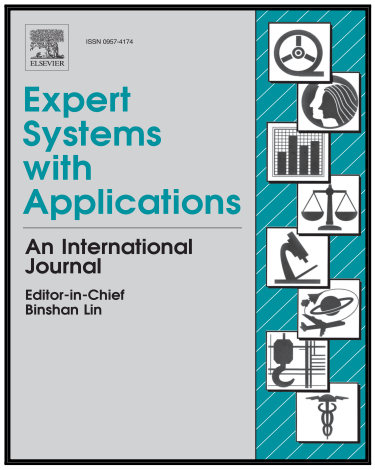
Accepted Manuscript



Community Detection and Influential Node Identification in Complex Networks using Mathematical Programming

Sharan Srinivas, Chandrasekharan Rajendran

PII: S0957-4174(19)30394-X
DOI: https://doi.org/10.1016/j.eswa.2019.05.059
Reference: ESWA 12712

To appear in: *Expert Systems With Applications*

Received date: 6 January 2019
Revised date: 8 May 2019
Accepted date: 31 May 2019

Please cite this article as: Sharan Srinivas, Chandrasekharan Rajendran, Community Detection and Influential Node Identification in Complex Networks using Mathematical Programming, *Expert Systems With Applications* (2019), doi: https://doi.org/10.1016/j.eswa.2019.05.059

This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

ACCEPTED MANUSCRIPT

Highlights

- Studied the structure of complex relational networks
- Proposed mathematical model for finding community structures and influential nodes
- Evaluated proposed model by testing it on various real-life network datasets
- Results indicate promising performance compared to existing approaches
- Provides faster coverage and better call for response when applied to networks

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

TITLE PAGE

Community Detection and Influential Node Identification in Complex Networks using Mathematical Programming

Sharan Srinivas*

Department of Industrial and Manufacturing Systems Engineering, College of Engineering, and Department of Marketing, Trulaske College of Business,
University of Missouri, Columbia, MO 65211
Email: Dr.SharanSrinivas@gmail.com
Phone: +1 (607) 768 - 3757

Chandrasekharan Rajendran

Department of Management Studies,
Indian Institute of Technology Madras Chennai, India
Email: craj@iitm.ac.in
Phone: +91 44 2257 4559

***Corresponding Author:** Sharan Srinivas

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

Article Type: Short Communication

Revised Version 2

Community Detection and Influential Node Identification in Complex Networks using Mathematical Programming

Abstract

Integer programming models for community detection in relational networks have diverse applications in different fields. From making our lives easier by improving search engine optimization to saving our lives by aiding in threat detection and disaster management, researches in this niche have added value to human experience and knowledge. Besides the community structure, the influential nodes or members in a complex network are highly effective at diffusing information quickly to others in the community. Prior research dealing with the use of optimization models for clustering networks has independently focused on detecting communities. In this research, we propose a new integer linear programming model to detect community structure in real-life networks and also identify the most influential node within each community. We validate the proposed model by testing it on a well-established community network. Further, the performance of the proposed model are evaluated by comparing it with the existing best performing optimization model as well as three heuristic approaches for community detection. The experimental results indicate that in most cases the proposed integer programming model performs better than the existing optimization model with respect to modularity, Silhouette coefficient and computational time. Besides, our model yields superior Silhouette and competitive modularity values compared to the heuristic approaches in many cases.

Keywords: Clustering, Networks, Community detection, Influential nodes, Integer linear programming.

1. Introduction

A large number of complex systems can be expressed through networks, which is a group of nodes associated through edges or links (Raghavan et al., 2007). Social networks such as Facebook and Twitter are such examples, where the users are represented by the nodes and the ties/friendship between them is indicated by edges. Similarly, in supply chain logistics, nodes represent the supplier and customer locations, while edges denote the paths connecting

ACCEPTED MANUSCRIPT

the nodes. Due to its theoretical importance and practical applications, research on social, technological, physical and biological networks has gained importance in different branches of sciences. Representing the complex systems through relational networks provides an insight into the intricacy and the dynamic behavior of such systems (Pan et al., 2014). A network is said to exhibit a community structure if its nodes can be grouped into different clusters such that the nodes within the clusters are densely connected compared to the nodes outside the cluster (Radicchi et al., 2004). Finding communities in relational networks takes us one step further to better understanding complex systems, which in turn results in numerous benefits and applications (Girvan and Newman, 2002).

Communities are formed by clustering the nodes in a relational network, where the nodes (or objects) are placed into groups such that the objects in the same group have maximum similarities, while the ones in different groups have the least in common. Since the parameters used for clustering, such as the number of clusters and degree of closeness, vary according to the study and application, there have been numerous methods and algorithms developed for use (e.g., Girvan and Newman, 2004; Pons and Latapy, 2005; Raghavan et al., 2007; Pirim et al., 2018). Some of the most common clustering methods include partitioning, hierarchical agglomeration, and hierarchical division (Larose and Larose, 2015). Network partitioning method is one of the standard approaches for clustering, in which the entire network is divided into a fixed number of user-specified clusters (Pirim et al., 2018). On the other hand, certain approaches, such as hierarchical division and hierarchical agglomeration, automatically determine the number of communities by studying the network structure (James et al., 2013). While hierarchical division splits the entire network successively into smaller clusters until the objects within the clusters are similar, hierarchical agglomeration uses a bottom-up approach by grouping nodes that are similar through successive iterations. The communities detected by these algorithms are evaluated using internal validation metrics such as modularity (assesses the strength of partition by comparing the fraction of within-community edges to the fraction of randomly distributed edges) or Silhouette coefficient (measures similarity within cluster and dissimilarity between clusters).

Even though solution approaches related to heuristic algorithms are computationally efficient in most cases, they do not guarantee the global optimal community structure that maximizes (or minimizes) a specific criterion (e.g., distance within a community). To overcome this limitation, some studies have developed optimization or Operations-Research (OR)-based models to find the optimal community structure (e.g., Agarwal and Kempe, 2008; Xu et al., 2010; Lin et al., 2015). Even though these optimization models become computationally intractable for large

ACCEPTED MANUSCRIPT

networks, they provide meaningful insight and patterns on detecting the optimal community structure, which often forms the basis for the development of new heuristic algorithms (e.g., Agarwal and Kempe, 2008; Pirim et al., 2018). The objective of our research is to propose an optimization model that identifies communities characterized by coherent nodes within a cluster and sparser links across communities.

We have identified the following gaps pertaining to the literature on partitioning a network into non-overlapping communities (or clusters). Influential member, a node that is closest to all the other nodes within the community, is typically present in the case of group formation with related community characteristics such as shared ideology, beliefs or behavior (Aral and Walker, 2012). Identification of an influential node within each community is crucial as it enables effective dissemination of information, which is useful in real-life situations such as mitigation of disasters, emergencies, and political crisis (Tulu et al., 2018). To the best of our knowledge, existing community detection algorithms are only able to partition the network into clusters (e.g., Girvan and Newman, 2004; Pons and Latapy, 2005; Blondel et al., 2008; Pirim et al., 2018), and do not consider a model that jointly identifies the community structure as well as the community-specific influential member. Second, most prior research considers only one performance metric (e.g., modularity) to evaluate the strength of the community structure (e.g., Raghavan et al., 2007; Xu et al., 2007; Agarwal and Kempe, 2008). However, a community partition that yields a good value on one metric may result in a poor score on another validation index (Arbelaitz et al., 2013). Therefore, to ensure a robust community structure, the strength of the community partition must be assessed using multiple metrics. Finally, the current modeling approach adopted in developing an optimization model makes it computationally intractable even for medium-sized networks (e.g., Saglam et al., 2006; Xu et al., 2007; Agarwal and Kempe, 2008; Cafieri and Hansen, 2014; Pirim et al., 2018), thereby limiting its potential to gain insight on relatively large networks.

Our work aims to overcome the aforementioned gaps in the literature. We propose a novel integer programming model (also referred to as OR model, optimization model or mathematical programming model) for partitioning a network into non-overlapping communities such that the intra-community distance is minimized (compact community where nodes are densely connected) and inter-cluster distance is maximized (nodes across communities are well-separated). Unlike the previous models and algorithms on community detection, our proposed approach concurrently identifies the community structure and the influential nodes associated with each community. Further, we consider multiple performance measures to evaluate the community structure. Finally, we adopt a novel approach to formulate the optimization model and ana-

ACCEPTED MANUSCRIPT

lyze its capability to solve relatively large-sized networks (with more than 1000 nodes) with respect to the chosen measures of performance (e.g., modularity, Silhouette coefficient) and computational time.

The remainder of the paper is structured as follows. Section 2 provides a brief literature review by highlighting some notable works on community detection in networks. Section 3 presents the problem statement. Section 4 first describes our proposed optimization model. We also provide a discussion in Section 4.1 on the commonly used measures of performance related to cluster formation and community detection. Besides, we present a description of the solution methodology in Section 4.2. Section 5 presents the details of computational experiments with the associated results and discussion. Section 6 relates the proposed model to real-life situations. Conclusions and scope for future work are presented in Section 7.

2. Literature Review

Extensive research has been done for partitioning a complex network into communities (e.g., Xie et al., 2013; Yang et al., 2016). The pioneering works of Zachary (1977), Girvan and Newman (2002), and Newman and Girvan (2004) triggered the research in this direction. The identification of cohesive groups or clusters within a social network is a key objective of any community detection algorithm (Pirim et al., 2018). Fortunato and Castellano (2012) provided a survey of community structure in graphs and describe methods developed over the last 10 years in different disciplines such as computer science, social science and physics. Further, the authors also provided a detailed review of quality functions used to evaluate communities along with its limitations. Recently, Bedi and Sharma (2016) conducted a comprehensive review of existing community detection algorithms or approaches. Likewise, Buluc et al. (2016) presented the recent advances in algorithms on network partitioning and its applications such as community detection in power grids and biological networks. Typically, the approaches adopted for detecting communities can be broadly classified into two categories: optimization and heuristic methods. In this section, we review some of the notable works pertaining to these two approaches.

2.1. Optimization-based Approaches for Community Detection

An optimization-based approach detects a community structure by finding the global optimal solution that maximizes or minimizes a specific criterion (i.e., inter-cluster distance, intra-cluster similarity) while satisfying a set of constraints (restrictions) imposed on the network (e.g., Rao, 1971; Glover and Kochenberger, 2006; Saglam et al., 2006; Agarwal and Kempe, 2008). Rao (1971) proposed two different integer programming models for clustering - a non-linear model

ACCEPTED MANUSCRIPT

that minimizes the within-cluster sum of squares and the sums of averaged square distances across clusters, and a linear model that minimizes the total as well as the maximum group distance for forming effective clusters. A combinatorial method was proposed by Mehrotra and Trick (1998) to solve the clustering problems based on a graph partitioning method. The authors tested their model on four small networks which had fewer than 40 nodes. Saglam et al. (2006) formulated a mixed-integer programming model for clustering such that the objective function minimizes the maximum cluster diameter among all the clusters. Their proposed model was used to segment customers based on transactional factors obtained from a broadcasting company. While their model outperformed existing non-linear models, its computational performance was inferior compared to other algorithms in the literature. Glover and Kochenberger (2006) used a clique partitioning formulation for clustering and their node-based method associates variable with nodes, and not edges. They conducted computational experiments and showed that their optimization model produced better Silhouette coefficients compared to the standard k-means clustering heuristic.

Tan et al. (2007) proposed a mixed integer non-linear program for clustering gene expressions and solved it using the Benders Decomposition method. Their algorithm produced competitive intra-cluster similarity and inter-cluster dissimilarity when compared to four existing clustering algorithms, namely, k-means, k-medians, k-Corr, and k-CityBlock. Brandes et al. (2007) used an integer linear program model for clustering with the goal of maximizing the modularity measure. Their optimization model was tested on two small real-life networks and shown to produce superior results compared to a greedy algorithm. Agarwal and Kempe (2008) improved the computation time required to form clusters with maximum modularity by using a combination of relaxed integer linear program and rounding algorithm. Cafieri and Hansen (2014) developed a mixed integer optimization model to improve the clusters established by heuristic approaches, and conducted extensive computational experiments to illustrate its effectiveness. Martins (2016) proposed an optimization model to partition an undirected weighted network into sub-graphs and applied it to biological networks. Recently, Pirim et al. (2018) proposed a mixed integer linear programming model to form compact and separate clusters. The authors used real-life datasets and evaluated their model using multiple performance measures. The results indicated that their model resulted in better Silhouette coefficient for most cases evaluated. Besides, they also observed that without being designed to maximize the modularity, their model provided values that are competitive to the ones produced by a heuristic algorithm. Hence, the model by Pirim et al. (2018) serves as a benchmark for our work.

