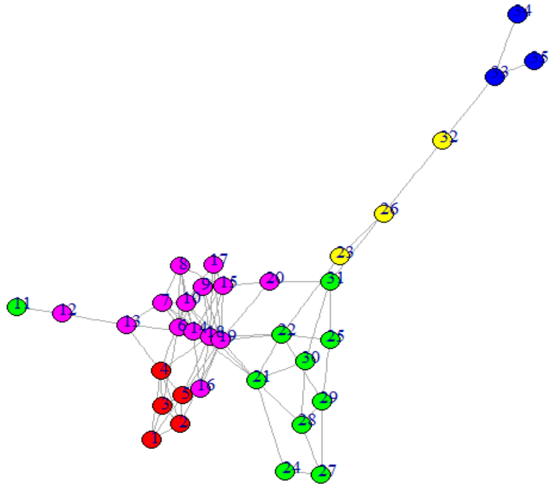


Figure A-4 Optimal solution obtained by MILP of MDM for Korea2 data set

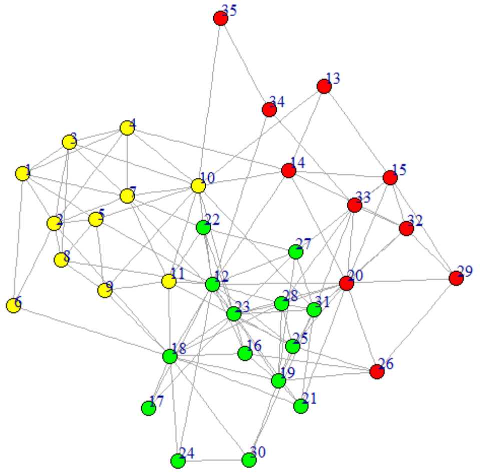


Figure A-5 Optimal solution obtained by MILP of MMDM for Mexico data set

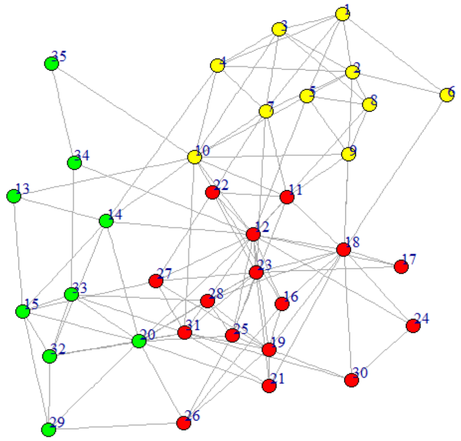


Figure A-6 Optimal solution obtained by MILP of MDM for Mexico data set

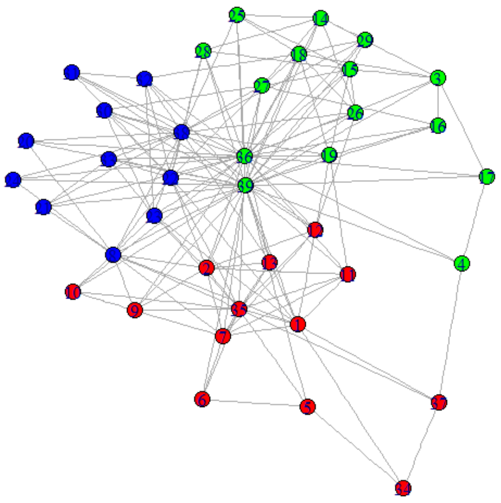


Figure A-7 Optimal solution obtained by MILP of MMDM for Chesapeake data set

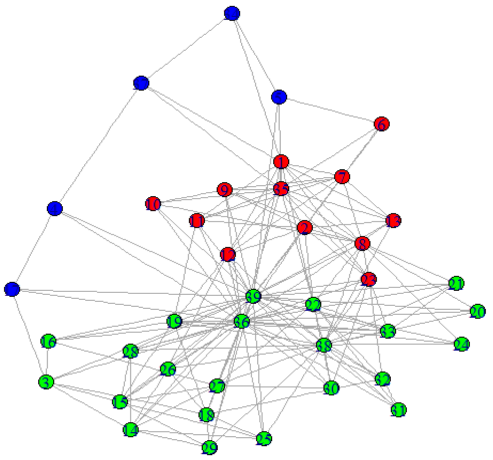


Figure A-8 Optimal solution obtained by MILP of MDM for Chesapeake data set

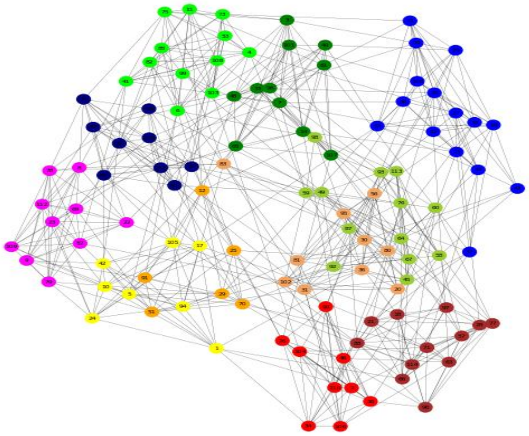


Figure A-9 Best value obtained by applying Density Ratio heuristic on MMDM for Football data set

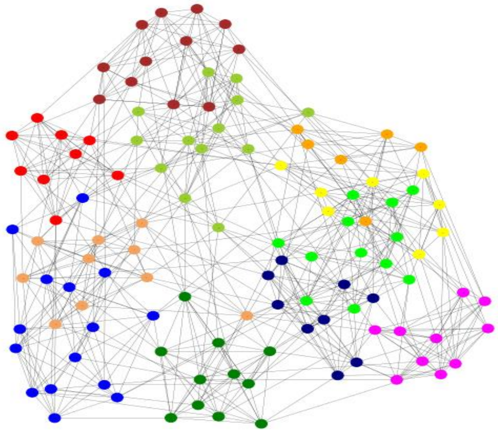


Figure A-10 Best value obtained by applying Density Ratio heuristic on MDM for Football data set

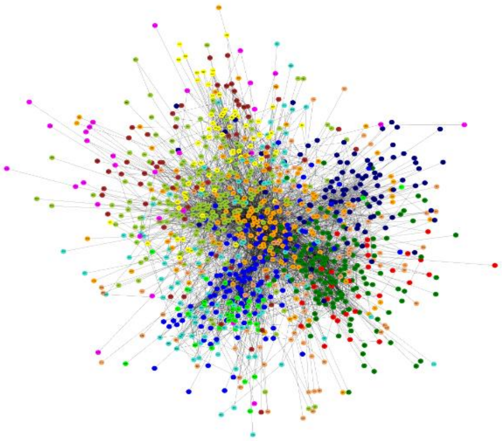