ACCEPTED MANUSCRIPT

2.2. Heuristic Methods for Community Detection

There are three types of heuristic approaches that are widely used in community detection algorithms - divisive, agglomerative and diffusive. Divisive approaches initially consider all the nodes in a network to be in the same community and then recursively split into smaller communities based on a criterion (e.g., Girvan and Newman, 2002; Radicchi et al., 2004; Arasteh and Alizadeh, 2019). Newman and Girvan (2004) introduced a divisive approach to identify communities in networks by using edge-betweenness as a critical measure for clustering. Since the edges connecting nodes from different communities have a higher betweenness value compared to those connecting nodes within a community, their algorithm successively removes edges with high betweenness values resulting in a network with edges only within the community members. Moreover, they also introduced the measure of modularity to evaluate the quality of clusters or communities formed by an algorithm. Subsequently, researchers developed heuristic and meta-heuristic-based search methods for community detection (e.g., Pizzuti, 2012; Zadeh and Kobti, 2015; Guerrero et al., 2018; Ji et al., 2019).

On the other hand, agglomerative clustering approaches consider each node as a separate community at the outset and then iteratively combines pairs of closest communities until there is no improvement in the chosen performance measure (such as modularity). Generally, agglomerative clustering algorithms are computationally expensive (Gaume, 2004; Yen et al., 2005). To overcome this difficulty, Pons and Latapy (2005) proposed the Walktrap algorithm, where a random walk process is used to measure the structural similarity between two vertices in the network. Their algorithm iteratively groups the nodes to communities depending on their similarity. Likewise, Blondel et al. (2008) developed an agglomerative heuristic, Louvain method, which adopts a greedy approach to maximize modularity. The Louvain algorithm initially treats each node as a cluster and then groups neighboring nodes to form a community if it results in the highest increase in the objective. This procedure is stopped only when there is no further improvement in the modularity value.

A diffusive approach of label propagation to cluster networks was introduced by Raghavan et al. (2007), where the nodes are given different labels denoting the community to which they belong and then during successive iterations, the nodes assume the label that is most shared by its neighbors. By this way, the label tends to propagate until it is feasible to do so, and then the nodes having the same labels are grouped into a cluster or community. Subsequently, many researchers have worked on different variants of the label propagation algorithm (e.g. Zhang et al., 2017; Deng et al., 2019).

Recently, Yang et al. (2016) conducted a comparative analysis on the performance of eight

ACCEPTED MANUSCRIPT

state-of-the-art heuristic algorithms using 100 randomly generated small (fewer than 1000 nodes) and large networks. The authors concluded that the Walktrap algorithm (Pons and Latapy, 2005) and Louvain method (Blondel et al., 2008) consistently performed well on small and large dataset, while the edge-betweenness algorithm (Newman and Girvan, 2004) performed well only when the number of nodes in a network was less than 1000. Hence, the performance of the mathematical model proposed in our work is compared against these three heuristic approaches.

3. Problem Statement

In this paper, we consider an undirected network characterized by a set of nodes and edges. A node can represent different entities depending on the type of the network under study, while an edge represents the connection (or relationship) between any two nodes. For example, in a social network, a person is represented as a node and the interaction (or friendship) between two people is represented as an edge. Further, the network is not homogeneous. Thus, closely interconnected nodes can be clustered in localized areas of the network resulting in two or more communities. Besides, each community has an influential member, a node that is close to the other nodes within the community. Figure 1 illustrates the network with two communities and influential nodes.

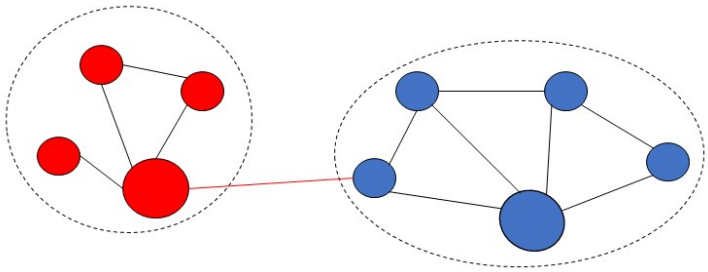


Figure 1: A network with two communities, enclosed in dashed lines, along with their influential nodes (identified by their bigger size)

As reported by Zachary (1977) in a pioneering study (with a great impact on related research in the years to follow) on fission in small groups, the formalization of ethnographic relationship of people (for example, in a political crisis and the associated fission process related to factions) in the form of mathematical models is necessary to understand the strengths or confrontations (e.g. related to ideology) within or among the factions. As the basis for the study on social relationships, Zachary observed the karate club based in a university and its activities over three years, especially related to the conflict between the influential members, namely, its President and the Instructor, leading to two subgroups that had improved friendship bonds within a subgroup in comparison to the group as a whole. Quite often, this real-life study on the karate

ACCEPTED MANUSCRIPT

club serves as a key reference or benchmark for most studies on community detection (e.g. Raghavan et al., 2007; Pirim et al., 2018).

The network graph G with N nodes ($i, j = 1, 2, ..., N$) can be represented as a $N \times N$ adjacency matrix (A_{NN}), where the element of the matrix is 1 if node i is connected to node j and 0 otherwise. Since it is an undirected graph, A_{NN} is symmetric (i.e., $A_{NN} = A_{NN}^T$). In other words, if A_{ij} denotes the entry in row i and column j of A_{NN}, then $A_{ij} = A_{ji}$ for every i and j. Moreover, every node in network G is either directly or indirectly connected to every other node in the network. If nodes i and j are directly connected, then the shortest distance between them (D_{ij}) is 1. On the other hand, if j can only be reached from node i through $N'(< N)$ other nodes, then D_{ij} is equal to $N' + 1$.

As the network is heterogeneous, G can be grouped into C sub-graphs ($l = \{1, 2, ..., C\}$ and $l \subset G$). Each sub-graph l will have $N_l < N$ distinct nodes and one influential node j which most frequently acts as a common link on the shortest distance between the other nodes in the community. Moreover, in our study, the communities are separated and assumed not to overlap. Therefore, each node i in network G belongs to exactly one community l. Hence, the total number of possible edges within community l is $N_l(N_l - 1)/2$, while the maximum possible edges from community l to other communities in the network is $N_l(N - N_l)$.

The degree of node i within its community, K_i^{in}, is equal to the number of edges connecting node i to other vertices in the same community. On the other hand, the degree of node i outside its community, K_i^{out}, is the number of direct connections to the nodes that belong to different communities. The total degree of node i is $K_i = K_i^{in} + K_i^{out}$. If $\mathcal{S}_l$ denotes the set of nodes in community l, then the actual number of edges within community l is $\sum_{i \in S_l} K_i^{in}/2$ and the actual number of inter-cluster edges of community l is $\sum_{i \in S_l} K_i^{out}$. The intra-community density of community l (ρ_l^{int}) is $\dfrac{\sum_{i \in S_l} K_i^{in}/2}{N_l(N_l - 1)/2}$, while the inter-community density of community l (ρ_l^{ext}) is $\dfrac{\sum_{i \in S_l} K_i^{out}}{N_l(N - N_l)}$. A community l is considered to be strong if it has high intra-community density and low inter-community density (i.e., $\rho_l^{int} > \rho_l^{ext}, l = 1, 2, ..., C$).

Thus, given network G and the number of communities to form (C), the objective of this research is to find the nodes and influential member belonging to each community l such that the nodes are closely connected (i.e., high cohesiveness being related to adjacency and the distance of a node from its corresponding influential node) within a given community and sparsely linked to other communities (i.e., separation from the unrelated communities).

4. Proposed Model for Detecting Communities and Locating Influential Nodes

The proposed integer linear programming model aims to form compact clusters coupled with finding the influential or nucleus node that shares the maximum relationship with other nodes in the same cluster. A common approach is to determine whether to assign node i to a given cluster l associated with a binary variable x_{il} (Pirim et al., 2018). In our model, we determine whether a given node i can be assigned to an influential node j associated with a binary variable x_{ij}. Note that a given node i can also act as an influential node by itself. This definition of the binary variable enables us to achieve simultaneously the twin objectives of determining the set of influential nodes and identifying cohesive and compact community formations. The proposed model is developed by keeping these considerations while clustering.

Parameters

N	Number of nodes (or vertices) in the network
C	Number of clusters to be established
A_{ij}	Adjacency of two nodes i and j, where A_{ij} is 1 if nodes i and j have a direct connection and 0 otherwise
K_j	Degree of vertex j (number of edges connected to node j, $\sum_{\substack{i=1 \\ i\neq j}}^{N} A_{ij}, \forall j$)
D_{ij}	Shortest distance between nodes i and j
$\bar{D}_j$	Average distance of node j from other nodes, $\bar{D}_j = \dfrac{\sum_{\substack{i=1 \\ i\neq j}}^{N} D_{ij}}{N-1}, \forall j$

Decision Variables

x_{ij}	An indicator (binary) variable that takes the value 1 if node i is allotted to node j that forms the influential or nucleus node for all such nodes; 0 otherwise.

If $x_{ij} = 1$, then such nodes are associated with the same cluster in which node j acts as the influential or nucleus node. Also, x_{jj} takes the value 1 if node j acts as the influential node for its cluster; 0 otherwise.

We consider the sum of adjacencies and the sum of distances of nodes associated with the influential node j to characterize cohesiveness and compactness so that the choice of an influential node and the associated nodes lead to the best possible community formation. The objective function (Equation 1) aims at minimizing the sum of (shortest) distances of all associated nodes from their respective influential nodes so that the maximum cluster density can be achieved.

ACCEPTED MANUSCRIPT

$$\text{Minimize} \quad Z = \sum_{i=1}^{N} \sum_{\substack{j=1 \\ i \neq j}}^{N} D_{ij} \times x_{ij} \tag{1}$$

For a network with C communities, it is necessary to identify an influential node associated with each community. This is ensured using Constraint (2), where the number of influential nodes established is exactly equal to the number of clusters to be formed (C). It is to be noted that the binary variable x_{jj} takes the value 1 if node j acts as an influential member.

$$\sum_{j=1}^{N} x_{jj} = C \tag{2}$$

To ensure a non-overlapping community structure, a node cannot be associated with more than one community. In other words, a member in the community (say, node i), should be associated with only one influential member, namely member j. This is achieved using Constraint (3), where a given node is assigned to exactly one nucleus node (or a community).

$$\sum_{j=1}^{N} x_{ij} = 1 \qquad \forall i \tag{3}$$

Likewise, when node j does not act as an influential node by itself (i.e., $x_{jj} = 0$), then node i cannot be associated with node j (i.e., $x_{ij} = 0$). However, if node j acts as an influential node (i.e., $x_{jj} = 1$), then node i may or may not be associated with node j (i.e., x_{ij} can be 0 or 1). This is ensured using Constraint (4), where a given node can be assigned to (i.e., associated with) a node only when the latter acts as a influential node.

$$x_{ij} \leq x_{jj} \qquad \forall i, j \tag{4}$$

While determining the set of nodes to be associated with a given influential node j, we ensure that the sum of adjacencies of these nodes associated with the influential node j leads to the resultant cluster having a high level of cohesiveness in comparison to all adjacencies associated with this influential node j. Constraint (5) achieves this consideration by ensuring that the product of the number of clusters and the sum of adjacencies of nodes associated with the influential node j is greater than or equal to all adjacencies with respect to node j. It is evident that when node j acts as an influential member (i.e., $x_{jj} = 1$), then a large gap between the Left-Hand Side (LHS) and the Right-Hand Side (RHS) in Constraint (5) indicates a high level of community cohesiveness (also see Section 5.2 for a related discussion on Constraint (5) through

a numerical illustration).

$$C \times \Big(\sum_{\substack{i=1 \\ i \neq j}}^{N} A_{ij} \times x_{ij} \Big) \geq K_j - (N+1)(1 - x_{jj}) \qquad \forall j \qquad (5)$$

As a measure of cluster density, we consider the sum of (shortest) distances of nodes associated with the influential node j. For arriving at the best possible allocation of nodes to the influential node j, we state in Constraint (6) that the product of the mean distance of nodes from a given influential node j and the number of nodes associated with node j is greater than or equal to the sum of the shortest distances of all nodes associated with the influential node j. This ensures that the resultant cluster density is quite high, leading to a highly compact community. Therefore, when node j acts as an influential node, a large gap between the LHS and the RHS in Constraint (6) indicates a high level of community compactness (also see Section 5.2 for a related discussion using a numerical illustration).

$$\bar{D}_j \times \sum_{\substack{i=1 \\ i \neq j}}^{N} x_{ij} \geq \sum_{\substack{i=1 \\ i \neq j}}^{N} \Big(D_{ij} \times x_{ij} \Big) - \Big(\sum_{\substack{i=1 \\ i \neq j}}^{N} D_{ij} \Big)(1 - x_{jj}) \qquad \forall j \qquad (6)$$

A node i is either associated with an influential node j ($x_{ij} = 1$) or does not belong to node j ($x_{ij} = 0$). In other words, the decision variable x_{ij} takes only one of the two possible values (0 or 1), and this binary restriction is assured using Constraint (7).

$$x_{ij} \in \{0, 1\} \qquad \forall i, j \qquad (7)$$

The impact of the objective function and constraints on the optimal solution is numerically illustrated in Section 5.2.

4.1. Evaluating the Community Structure

It is necessary to assess the strength of the community structure established from the optimization model using one or more performance metrics. For networks whose ground-truth (i.e., network whose community structure is well-established in reality or recognized ahead of time) is known, the communities obtained from our model can be evaluated using Normalized Mutual Information (NMI) - a popular metric that is commonly used to measure the similarity between community structures of two networks (Danon et al., 2005). A community structure for a network with N vertices obtained from the optimization model (O) and the ground-truth (T) can have C_O and C_T communities, respectively. Further, each community in the proposed

ACCEPTED MANUSCRIPT

model and ground-truth network has N_l^O $\{l = 1, 2, .., C_O\}$ and $N_{l'}^G$ $\{l' = 1, 2, .., C_G\}$ vertices, respectively. If $\bar{N}_{ll'}$ indicates the vertices that are common between community l of the established network and community l' of the ground-truth network, then the NMI is given by Equation (8).

$$NMI = \frac{-2\sum_{l'=1}^{C_T}\sum_{l=1}^{C_O}\bar{N}_{ll'}log\frac{\bar{N}_{ll'}N}{N_l^O N_{l'}^G}}{\sum_{l=1}^{C_O}N_l^O log\frac{N_l^O}{N} + \sum_{l'=1}^{C_G}N_{l'}^G log\frac{N_{l'}^G}{N}} \tag{8}$$

The value of NMI ranges from 0 to 1, where zero indicates no similarity and one indicates identical community structure for the two networks.

However, the ground-truth community structure of most real-life networks are not known apriori (Newman and Girvan, 2004). Therefore, in the absence of ground-truth, modularity and Silhouette are important and popular cluster measures for understanding the structural properties of the network and measuring the strength of the community partition (Chen, 2014). Modularity (Q) assesses the strength of links within the community as compared to the connections outside the community, and is given by Equation (9). It ranges from -1 to 1, where a higher value indicates a better partition. Moreover, a modularity of 0 indicates that the community structure established is indifferent to a randomly established partition.

$$Q = \frac{1}{2N}\sum_{i,j}\Big[A_{ij} - \frac{K_i K_j}{2N}\Big]\delta_{ij} \tag{9}$$

where δ_{ij} is 1 if node i and node j belong to the same cluster and 0 otherwise.

On the other hand, Silhouette coefficient (S) measures the intra-cluster similarity (cohesion) and between cluster dissimilarity (separation). It is found to be a good metric for validating community structures under different experimental settings and consistently ranked in the first group among 30 validation indices (Arbelaitz et al., 2013). The Silhouette score for each node i (S_i) in the network is given by Equation (10), where a_i is the average distance between node i and other nodes that are in the same cluster, and b_i is the average distance between node i and all other nodes in a different community. The Silhouette coefficient (or overall average Silhouette) is used to evaluate network partition and ranges from -1 to +1, where a higher value indicates good cohesion as well as separation.

$$S_i = \frac{b_i - a_i}{\max\{a_i, b_i\}} \tag{10}$$

ACCEPTED MANUSCRIPT

Closeness centrality ($CLOSE$) is a well-known powerful measure to identify the critical nodes in a network (Qiao et al., 2017). It measures the proximity of each node to all the other nodes in the network. Therefore, the $CLOSE$ of a node is the reciprocal of the sum of the shortest distance between that node and all the other nodes in the network. To identify the influential node in each community, we consider each community independently and compute the normalized closeness centrality for the nodes in that community as shown in Equation (11).

$$CLOSE_i = \frac{N-1}{\sum_{j \in c_i} D_{ij}} \tag{11}$$

where c_i is the set of nodes belonging to the same community as node i. The $CLOSE$ for each vertex can range from 0 to 1, where higher values indicates better influence within the community. Thus, for each community l, the node with the highest closeness centrality ($CLOSE_l^{MAX}$) is deemed as the influential node (N_l^{BEST}) for that community.

4.2. Solution Approach

We seek an exact method which finds the global optimal solution that minimizes the given objective function, while satisfying the constraints of the proposed optimization model. The solution space (or feasible region) for the optimization model includes a set of all possible feasible solutions (set of values for the decision variables that satisfies all the constraints). Since the decision variables in the proposed model are all binary variables, the search space is a discrete set of feasible solutions and not a convex set. As a result, the solution space typically increases exponentially with the problem size. In other words, if there are 'n' binary variables in an optimization model, then there are 2^n possible values of the decision variables. Thus, an exhaustive enumeration of all the solutions in the search space is not efficient. There are numerous ways to efficiently solve the discrete optimization problem such as the branch-and-bound (B&B) method (Lawler and Wood, 1966). These well-established approaches efficiently find an optimal solution by progressively breaking the original problem into sub-problems and adopting different pruning methods to eliminate further division of certain sub-problems (Lawler and Wood, 1966). The pseudo-code for the B&B method with respect to our problem is given in Appendix A as Algorithm 1.

In this research, we are interested in finding multiple (or alternate) optimal solutions (different community structures that result in the same objective function) if they exist for a given network. However, the traditional branch-and-bound method may not be the most efficient approach as its pruning strategy may restrict it from exploring the alternate optima. Therefore, to overcome this drawback, we use the One-Tree algorithm proposed by Danna et al. (2007).

It is an adaptation of the branch-and-bound method and uses a two-phase approach to find the alternate optima. The first phase is similar to the traditional branch-and-bound procedure, but the sub-problems are stored to be used in the second phase instead of being discarded. During the second phase, the sub-problems from the first-phase are further explored to find alternate optima. The pseudo-code for the first and second phase of the one-tree algorithm is given in Appendix A as Algorithms 2 and 3, respectively.

A step-by-step procedure to obtain the nodes and corresponding influential member belonging to a community using the proposed model is summarized as follows.

Step 1: Specify the network parameters as inputs to the optimization model: Given a network graph G, determine the number of nodes (N), adjacency list for all the nodes ($A_{ij}, \forall\, i, j$), shortest distance between any two nodes ($D_{ij}, \forall\, i, j$), average distance of node j from other nodes ($\bar{D}_j, \forall\, j$), and degree of vertex j ($K_j, \forall\, j$). In addition, specify the number of communities (C) to establish.

Step 2: Obtain the output (community structure of network G and influential member associated with each community): Solve the proposed mathematical model using the first phase of the one-tree algorithm (i.e., Algorithm 2 in Appendix A) to determine the optimal objective function value Z^* and community structure ($x_{ij}, \forall\, i, j$). Note that the variable x_{jj} is 1 only if the one-tree algorithm identifies node j as an influential member in a community. Further, the format of the community structure is depicted by the value of the decision variable in the optimal solution, x_{ij}. A node i in graph G belongs to a community in which node j is the influential member (i.e., $x_{ij} = 1$). Thus, all the nodes associated with an influential member j form a community. Further, the total number of nodes in a community with influential member j is given by $\sum_i x_{ij}$.

Step 3: Determine up to m alternate optimal solutions: For the same the optimal objective function value Z^*, it may be possible that there exists a community structure that is different compared to an already identified structure. In other words, if Z^* is the optimal objective function and $x_{ij}^{(1)},\ \forall i, j$, is the corresponding decision variables (or community structure) obtained, then another solution $x_{ij}^{(2)},\ \forall i, j$, is considered to be an alternate optima only if it achieves the same objective function value Z^* and at least one different assignment in the community structure or decision variables, $\sum_j \sum_i |x_{ij}^{(2)} - x_{ij}^{(1)}| \geq 1$. It is important to identify the alternative community structures (or alternate optima) because each community structure might provide a different value of modularity or Silhouette (discussed in Section 4.1) even though they achieve same objective function

Z^*. Therefore, using the second phase of the one-tree algorithm (i.e., Algorithm 3 in Appendix A), we obtain up to m different alternative community structures that result in the same objective function value Z^*.

Step 4: Evaluate the quality of $m + 1$ community partitions: Using the one-tree algorithm, we obtain up to $m + 1$ solutions (Steps 2 and 3) which have the same objective function Z^* but different community partitions. For each of the $m + 1$ community structure, compute the modularity, Silhouette coefficient and closeness centrality using Equations (9), (10), and (11), respectively. Besides, if the ground-truth partition of the network G is known, then estimate the NMI for all the $m + 1$ solutions using Equation (8).

Step 5: Choose best community partitions: Suppose $x^1, x^2, ..., x^{m+1}$ denote the vector of $m + 1$ different community partitions. Further, if $Q^1, Q^2, ..., Q^{m+1}$ and $S^1, S^2, ..., S^{m+1}$ represent the modularity and Silhouette value for the $m + 1$ community structures, respectively. Then, it is possible to have up to two best partitions: a modularity maximizing community structure (x^{Q^*}) as shown in Equation (12) and a Silhouette maximizing community structure (x^{S^*}) as shown in Equation (15). The best modularity value (Q^*) and corresponding Silhouette coefficient (S^{Q^*}) of the modularity maximizing partition is given by Equations (13) and (14), respectively. Similarly, the best Silhouette coefficient (S^*) and the modularity of the Silhouette maximizing partition (Q^{S^*}) is given by Equations (16) and 17, respectively. Note that it is possible to have the same community partition that achieves the best modularity as well as best Silhouette coefficient (i.e., $x^{Q^*} = x^{S^*}$).

$$x^{Q^*} = x^{\text{argmax}(Q^1, Q^2, ..., Q^{m+1})} \tag{12}$$

$$Q^* = \max(Q^1, Q^2, ..., Q^{m+1}) \tag{13}$$

$$S^{Q^*} = S^{\text{argmax}(Q^1, Q^2, ..., Q^{m+1})} \tag{14}$$

$$x^{S^*} = x^{\text{argmax}(S^1, S^2, ..., S^{m+1})} \tag{15}$$

$$S^* = \max(S^1, S^2, ..., S^{m+1}) \tag{16}$$

$$Q^{S^*} = Q^{\text{argmax}(S^1, S^2, ..., S^{m+1})} \tag{17}$$

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

5. Computational Experiments: Results and Discussion

In this section, we study the effectiveness of the proposed model on several standard real-life network dataset adopted in the literature. We validate the proposed model using a network for which the actual community partition and influential node is known apriori. Besides, we compare the performance (modularity and Silhouette values) of our proposed optimization model with existing benchmark optimization model (see Appendix B) and well-performing heuristic approaches for community detection. The influential nodes, which are identified only by the proposed model, is evaluated using closeness centrality. Finally, we discuss the computational complexity of the proposed mixed integer programming model and compare it to the benchmark optimization model as they both use the same approach for solving the problem. The optimization models (proposed and benchmark) have been developed using General Algebraic Modeling System (GAMS 24.5.6) and solved using CPLEX 12.8 optimizer (Brooke et al., 2003), while the heuristic approaches are developed and solved using the R igraph library (Csardi and Nepusz, 2006). The computational study was conducted on a computer with Intel Core i7 4.20 GHz processor, 64-bit Windows 10 operating system and 64 GB RAM.

5.1. Description of Dataset Used for Experimentation

Our model is applied to the following real-life networks - karate club network (Zachary, 1971), dolphin social network (Lusseau et al., 2003), US political books network, American college football (Girvan and Newman, 2002) and human gene co-expression network (Pirim et al., 2018). Table 1 summarizes the characteristics of the network considered in this research.