Figure A-11 Best value obtained by applying Density Ratio heuristic on MMDM for Email data set

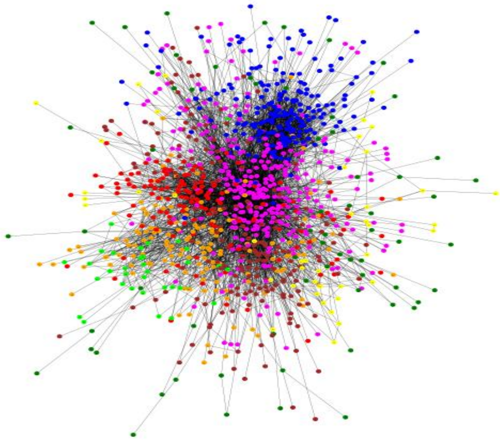


Figure A-12 Best value obtained by applying Density Ratio heuristic on MDM for Email data set

VITAE

Name :Zead Hamed Abdul Jaleel Saleh

Nationality :Yemeni

Date of Birth :3/1/1989

Email : zeadalshamiri2013@gmail.com

Address : Taiz (zip code/84657465), Yemen

Academic Background :

May 2016: BS. in Industrial Eng., KFUPM. (Second Honor).

December 2019: MS. in Industrial Eng., KFUPM. (Second Honor).

Experience :

June – August 2015: Trainee, worked in the flowing area (supply chain and logistics).

Worked as Lab instructor in Work Study and Process Improvements course at king Fahd university for 2 semesters.

Worked as Lab instructor in stochastic systems simulation course at king Fahd university for one semester.

Research Interest :

Data Mining specially graph mining.

Operation Research and inventory control.