Table 1: Description of dataset used for experimentation

Data set	**Vertices**	**Edges**
Karate	34	78
Dolphins	62	162
Books	105	441
Football	115	613
Human	349	1418
Texas Power Grid	1500	3626

The social relationships amongst the 34 members in a karate club at a US university were studied by Zachary (1971). The members are represented as nodes and the 78 edges represent the friendship between them. Due to conflict of opinion and leadership issues between two members (node 1 and node 34), the entire club was divided into two groups over a period

ACCEPTED MANUSCRIPT

of time. Lusseau et al. (2003) constructed the dolphin network by observing 62 bottlenose dolphins living in Doubtful Sound, New Zealand over seven years. The dolphins in the networks are represented as vertices and the communication-based relationship between the dolphins is indicated using 162 edges. The political books network is compiled based on the purchase history of books on American politics from an online retailer, amazon.com. The 105 nodes in the network represent the political books and 441 edges are used to link the books that are frequently purchased together. Most of the books in this network are categorized as conservative or liberal, while a very small proportion belongs to neither of these categories. The football network represents the Division I football game schedule in the year 2000 between the American college teams (Girvan and Newman, 2002). The 115 nodes in the network denote the college teams and 613 edges link the teams that have played each other during the regular season. The teams are divided into 12 conferences (or communities) with 8-12 teams in each conference. Typically, each team plays more intra-conference matches than inter-conference matches. The human dataset is a network of gene co-expression with 332 nodes and 1418 edges (Pirim et al., 2018). Each node represents a gene and an edge indicates a significant co-expression between two nodes. The Texas power grid represents a large high-voltage electricity distribution network with 1500 power system buses (nodes) and 3,626 transmission lines or edges (ICSEG, 2016). The partition of the nodes in a network into clusters and the influential nodes in each cluster is determined by solving the proposed optimization model to optimality.

5.2. Validation of the Proposed Community Detection Algorithm

We validate the proposed model by testing it on the karate club network that is widely used in the literature (e.g., Girvan and Newman, 2002; Raghavan et al., 2007; Pirim et al., 2018). Since the actual community memberships are known for the karate club data, we explicitly verify the ability of our proposed optimization model to accurately detect the two communities. Table 2 shows the community structure obtained using our algorithm, and compares it to the ground-truth partition.

The identified cluster membership is identical to the actual membership (or true solution) (Zachary, 1977). In other words, our proposed model achieved 100% NMI (proportion of the nodes that reflect the true community memberships). Since we are trying to establish two communities and identify their respective influential node, the model sets two self-assignments ($x_{1,1}$ and $x_{34,34}$ as shown in Table 2) to one due to the restriction imposed by Constraint (2). The node that is associated with itself is deemed as the influential node. Thus, the optimization model has identified node 1 and node 34 as the most influential nodes in their respective communities, which represent the two key members who had conflicts and were

ACCEPTED MANUSCRIPT

responsible for the fission in the group. Further, it can be observed from Table 2 that a vertex is always associated with only one influential member (either 1 or 34) due to Constraint (3). For example, node 2 is associated only with member 1 (i.e., $x_{2,1} = 1$) and does not belong to the other influential member ($x_{2,34} = 0$). Likewise, due to constraint (4), node i never associated with a node that is not influential. As shown in Table 2, node 2 is associated only with influential node 1 (i.e., $x_{2,1} = 1$) and is not associated with other nodes which are not influential (i.e., $x_{2,2} = x_{2,3} = x_{2,4} = ... = x_{2,33} = 0$).

Table 2: Comparison of Community Structure corresponding to Ground-Truth and Proposed Model for Zachary's Karate Club Network

Vertex Number	**Associated Influential Node**	**Binary Variable Assigned with Value 1***	**Community ID based on proposed model**	**Ground-Truth Community**
1	1	$x_{1,1}$	1	1
2	1	$x_{2,1}$	1	1
3	1	$x_{3,1}$	1	1
4	1	$x_{4,1}$	1	1
5	1	$x_{5,1}$	1	1
6	1	$x_{6,1}$	1	1
7	1	$x_{7,1}$	1	1
8	1	$x_{8,1}$	1	1
11	1	$x_{11,1}$	1	1
12	1	$x_{12,1}$	1	1
13	1	$x_{13,1}$	1	1
14	1	$x_{14,1}$	1	1
17	1	$x_{17,1}$	1	1
18	1	$x_{18,1}$	1	1
20	1	$x_{20,1}$	1	1
22	1	$x_{22,1}$	1	1
9	34	$x_{9,34}$	2	2
10	34	$x_{10,34}$	2	2
15	34	$x_{15,34}$	2	2
16	34	$x_{16,34}$	2	2
19	34	$x_{19,34}$	2	2
21	34	$x_{21,34}$	2	2
23	34	$x_{23,34}$	2	2
24	34	$x_{24,34}$	2	2
25	34	$x_{25,34}$	2	2
26	34	$x_{26,34}$	2	2
27	34	$x_{27,34}$	2	2
28	34	$x_{28,34}$	2	2
29	34	$x_{29,34}$	2	2
30	34	$x_{30,34}$	2	2
31	34	$x_{31,34}$	2	2
32	34	$x_{32,34}$	2	2
33	34	$x_{33,34}$	2	2
34	34	$x_{34,34}$	2	2

* Note: All other binary variables (i.e., all other x_{ij}s) are 0.

Recalling our discussion on the proposed model in Section 4, we now present how the model attempts to achieve improved community cohesiveness and compactness through Constraints (5) and (6). From the solution given by our model to the karate club problem, we find (as an example) that node 20 is associated with the influential node 1, and node 31 is associated with the influential node 34. This association leads to community cohesiveness and compactness.

ACCEPTED MANUSCRIPT

We have stated in Section 4 that a large gap between the LHS and the RHS in Constraints (5) and (6) is desirable to achieve higher cohesiveness and compactness. To demonstrate this, we have re-assigned node 20 to the influential node 34, and node 31 to the influential node 1. The objective function value along with the difference between LHS and RHS for Constraints (5) and (6), before and after re-assignment (with all other associations or assignments of other nodes to their respective influential nodes remaining the same) is shown in Table 3.

Table 3: Numerical Illustration of Cohesiveness and Compactness achieved by the Proposed Model

Parameter	**Optimal Solution**	**Solution after re-assignment**
Objective Function	35	36
For Influential Node 1		
• LHS - RHS in Constraint (5)	30 - 16 = 14	30 - 16 = 14
• LHS - RHS in Constraint (6)	31.64 - 21 = 10.64	33.39 - 23 = 10.39
For Influential Node 34		
• LHS - RHS in Constraint (5)	28 - 17 = 11	26 - 17 = 9
• LHS - RHS in Constraint (6)	25.45 - 14 = 11.45	23.64 - 13 = 10.64

It is evident that by virtue of Constraints (5) and (6), the value of the objective function leads to a better cluster density (indicated by a lower value of 35 given by our optimization model in comparison to the value of 36 after re-assignment). In addition, our model (through constraints (5) and (6)) achieves a cohesive and compact community formation as well (indicated by larger gaps between the LHS and the RHS of Constraints (5) and (6) in our optimized model as against those in the sub-optimal model with only one re-assignment considered for the sake of illustration).

5.3. Model Performance using Real-Life Social Networks

5.3.1. Comparison of Proposed Model's Performance to Existing Benchmark Optimization Model

While the objective of the proposed model is to detect communities and identify influential nodes, the ground truth (or the actual number of communities and membership associated with each node) is not explicit in most real-life situations. Therefore, as discussed in Section 4.1, we use two different internal validation indices, namely, modularity and Silhouette coefficient, to evaluate the clusters formed by the proposed model and compare it to the existing benchmark optimization model by Pirim et al. (2018). The cluster membership obtained from the mathematical model is used to calculate the modularity and Silhouette coefficient using the R igraph library (Csardi and Nepusz, 2006) and Cluster package (Maechler et al., 2006), respectively.

The number of communities to be established (C) must be specified as an input (or parameter) for the proposed and benchmark optimization models. We used the same number of clusters adopted by Pirim et al. (2018) to ensure an fair comparison of our model with

the benchmark model. However, we observe from the execution of the proposed and existing optimization models that there could exist alternate optimal solutions. Since the objective function does not explicitly maximize the internal validation indices, the alternate optima can provide different Silhouette or modularity values for the same objective function value. Hence, given the number of communities, we enumerate up to 20 alternate optima and choose the best solution with respect to modularity and observe its corresponding Silhouette value (refer to Table 4); thereafter we choose the best solution with respect to Silhouette value and observe its corresponding modularity (Table 5). This procedure is done for our model and the benchmark model. Due to this approach, we are able to obtain better modularity and Silhouette values for the benchmark model as opposed to the original values reported in Pirim et al. (2018), and we have used these improved values in our performance evaluation.

Table 4 presents the best modularity values for the proposed model (Q_P^*) and the benchmark model (Q_B^*) along with the corresponding Silhouette values for the proposed (S_P) and benchmark (S_B) models. Also, the percentage difference in modularity (Q_{diff}) and Silhouette values (S_{diff}) between the proposed and benchmark models are computed as shown in Equations (18) and (19), respectively. A positive difference indicates the improvement achieved by the proposed model over the benchmark model.

$$Q_{diff} = \left(\frac{Q_P - Q_B}{Q_B}\right) \times 100 \tag{18}$$

$$S_{diff} = \left(\frac{S_P - S_B}{S_B}\right) \times 100 \tag{19}$$

Our model, without being designed for optimizing modularity, provides solutions with better modularity values compared to the benchmark model. The effectiveness of the proposed optimization model can be observed from Table 4 as it performs better with respect to both the quality indices for a majority of the dataset. For one of the remaining cases, the modularity value produced by our model is only marginally less, and hence we are able to provide competitive results. The best modularity and corresponding Silhouette values improved by an average of 38% and 18%, respectively. Moreover, for the Zachary's karate club data with two clusters, the benchmark model (Pirim et al., 2018) incorrectly places nodes 1 and 34 (denoting the two key members who had conflicting issues and separated during the fission) within the same cluster, while the proposed model accurately identifies the central nodes and places them in different communities (Figure 2).

ACCEPTED MANUSCRIPT

Table 4: Best modularity value and corresponding Silhouette index for proposed and benchmark models for different real-life dataset

Dataset	Number of Clusters	Best Modularity Values			Corresponding Silhouette Index		
		Q_B^*	Q_P^*	Q_{diff}	S_B	S_P	S_{diff}
Karate	2	0.133	0.371	178.95	0.251	0.347	38.25
Karate	3	0.376	0.387	2.93	0.254	0.265	4.33
Dolphin	2	0.359	0.390	8.64	0.200	0.436	118.00
Dolphin	3	0.384	0.446	16.15	0.302	0.279	-7.62
Books	2	0.443	0.432	-2.48	0.392	0.402	2.55
Books	3	0.459	0.479	4.36	0.289	0.296	2.42
Football	6	–†	0.362	NA	0.141$^{\#\dagger}$	0.130	-7.80
Football	12	–†	0.428	NA	–†	0.213	NA
Human	3	0.271$^{\#\dagger}$	0.438	61.62	0.523$^{\#\dagger}$	0.491	-6.12
Texas Power Grid	20	–†	0.921	NA	–†	0.370	NA

*Denotes the best value among the 20 alternate optima evaluated.

†CPLEX solver did not achieve optimality even after six hours of execution on our system.

$^{\#}$Denotes the value reported in Pirim et al. (2018).

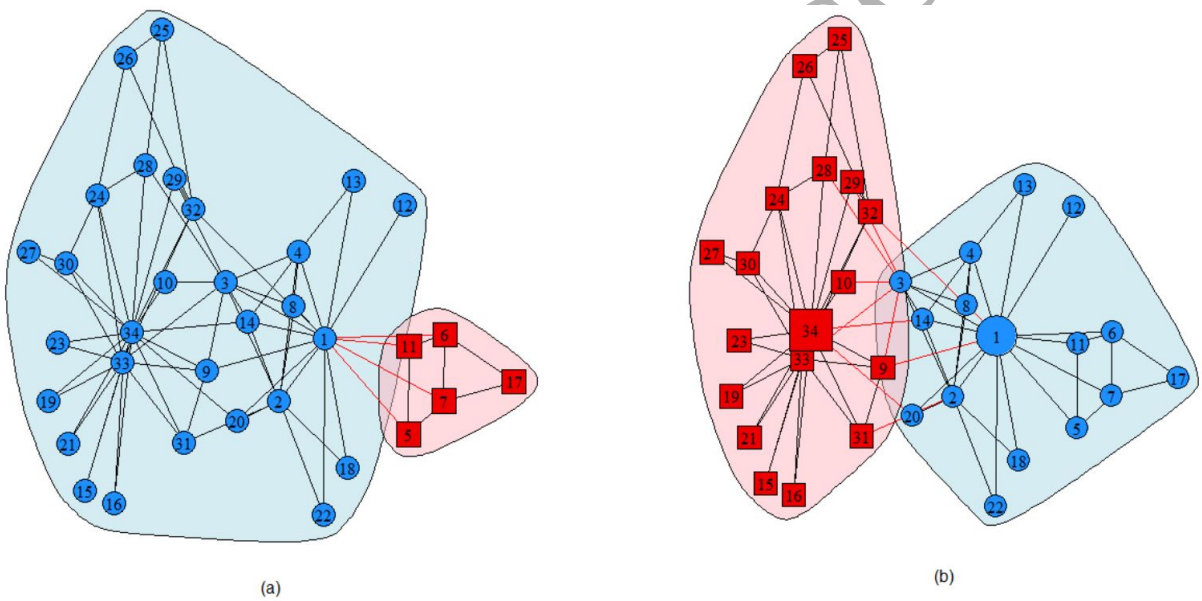


Figure 2: Two community partition of karate club network by (a) benchmark model and (b) proposed model (influential node denoted by bigger size)

Besides, we are unable to solve the benchmark model to optimality within a reasonable amount of time for the football network with 6 as well as 12 clusters. Therefore, the Silhouette value for the football network with 6 clusters in Table 4 is based on the results reported in Pirim et al. (2018). However, the modularity for the football network with 6 or 12 clusters, and the Silhouette value with 12 clusters using the benchmark model are not available in the literature.

While the benchmark model has the Silhouette index directly incorporated into the model's objective function, our proposed model is not aimed at explicitly optimizing the Silhouette index values. Yet our model is able to generate overall superior Silhouette coefficient values as shown in Tables 4 and 5. The best Silhouette index along with the corresponding modularity values

for the proposed and benchmark models are presented in Table 5. Consistent with the previous analysis, the best Silhouette coefficient and the corresponding modularity yielded by our model are better than those by the benchmark model. Our model is able to produce superior Silhouette coefficient values for six out of the nine cases. The proposed model resulted in an average improvement of 19% and 36% with respect to the best Silhouette values and corresponding modularity values, respectively. Moreover, it can be observed that the best Silhouette coefficient also corresponds to the best modularity value for the following instances in the proposed model - karate dataset with 2 clusters, football network with 6 clusters, Human gene co-expression graph with 3 communities and Texas Power Grid dataset with 20 communities. However, in all the other cases, an increase in the Silhouette value for the proposed model is achieved only with a slight decrease in the best modularity value.

Table 5: Best Silhouette value and corresponding modularity value for proposed and benchmark models for different real-life datasets

Dataset	**Number of Clusters**	**Best Silhouette Index**			**Corresponding Modularity Values**		
		S_B^*	S_P^*	S_{diff}	Q_B	Q_P	Q_{diff}
Karate	2	0.251	0.347	38.25	0.133	0.371	178.95
Karate	3	0.255	0.278	9.02	0.365	0.344	-5.75
Dolphin	2	0.200	0.438	119.00	0.359	0.372	3.62
Dolphin	3	0.302	0.282	-6.62	0.384	0.444	15.63
Books	2	0.392	0.406	3.57	0.443	0.426	-3.84
Books	3	0.290	0.302	4.14	0.454	0.471	3.75
Football	6	$0.141^{\#\dagger}$	0.130	-7.80	$-^{\dagger}$	0.362	NA
Football	12	$-^{\dagger}$	0.223	NA	$-^{\dagger}$	0.418	NA
Human	3	$0.523^{\#\dagger}$	0.491	-6.12	$0.271^{\#\dagger}$	0.438	61.62
Texas Power Grid	20	$-^{\dagger}$	0.370	NA	$-^{\dagger}$	0.921	NA

*Denotes the best value among the 20 alternate optima evaluated.

†CPLEX solver did not achieve optimality even after six hours of execution on our system.

#Denotes the value reported in Pirim et al. (2018).

In the case of the 3-cluster karate club network, we have identified an alternate optimal cluster partition for the benchmark model that results in higher modularity and Silhouette values as opposed to the original values (modularity: 0.335 and Silhouette: 0.222) reported in Pirim et al. (2018). Also, as illustrated in Figure 3, the proposed optimization model yields high-quality modularity and Silhouette maximizing communities that are better than the improved values of the benchmark model. Further, visualization of the community partition for the Dolphin, Books and Human networks considered in Tables 4 and 5 are presented in Appendix C.

ACCEPTED MANUSCRIPT

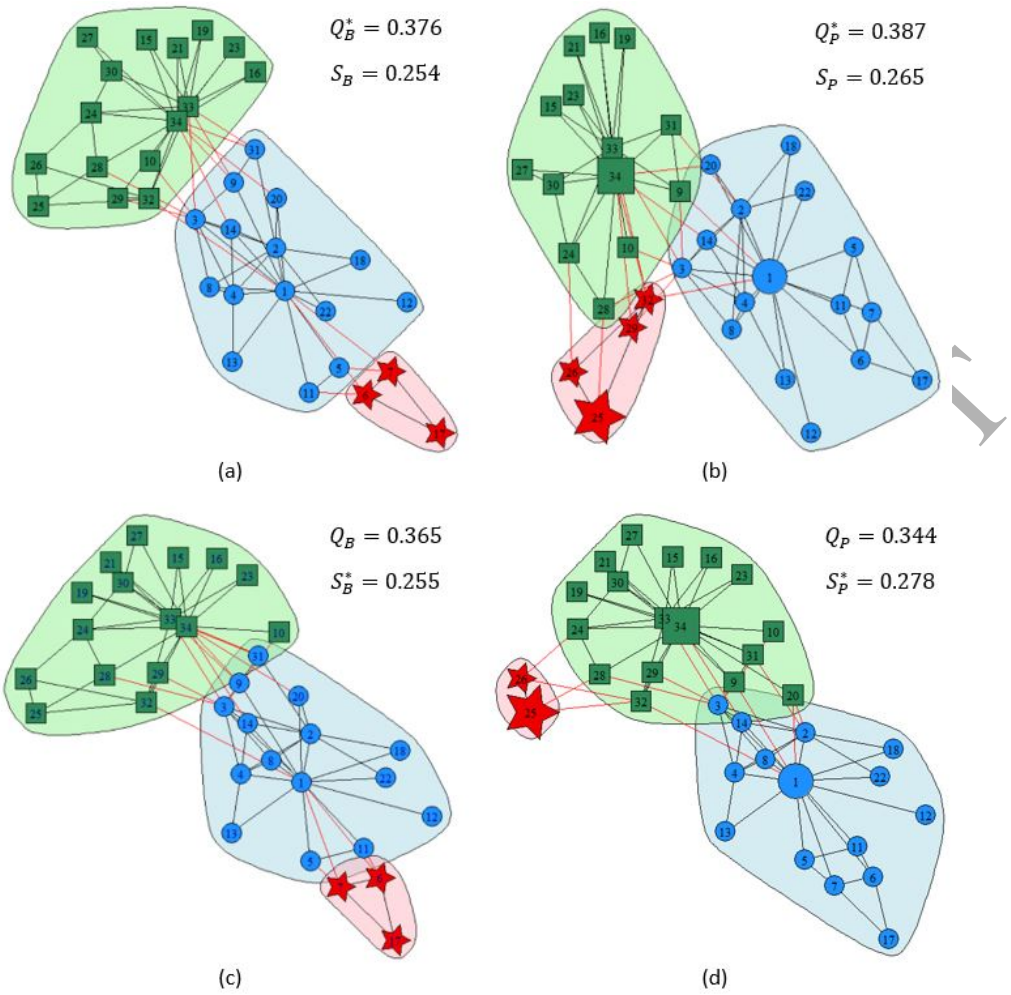


Figure 3: 3-cluster karate club network: (a) Best modularity partition of benchmark model and (b) proposed model, (c) Best Silhouette partition of benchmark and (d) proposed model (influential node identified by proposed model denoted by bigger size)

5.3.2. Comparison of Proposed Model's Performance with Existing Heuristic Algorithms

In this section, we present the performance evaluation of the proposed OR-based model with the well-performing heuristic approaches reported in the literature. Based on the comparative analysis of several heuristic algorithms by Yang et al. (2016), the edge-betweenness algorithm (Newman and Girvan, 2004), Walktrap algorithm (Pons and Latapy, 2005), and Louvain method (Blondel et al, 2008) were identified to be consistently well-performing on different datasets. Hence, these heuristic approaches are included in the present study for comparative performance evaluation. It is to be noted that these heuristic approaches require no input concerning the number of clusters or communities to be formed because they automatically evolve as these heuristics progress and terminate when the performance measure is maximized. However, in our OR-based model, we specify the number of communities as an input parameter. Therefore, to ensure an equivalent comparison, we varied the number of clusters in the proposed model from two to the maximum number of communities detected by the heuristic algorithms (in

increments of 1), and reported the best performance measure (with respect to modularity and Silhouette) achieved.

The best modularity and Silhouette achieved by the proposed model and the three heuristic approaches under consideration are presented in Tables 6 and 7, respectively. It can be observed that the optimization model proposed in this research achieves the best Silhouette values in all cases, except one. Even though our model is not designed to optimize modularity, it attains competitive values compared to the three heuristic approaches. In fact, the proposed integer programming model performs better than the heuristic approaches with respect to modularity (as well as with respect to Silhouette coefficient) in the case of medium (Human gene co-expression network with 349 nodes and 1418 edges) and large (Texas Power Grid network with 1500 nodes and 3626 edges) datasets. However, the proposed model does not achieve superior performance for the football network. As Pirim et al. (2018) noted, this may be because of the greater number of partitions present in a small network consisting of only 115 nodes.

Table 6: Comparison of best modularity values achieved by proposed optimization model and three well-performing heuristic approaches

Dataset	Best Modularity Values			
	Edge-betweenness	Louvain	Walktrap	Proposed
Karate	0.401	0.419	0.353	0.387
Dolphin	0.519	0.519	0.489	0.478
Books	0.517	0.520	0.507	0.479
Football	0.600	0.604	0.603	0.463
Human	0.658	0.656	0.638	0.661
Texas Power Grid	0.915	0.911	0.882	0.921

Table 7: Comparison of best Silhouette values achieved by proposed optimization model and three well-performing heuristic approaches

Dataset	Best Silhouette Coefficient Values			
	Edge-betweenness	Louvain	Walktrap	Proposed
Karate	0.1662	0.2195	0.1712	0.371
Dolphin	0.2876	0.234	0.2272	0.438
Books	0.2247	0.2415	0.2609	0.406
Football	0.3011	0.317	0.3128	0.224
Human	0.3019	0.3156	0.2844	0.491
Texas Power Grid	0.3298	0.3183	0.1918	0.370

Table 8 presents the number of communities detected by the heuristic approaches and compares it to the communities that yields the best modularity and Silhouette in the proposed model. Note that the heuristic approaches identify only one best partition, which is then used to determine the best modularity and Silhouette value. However, the proposed model may detect a different number of communities for the best modularity value and Silhouette coefficient.

It is seen that the best number of community partitions by the proposed mathematical programming model is always less than or equal to the number of partitions yielded by the existing heuristic approaches. This is beneficial and a key aspect of the proposed model because targeting a relatively smaller number of influential nodes is easy and quick, especially in disaster mitigation or risk management scenario, thereby making our model attractive and effective in community detection.

Table 8: Number of communities detected by heuristic approaches and proposed optimization model

Dataset	Communities leading to best performance				
	Edge-betweenness	Louvain	Walktrap	Proposed Model (Best Modularity)	Proposed Model (Best Silhouette)
Karate	5	4	5	3	2
Dolphin	5	5	4	4	2
Books	5	4	4	3	2
Football	10	10	10	9	9
Human	12	11	31	10	2
Texas Power Grid	26	30	51	20	20

5.3.3. Evaluation of Influential Node Detection by the Proposed Model

Unlike the benchmark optimization model and heuristic algorithms, the proposed model also identifies the influential node in each community. As discussed in Section 4.1, the influential nodes identified by the proposed model is evaluated using the closeness centrality. Given a community partition of a network, the closeness centrality of the nodes belonging to each community is calculated as shown in Equation (11), and is used to identify the influential node for each community. A node which has the highest closeness centrality within community l ($CLOSE_l^{MAX}$) is considered to be an influential node for that community (N_l^{BEST}). Tables 9 and 10 compare the influential node identified based on closeness centrality to the influential node detected by the proposed model for community partitions that yields the highest modularity and Silhouette values, respectively. It can be observed that the proposed model correctly identifies the influential nodes within each community.

Table 9: Evaluation of Influential Nodes Identified by the Proposed Model for Best Modularity Partition

Dataset	**Number of Clusters**	**Influential Nodes for Best Modularity Partition**	
		Identified by Proposed Model	Identified by Closeness Centrality $N_l^{BEST}(CLOSE_l^{MAX})$
Karate	3	1, 32, 34	1 (0.938), 32 (1.00), 34 (1.00)
Dolphin	4	15, 18, 31, 46	15 (0.704), 18 (0.643), 31 (0.778), 46 (0.737)
Books	3	9, 59, 85	9 (0.692), 59 (0.656), 85 (0.735)
Football	9	6, 7, 19, 52, 92, 94, 110, 111, 114	6 (0.824), 7 (0.750), 19 (1.00), 52 (1.00), 92 (1.00), 94 (1.00), 110 (1.00), 111 (1.00), 114 (0.813)
Human	10	37, 50, 57, 65, 70, 101, 196, 220, 246, 254	37 (0.797), 50 (0.52), 57 (0.525), 65 (0.566), 70 (0.652), 101 (0.55), 196 (0.643), 220 (0.563), 246 (0.517), 254 (0.597)
Texas Power Grid	20	107, 160, 192, 244, 336, 475, 599, 635, 730, 805, 958, 1033, 1040, 1098, 1187, 1221, 1279, 1355, 1400, 1415	107 (0.194), 160 (0.187), 192 (0.230), 244 (0.169), 336 (0.190), 475 (0.178), 599 (0.177), 635 (0.194), 730 (0.187), 805 (0.218), 958 (0.186), 1033 (0.167), 1040 (0.175), 1098 (0.248), 1187 (0.209), 1221 (0.180), 1279 (0.141), 1355 (0.195), 1400 (0.209), 1415 (0.242)

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

Table 10: Evaluation of Influential Nodes Identified by the Proposed Model for Best Silhouette Partition

Dataset	Number of Clusters	Influential Nodes for Best Silhouette Partition	
		Identified by Proposed Model	Identified by Closeness Centrality $N_l^{BEST}(CLOSE_l^{MAX})$
Karate	2	1, 34	1 (0.938), 34 (0.895)
Dolphin	2	15, 18	15 (0.543), 18 (0.618)
Books	2	9, 31	9 (0.676), 31 (0.611)
Football	9	6, 7, 19, 52, 92, 94, 110, 111, 114	6 (0.824), 7 (0.750), 19 (1.00), 52 (1.00), 92 (1.00), 94 (1.00), 110 (1.00), 111 (1.00), 114 (0.813)
Human	2	18, 50	18 (0.317), 50 (0.352)
Texas Power Grid	20	107, 160, 192, 244, 336, 475, 599, 635, 730, 805, 958, 1033, 1040, 1098, 1187, 1221, 1279, 1355, 1400, 1415	107 (0.194), 160 (0.187), 192 (0.230), 244 (0.169), 336 (0.190), 475 (0.178), 599 (0.177), 635 (0.194), 730 (0.187), 805 (0.218), 958 (0.186), 1033 (0.167), 1040 (0.175), 1098 (0.248), 1187 (0.209), 1221 (0.180), 1279 (0.141), 1355 (0.195), 1400 (0.209), 1415 (0.242)

5.4. Computational Complexity

In this section, we evaluate the computational complexity of the proposed integer programming model and compare it to the optimization model by Pirim et al. (2018) as both these models adopt the same methodological and solution approach. Table 11 tabulates the computational time taken by the proposed model (t_P) and compares it to the time taken by the benchmark optimization model (t_B). In addition, it also provides the number of iterations (i.e., number of sub-problems solved) taken by the benchmark (I_B) and proposed (I_P) models to find the integral optimal solution. It is evident from Table 11 that the proposed integer programming model is computationally efficient compared to the optimization model developed by Pirim et al. (2018) for all the cases. While the benchmark model could not achieve optimality for medium (Football and Human dataset) and large networks (Texas Power Grid dataset) even after six hours of execution, the proposed model is able to detect the communities and achieve optimality in a reasonable time. It is to be noted that in the proposed model, due to Constraints (2) and (3), the number of active binary variables is only N, and due to Constraint (4), we restrict the search space related to the possible allocation of a node to another node only when the latter node serves as an influential node. These aspects make our model computationally fast compared to the benchmark optimization model.

It had been shown that the problem of community detection is akin to the problem of network partitioning, which deals with the separation of a graph into a given number of groups of almost equal sizes and minimizing the number of edges between such groups. This is an NP-hard problem (Kernighan, 1970; Karger, 2000). In other words, the proposed and existing models cannot be expected to solve the community detection problem under study in polynomial time and the worst-case time complexity is exponential. As a result, when the problem size increases (i.e., number of variables and constraints), the optimization models can become intractable and difficult to solve. In our case, the size of the optimization model depends on the number of vertices in the network. The proposed model has N^2 binary decision variables and N^2+3N+1

constraints. Nevertheless, our model is very well-suited for identifying community structure and influential nodes quickly in small and medium-sized networks, especially due to the active binary variables being only N, and the search space being restricted in our model.

Table 11: Comparison of the computational time observed for benchmark and proposed models

Dataset	**Number of Clusters**	**Computational time (in seconds)**			**Number of Iterations**		
		t_B	t_P	Percentage Reduction $\left(\frac{t_B - t_P}{t_P}\right) \times 100$	I_B	I_P	Percentage Reduction $\left(\frac{I_B - I_P}{I_P}\right) \times 100$
Karate	2	0.13	0.13	0	1,169	92	1170.7
Karate	3	0.56	0.23	143.5	2,093	92	2175.0
Dolphin	2	0.39	0.38	2.6	378	363	4.1
Dolphin	3	2.95	0.44	570.5	4,305	808	432.8
Books	2	0.52	0.50	4.0	7,021	2,274	208.8
Books	3	11.60	1.16	900	218,642	4,845	4412.7
Football	6	–†	0.90	NA	–†	3,281	NA
Football	12	–†	4.76	NA	–†	12,796	NA
Human	3	–†	11.24	NA	–†	0.13	NA
Texas Power Grid	20	–†	2656.56	NA	–†	283,222	NA

†CPLEX solver did not achieve optimality even after six hours of execution on our system.

Further, identifying community structures that maximize the modularity is an NP-hard problem (Xu et al., 2010). However, the proposed model performs better than the existing optimization model with respect to modularity and Silhouette index in most cases, even though the objective function does not explicitly aim to maximize them. This suggests that the proposed model could be a valuable tool and enable the development of good heuristic algorithms for large-sized networks. Further, the proposed model can be attempted to solve large-sized problems by adopting a two-stage approach (e.g., see Xu et al., 2010), where the model is solved for a fixed time duration to obtain a good feasible solution (i.e., an acceptable community partition), and then using a heuristic to further enhance the community structure.

6. Discussion on Potential Real-life Applications of Our Model

One of the key benefits of the proposed model is its ability to find the central or influential nodes, coupled with detecting compact communities. This makes our model very useful for communication and information spreading, especially during an emergency. More specifically, our model provides the following advantages when applied to social networks concerning information distribution and call for response.

6.1. Faster Coverage

Compactness of clusters is achieved due to the formulation of the objective that minimizes the sum of the distances between the nodes in the cluster and the corresponding nucleus node.

ACCEPTED MANUSCRIPT

Compact communities mean well-knit people; hence information travels faster within such communities. In a study conducted by Mulyasari and Shaw (2014), it was shown that risk communication and the response to a disaster in Bandung, Indonesia, were made more efficient through community-based societal organizations. Here, well-knit communities like the women's groups, youth groups, and faith-based groups are employed for risk communication. Our model will be able to detect such communities and the key contact persons in them, and these people can be used to diffuse information faster within their respective communities.

6.2. Better Call for Response

Our model is unique due to its ability to find the influential nodes along with finding compact communities. This node or the person has the maximum relationship with the rest of the people within the same community, thereby making our central node a trustworthy and influential person within the group. The study done by Gultom (2016) has information regarding the importance of having sources that recipients find credible during disasters. Moreover, the influential persons found in various compact communities detected by our model can aid in gaining large-scale consensus, which has varied applications such as controlling community riots or obtaining support from a community for a relief activity or for a novel cause/movement.

Our model's ability to detect compact communities and its ability to find central nodes from a given social network thus help us with faster information coverage and a better call for response making it particularly useful in areas like disaster management, risk communication and organizational effectiveness in terms of influencing workgroups.

7. Conclusions

In this study, we have proposed a new integer linear programming model for clustering social networks by detecting the natural partitions available in the data corresponding to the network. Though our primary objective is not to optimize the modularity nor the Silhouette index, our model produces overall superior values for these evaluation metrics when compared to one of the best performing optimization models in the literature. Besides, in many cases, our model yields superior Silhouette coefficients and competitive modularity values compared to the three well-performing existing heuristic approaches. Moreover, our model also detects the influential nodes during the clustering process, which has applications in a variety of fields including disaster management, risk communication, organizational effectiveness, and communication effectiveness. Scope for future works includes the development of metaheuristic and heuristic methods, possibly derived from our proposed novel integer linear programming model to solve large-scale problems.

ACCEPTED MANUSCRIPT

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors are grateful to the anonymous reviewers for their valuable feedback that helped us improve our work significantly. The authors are indebted to Dr. Harun Pirim for clarifying our doubts and sharing the Human network dataset. The authors would like to thank Mr. Surya Ramachandiran and Ms. Monica Devarajulu for their assistance.

References

Agarwal, G., & Kempe, D. (2008). Modularity-maximizing graph communities via mathematical programming. *European Physical Journal B*, 66(3), 409-418.

Aral, S., & Walker, D. (2012). Identifying influential and susceptible members of social networks. *Science*, 337(6092), 337-341.

Arasteh, M., & Alizadeh, S. (2019). A fast divisive community detection algorithm based on edge degree betweenness centrality. *Applied Intelligence*, 49(2), 689-702.

Arbelaitz, O., Gurrutxaga, I., Muguerza, J., Perez, J.M., & Perona, I. (2013). An extensive comparative study of cluster validity indices. *Pattern Recognition*, 46(1), 243-256.

Blondel, V. D., Guillaume, J. L., Lambiotte, R., & Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10), P10008.

Brandes, U., Delling, D., Gaertler, M., Gorke, R., Hoefer, M., & Nikoloski, Z. (2008). On modularity clustering. *IEEE Transactions on Knowledge and Data Engineering*, 20 (2), 172-188.

Brooke, A., Kendrick, D., Meeraus, A., & Raman, R. (2003). GAMS/CPLEX: A user's guide. GAMS Development Corporation, 3.

Bulu, A., Meyerhenke, H., Safro, I., Sanders, P., & Schulz, C. (2016). Recent advances in graph partitioning. *Algorithm Engineering*, pp. 117-158. Springer, Cham.

Cafieri, S., & Hansen, P. (2014). Using mathematical programming to refine heuristic solutions for network clustering. In *Models, Algorithms and Technologies for Network Analysis*, pp. 9-20. Springer, Cham.

Csardi, G., Nepusz, T. (2006). The igraph software package for complex network research. *Int. Complex Syst.*, 1695, 1-9.

ACCEPTED MANUSCRIPT

Danna, E., Fenelon, M., Gu, Z., & Wunderling, R. (2007). Generating multiple solutions for mixed integer programming problems. In *International Conference on Integer Programming and Combinatorial Optimization*, 280-294, Springer, Berlin, Heidelberg.

Danon, L., Diaz-Guilera, A., Duch, J., & Arenas, A. (2005). Comparing community structure identification. *Journal of Statistical Mechanics: Theory and Experiment*, 2005(09), P09008.

Deng, Z. H., Qiao, H. H., Song, Q., & Gao, L. (2019). A complex network community detection algorithm based on label propagation and fuzzy C-means. *Physica A: Statistical Mechanics and its Applications*, 519, 217-226.

Gaume, B. (2004). Random walks in lexical small worlds. *Information-Interaction-Intelligence International Journal*, 4(2), 39-96.

Girvan, M., & Newman, M.E.J. (2002). Community structure in social and biological networks. In *Proceedings of the National Academy of Sciences*, 99(12), 7821-7826.

Glover, F.K., & Kochenberger, G. (2006). New optimization models for data mining. *International Journal of Information Technology and Decision Making*, 5(4) , 605-609.

Gultom, D. I. (2016). Community-based disaster communication: How does it become trustworthy? *Disaster Prevention and Management*, 25(4), 478-491.

Illinois center for a smarter electric grid (ICSEG). (2016). Texas 2000, Coordinated Science Laboratory, icseg.iti.illinois.edu/synthetic-power-cases/texas2000-june2016/ (accessed: April 10, 2019).

James, G., Witten, D., Hastie, T., & Tisbshirani, R. (2013). An introduction to statistical learning with applications in R. New York: Springer, 385-410.

Ji, P., Zhang, S. and Zhou, Z., (2019). A decomposition-based ant colony optimization algorithm for the multi-objective community detection. *Journal of Ambient Intelligence and Humanized Computing*, 1-16.

Karger, D. R. (2000). Minimum cuts in near-linear time. *Journal of the ACM (JACM)*, 47(1), 46-76.

Karmarkar, N. (1984). A new polynomial-time algorithm for linear programming. In *Proceedings of the Sixteenth Annual ACM Symposium on Theory of Computing*, pp. 302-311.

Kernighan, B. W., & Lin, S. (1970). An efficient heuristic procedure for partitioning graphs. *Bell System Technical Journal*, 49(2), 291-307.

Larose, D.T.,& Larose, C.D. (2015). Data mining and predictive analytics. John Wiley and Sons Inc., Hoboken, New Jersey, 764-882.

Lawler, E. L., & Wood, D. E. (1966). Branch-and-bound methods: A survey. *Operations Research*, 14(4), 699-719.

ACCEPTED MANUSCRIPT

Lin, C. C., Kang, J. R., & Chen, J. Y. (2015). An integer programming approach and visual analysis for detecting hierarchical community structures in social networks. *Information Sciences*, 299, 296-311.

Lusseau, D., Schneider, K., Boisseau, O., Haase, P., Slooten, E., & Dawson, S. (2003). The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations. *Behavioral Ecology and Sociobiology*, 54, 396.

Maechler, M., Rousseeuw, P., Struyf, A., Hubert,M., & Hornik, K. (2017). Cluster: Cluster analysis basics and extensions, R package version 2.0.6 edn.

Martins, P. (2016). Modeling the maximum edge-weight k-plex partitioning problem. *arXiv preprint* arXiv:1612.06243.

Mehrotra, A., & Trick, M.A. (1998) Cliques and clustering: A combinatorial approach. *Operations Research Letters*, 22(1), 1-12.

Mulyasari, F., & Shaw, R. (2014) Risk communication through community-based society organizations as local response to disaster in Bandung, Indonesia. *Benchmarking*, 231-250.

Newman, M. E.J., & Girvan, M. (2004). Finding and evaluating community structure in networks. *Physical Review E*, 69, 026113.

Pirim, H., Eksioglu, B., & Glover, F. (2018). A novel mixed integer linear programming model for clustering relational networks. *Journal of Optimization Theory and Applications*, 176(2), 492-508.

Pons, P., & Latapy, M. (2005). Computing communities in large networks using random walks. In *International Symposium on Computer and Information Sciences*, pp. 284-293. Springer, Berlin, Heidelberg.

Radicchi, F., Castellano, C., Cecconi, F., Loreto, V., & Parisi, D. (2004). Defining and identifying communities in networks. In *Proceedings of the National Academy of Sciences*, 101(9), 2658-2663.

Raghavan, U. N., Albert, R., & Kumara, S. (2007). Near linear time algorithm to detect community structures in large-scale networks. *Physical Review E*, 76, 036106.

Rao, M.R. (1971). Cluster analysis and mathematical programming. *Journal of the American Statistical Association*, 66(335), 622-626.

Tan, M.P., Broach, J.R., & Floudas, C.A. (2007). A novel clustering approach and prediction of optimal number of clusters: Global optimum search with enhanced positioning. *Journal of Global Optimization*, 39 (3), 323-346.

Taylor, B. (2009). Integer programming: The branch and bound method. Introduction to Management Science. New Jersey: Prentice Hall.

Tulu, M. M., Hou, R., & Younas, T. (2018). Identifying influential nodes based on community structure to speed up the dissemination of information in complex network. *IEEE Access*, 6, 7390-7401.

Saglam, B., Salman, F.S., Sayin, S., & Turkay, M. (2006). A mixed-integer programming approach to the clustering problem with an application in customer segmentation. *European Journal of Operational Research*, 173(3) , 866 - 879.

Xu, G., Tsoka, S., & Papageorgiou, L. G. (2007). Finding community structures in complex networks using mixed integer optimisation. *The European Physical Journal B*, 60(2), 231-239.

Xu, G., Bennett, L., Papageorgiou, L. G., & Tsoka, S. (2010). Module detection in complex networks using integer optimisation. *Algorithms for Molecular Biology*, 5(1), 36.

Xie, J., Kelley, S., & Szymanski, B. K. (2013). Overlapping community detection in networks: The state-of-the-art and comparative study. *ACM Computing Surveys*, 45(4), 43.

Yen, L., Vanvyve, D., Wouters, F., Fouss, F., Verleysen, M., & Saerens, M. (2005). Clustering using a random walk based distance measure. In *Proceedings of the Thirteenth Symposium on Artificial Neural Networks*, 317-324.

Yang, Z., Algesheimer, R., & Tessone, C. J. (2016). A comparative analysis of community detection algorithms on artificial networks. *Scientific Reports*, 6, 30750.

Zachary, W. (1977). An information flow model for conflict and fission in small groups. *Journal of Anthropological Research*, 33,452.

Pan, G., Zhang, W., Wu, Z., & Li, S. (2014). Online community detection for large complex networks. *PloS One*, 9(7), e102799.

Zhang, X. K., Ren, J., Song, C., Jia, J., & Zhang, Q. (2017). Label propagation algorithm for community detection based on node importance and label influence. *Physics Letters A*, 381(33), 2691-2698.

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

Appendix A - Algorithms for solving 0-1 Integer Programs

Branch-and-Bound Algorithm

The Branch-and-Bound (B&B) algorithm is used to find the optimal solution of an integer programming model. It starts the solution procedure by considering a sub-problem, which is a relaxed version of the proposed optimization model (R) that ignores the binary integer restrictions of the decision variables (i.e., Objective Function (1) is subject to Constraints (2) - (6)). The relaxed version is therefore a linear program, which can be solved very efficiently and quickly since it has polynomial-time average case complexity (Karmarkar, 1984). Upon solving the relaxed problem, the solution procedure creates two new sub-problems by adding a constraint to the existing relaxed problem. This procedure is repeated iteratively until the optimal solution to the integer program model is found. The step by step procedure of B&B algorithm is illustrated in Taylor (2009), and is summarized here with respect to the problem under study for the sake of completeness.

Algorithm 1 Branch and Bound Algorithm for 0-1 Integer Program

Initialize the proposed optimization model without binary restrictions (P^R) as a sub-problem, $P^S \leftarrow R$
Set a very high current best objective function value, $Z^* \leftarrow +\infty$
while $N^S \neq \emptyset$ **do**
    Choose a sub-problem $p_k \in P^S$
    Obtain the optimal solution, s_k and objective function value Z_k by solving p_k as a linear program
    if $Z_k > Z^*$ **then**
        Remove p_k from the set of sub-problems, $P^S \leftarrow P^S \setminus p_k$
    else
        if s_k satisfies binary restrictions **then**
            Updated current best solution and objective function: $s^* \leftarrow s_k$ and $Z^* \leftarrow Z_k$
        else
            Choose a variable x_ij in p_k which does not satisfy binary restrictions
            Create two sub-problems (p_k^1 and p_k^2) by adding an additional constraint which forces $x_{ij} = 0$ in p_k^1 and $x_{ij} = 1$ in p_k^2 (i.e., $p_k^1 = p_k \cup \{x_{ij} = 0\}$ and $n_k^2 = n_k \cup \{x_{ij} = 1\}$)
            Update the list of sub-problems, $P^S \leftarrow P^S \cup \{p_k^1, p_k^2\} \setminus \{p_k\}$
        end if
    end if
end while
return optimal solution s^* and optimal objective function Z^*

Note: The procedure to solve the linear program (or relaxed model) is not presented here as it is well-established. Interested readers are referred to the OR textbooks on simplex method and dual-simplex method for detailed solution procedures.

ACCEPTED MANUSCRIPT

One-Tree Algorithm for Generating Multiple Optimal Solutions of 0-1 Integer Programs

The One-Tree algorithm is a variant of the standard B&B algorithm and aims at generating alternate optimal solutions instead of just an optimal solution. It uses a two phase approach to generate multiple optimal solutions. The first phase works similar to the standard B&B algorithm to obtain the optimal solution besides storing some bounded sub-problems in a set (P^{Stored}). During the second phase, the stored sub-problems are further explored to identify an alternative solution that results in the same objective value obtained in Phase 1. The step by step procedure of One-Tree algorithm is illustrated in Danna et al. (2007), and is summarized here with respect to our problem for the sake of completeness. Algorithms 2 and 3 illustrates the first and second phase of the One-Tree algorithm, respectively.

Algorithm 2 Phase 1 of One-Tree Algorithm

Initialize the proposed optimization model without binary restrictions (R) as a sub-problem, $P^S \leftarrow R$
Initialize the set of sub-problems stored for analysis in Phase 2 $P^{Stored} \leftarrow \emptyset$
Set a very high current best objective function value, $Z^* \leftarrow +\infty$
while $N^S \neq \emptyset$ **do**
    Choose a sub-problem $p_k \in P^S$
    Obtain the optimal solution, s_k and objective function value Z_k by solving p_k as a linear program
    if $Z_k > Z^*$ **then**
        Remove p_k from the set of sub-problems, $P^S \leftarrow P^S \setminus p_k$
    else
        if s_k satisfies binary restrictions **then**
            Update the current best solution and objective function: $s^* \leftarrow s_k$ and $Z^* \leftarrow Z_k$
            Update the list of stored sub-problems: $P^{Stored} \leftarrow P^{Stored} \cup \{p_k\}$
        else
            Choose a variable x_ij in p_k which does not satisfy binary restrictions
            Create two sub-problems (p_k^1 and p_k^2) by adding an additional constraint which forces $x_{ij} = 0$ in p_k^1 and $x_{ij} = 1$ in p_k^2 (i.e., $p_k^1 = p_k \cup \{x_{ij} = 0\}$ and $n_k^2 = n_k \cup \{x_{ij} = 1\}$)
            Update the list of sub-problems, $P^S \leftarrow P^S \cup \{p_k^1, p_k^2\} \setminus \{p_k\}$
        end if
    end if
end while
return optimal solution s^* and optimal objective function Z^*

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

Algorithm 3 Phase 2 of One-Tree Algorithm

Initialize the set of alternate optimal solutions, $S^{Alternative} \leftarrow s^*$
while $P^{Stored} \neq \emptyset$ **do**
    Choose a sub-problem $p_k \in P^{Stored}$
    Obtain the optimal solution, s_k and objective function value Z_k by solving p_k as a linear program
    if $Z_k \neq Z^*$ **then**
        Remove p_k from the set of stored sub-problems, $P^{Stored} \leftarrow P^{Stored} \setminus p_k$
    else
        if s_k satisfies binary restrictions **and** $s_k \notin S^{Alternative}$ **then**
            Update the set of alternate optimal solutions: $S^{Alternative} \leftarrow S^{Alternative} \cup s_k$
            Update the list of stored sub-problems: $P^{Stored} \leftarrow P^{Stored} \cup \{p_k\}$
        else
            Choose a variable x_ij such that it is not fixed by the local bounds of p_k
            Create two sub-problems (p_k^1 and p_k^2) by adding an additional constraint which forces $x_{ij} = 0$ in p_k^1 and $x_{ij} = 1$ in p_k^2 (i.e., $p_k^1 = p_k \cup \{x_{ij} = 0\}$ and $n_k^2 = n_k \cup \{x_{ij} = 1\}$)
            Update the list of stored sub-problems, $P^{Stored} \leftarrow P^{Stored} \cup \{p_k^1, p_k^2\} \setminus \{p_k\}$
        end if
    end if
end while
return the set of alternate optimal solutions, $S^{Alternative}$

ACCEPTED MANUSCRIPT

Appendix B: Formulation of the Benchmark Model

The benchmark optimization model (Pirim et al., 2018) aims to create compact and separated clusters using a mixed integer linear programming model. Compact clusters are obtained by minimizing the maximum of all cluster diameters (d_m), and separated clusters are formed by minimizing the maximum number of connections a node has with the nodes in other clusters (k_m^o). Model parameters used include the number of nodes (N), number of clusters (C), the minimum distance between two nodes i and j (D_{ij}), and the adjacency of two nodes i and j (A_{ij}) , where A_{ij} assumes the value of 1 if nodes i and j are connected directly else it assumes the value of 0. The decision variables are the maximum of all cluster diameters (d_m), the out connection number of the nodes with the maximum number of connections to nodes outside its cluster (k_m^o) and a binary variable x_{il} , which assumes the value of 1 if node i is assigned to group l. Readers may see the original article for complete details of the benchmark model. However, to ensure completeness we have summarized the benchmark model here.

$$\underset{d_m, k_m^o, x_{il}}{\text{Minimize}} \quad (d_m + k_m^o) \tag{20}$$

subject to:

$$d_m \geq D_{ij}(x_{il} + x_{jl} - 1) \qquad \forall i, j, l, (i < j) \tag{21}$$

$$\sum_{l=1}^{C} x_{il} = 1 \qquad \forall i \tag{22}$$

$$\sum_{i=1}^{N} x_{il} \geq 1 \qquad \forall l \tag{23}$$

$$\sum_{j=1}^{N} \Big(A_{ij} \times x_{jl} \Big) \geq \Big(\frac{\sum_{j=1}^{N} A_{ij}}{2} \Big) x_{il} \qquad \forall i, l \tag{24}$$

$$\sum_{j=1}^{N} \Big(A_{ij} \times x_{jl} \Big) \geq \Big(\sum_{j=1}^{N} A_{ij} \Big) x_{il} - k_m^o \qquad \forall i, l \tag{25}$$

$$x_{il} \in \{0, 1\} \qquad \forall i, l \tag{26}$$

ACCEPTED MANUSCRIPT

Appendix C: Visualization of Community Partition for Selected Networks

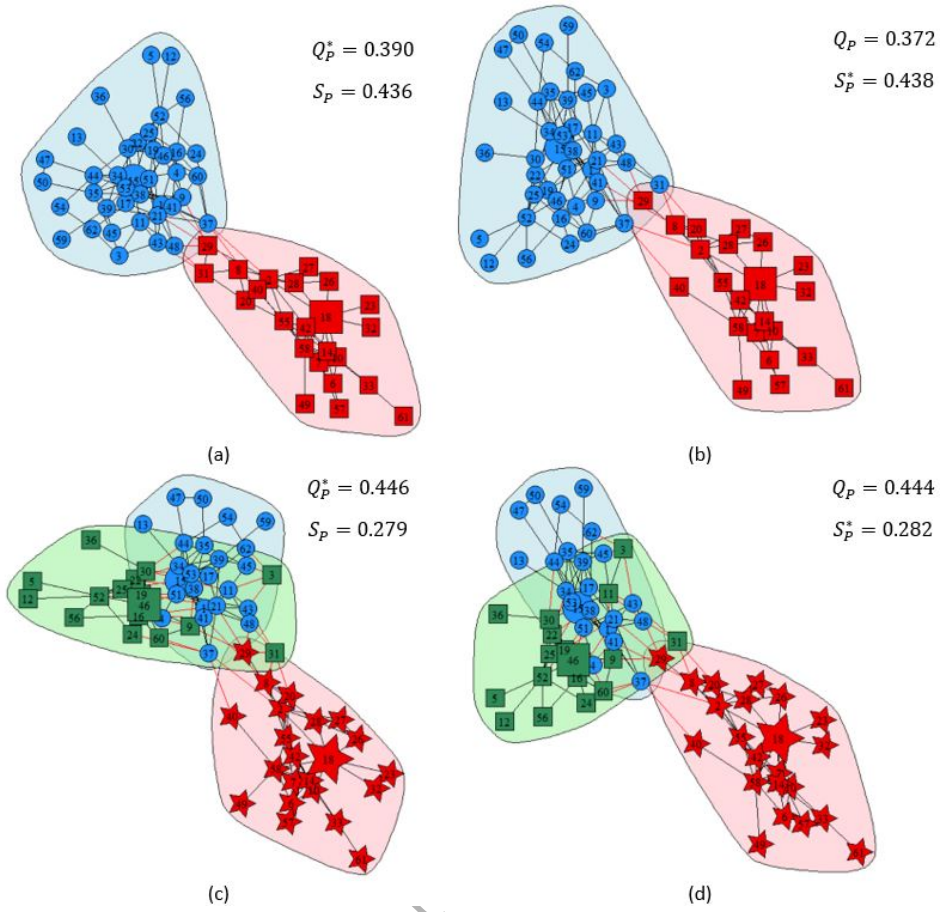


Figure 4: Dolphin network partition by proposed model: (a) 2-community best modularity partition (b) 2-community best Silhouette partition , (c) 3-community best modularity partition and (d)3-community best Silhouette partition (influential node identified by proposed model denoted by bigger size)

ACCEPTED

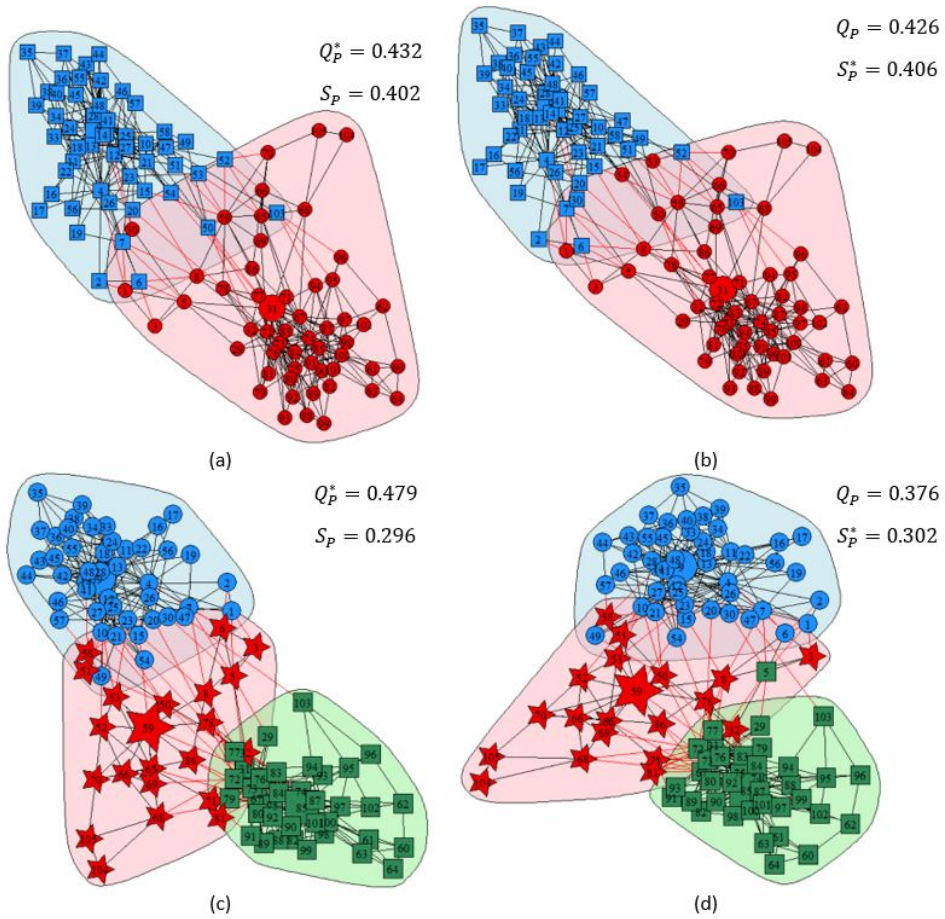


Figure 5: Book network partition by proposed model: (a) 2-community best modularity partition (b) 2-community best Silhouette partition , (c) 3-community best modularity partition and (d)3-community best Silhouette partition (influential node identified by proposed model denoted by bigger size)

ACCEPTED MANUSCRIPT

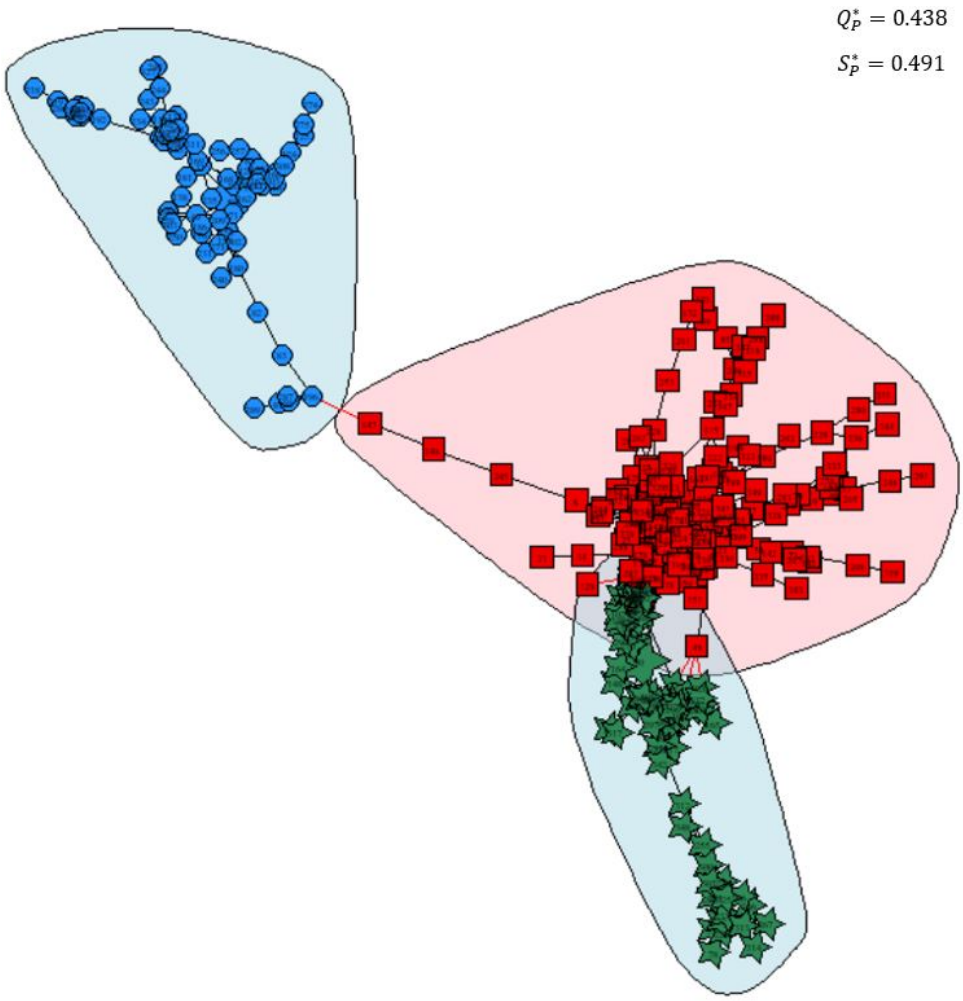


Figure 6: Human gene co-expression network partition by proposed model (influential node denoted by bigger size)

ACCEPTED

ACCEPTED MANUSCRIPT

Credit Author Statement

Manuscript title: Community Detection and Influential Node Identification in Complex Networks using Mathematical Programming

Authors: Sharan Srinivas and Chandrasekharan Rajendran

This statement is to acknowledge that all persons who meet authorship criteria are listed as authors, and all authors certify that they have participated sufficiently in the work to take public responsibility for the content, including participation in the concept, design, analysis, writing, or revision of the manuscript. Furthermore, each author certifies that this research work has not been submitted elsewhere for consideration.

The first author, **Sharan Srinivas**, was responsible for the following aspects of the research work:

- Conceptualization
- Data curation
- Formal analysis
- Methodology
- Investigation
- Visualization
- Original draft
- Review and Editing

The second author, **Chandrasekharan Rajendran**, also contributed equally for the following aspects:

- Conceptualization
- Methodology
- Investigation
- Original draft
- Review and Editing

Thank you.

Author's Name	**Author's Signature**	**Date**
Sharan Srinivas		May 7, 2019
Chandrasekharan Rajendran		May 7, 2019

ACCEPTED MANUSCRIPT