J Optim Theory Appl
https://doi.org/10.1007/s10957-017-1213-1

A Novel Mixed Integer Linear Programming Model for Clustering Relational Networks

Harun Pirim[1] · **Burak Eksioglu**[2] · **Fred W. Glover**[3]

Received: 7 June 2017 / Accepted: 26 December 2017

Abstract Integer programming models for clustering have applications in diverse fields addressing many problems such as market segmentation and location of facilities. Integer programming models are flexible in expressing objectives subject to some special constraints of the clustering problem. They are also important for guiding clustering algorithms that are capable of handling high-dimensional data. Here, we present a novel mixed integer linear programming model especially for clustering relational networks, which have important applications in social sciences and bioinformatics. Our model is applied to several social network data sets to demonstrate its ability to detect natural network structures.

Keywords Clustering · Mixed integer programming · Social networks

Mathematics Subject Classification 05C12 · 68R05 · 68R10 · 90C05 · 90C11 · 90C27 · 90C35 · 90C90

Communicated by Panos M. Pardalos.

✉ Harun Pirim
harunpirim@kfupm.edu.sa

Burak Eksioglu
burak@clemson.edu

Fred W. Glover
fredwglover@yahoo.com

[1] Systems Engineering, King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia

[2] Department of Industrial Engineering, Clemson University, Clemson, SC 29634, USA

[3] College of Engineering and Applied Science, University of Colorado, Boulder, CO 80309, USA

Published online: 05 January 2018

1 Introduction

Many industrial, biological, and sociological problems can be represented as networks, or graphs, including supply chain networks, gene co-expression networks, and ego networks. These networks, in turn, are constructed based on the complex relationships among objects within the massive populations that characterize either an organism or actors of a community. The use of clustering to understand and identify the most effective groups within a given population and to focus on the most remarkable components of a relational network fosters a better understanding of the relationships among objects.

Clustering approaches are mainly grouped into two categories: hierarchical and partition-based. In bottom-up hierarchical approaches, members form individual clusters at the beginning that then merge based on a certain proximity measure, and merged members form new clusters. These new clusters likewise merge in an iterative process that proceeds vertically from one stage to the next, and the construction terminates when all the members are in one cluster. By contrast, partition-based approaches proceed horizontally to allocate and reallocate members to groups, typically employing a pre-defined number of clusters.

Hierarchical and partition-based clustering methods comprise a vast set of clustering algorithms. While these algorithms are typically presented using a statistics or computer science perspective, it can also be valuable to draw on the perspective of discrete optimization to derive models to complement and improve the solutions obtained by other approaches. Integer programming (IP) formulations commonly generate clusters based either on maximizing the similarities among objects within clusters or on minimizing the similarities between clusters [1–3]. Maximizing the similarities among objects within clusters creates compact or homogeneous clusters, while minimizing the similarities between different clusters generates well-separated clusters.

With no universally accepted definition for "cluster", the study of clustering networks remains an ill-defined problem. Diverse and often ambiguous clustering goals result in a wide range of cluster definitions and approaches. In an effort to be more precise and to capture a spectrum of useful clustering goals effectively, we present a novel and rigorously defined mixed integer linear programming (MILP) model for graph clustering, utilizing an objective function designed to simultaneously engender compactness and separation of the clusters. By "compactness" we mean a property associated with a measure of cluster cohesiveness, and to ensure clusters that exhibit such a property we utilize an objective function that minimizes the largest diameter of all the clusters. By "separation" we mean a property associated with a measure of the distance of a cluster from its neighbors and we undertake to produce clusters with such a property by introducing constraints that assign a member to a cluster to yield fewer connections with objects outside the cluster.

Many network clustering algorithms have been proposed to identify coherent groups that exist inside a population represented as a network. However, not as much research has been devoted to analyzing clusters in a relational network by reference to discrete optimization models. We view the use of IP formalism to guide clustering algorithms as a relevant tool which, if managed properly, can fill a major gap in the literature.

IP formulations for clustering are generally presented with the goal of guiding the development of clustering algorithms, rather than with the purpose of submitting them to off-the-shelf IP solution software, since clustering is usually applied to big data sets, which can offer challenges to general IP software for finding optimal solutions.

A variety of IP formulations have appeared in recent years whose objectives exhibit varying degrees of relevance for developing new clustering algorithms. A comprehensive review of clustering algorithms designed for analyzing gene expression data is provided in [4]. We will briefly summarize some of the more salient contributions. Rao [5] investigated two clustering models based on integer programming representations. One was a nonlinear IP model that minimizes groups' sums of squares and the sums of the average squared distances among those groups. The second was a linear IP model that minimizes, respectively, the total and the maximum group distance. Extending Rao's work with an analysis of the IP formulations of clustering problems, Kusiak [6] discussed five different IP formulations for three clustering problems. The first problem has a traveling salesman formulation, and the second and third, detailed in [5], consist of an m-median problem that forms m clusters from n objects to minimize the sum of distances among objects to cluster medians.

To optimize data mining approaches for customer relations management applications, such as maximizing customer life-time value, Saglam et al. [7] applied a similar approach (to that found in [5]) of minimizing the maximum established group distance through a mixed integer programming (MIP) formulation. Similarly, Mehrotra and Trick [8] used a combinatorial approach based on graph-partitioning problems, to solve clique and clustering problems. Approaches included forming subgraphs, such that the sum of edge weights in every cluster is maximized, and the total edge weight among clusters is minimized. The authors presented mathematical formulations for the uncapacitated and capacitated clustering problems, as well as a set partitioning formulation that captured both of them.

Glover and Kochenberger [2] studied a clique-partitioning formulation based on associating variables with nodes, not edges. The new node-based formulation differs from the set partitioning formulation in [8] because it incorporates quadratic terms that are handled directly, not linearized, while the contrasting edge-based formulation is also similar to the uncapacitated clustering problem formulation developed in [8].

Xu et al. [9] formulated a mixed integer quadratic program (MIQP) to maximize the modularity score, a widely employed partitioning quality measure. The modularity score is expressed in quadratic terms with linear constraints and binary or continuous variables. The test networks presented here have 34, 62, 76, and 104 nodes. Similarly, Cafieri and Hansen [10] developed an MIQP model to refine heuristic solutions in relation to modularity maximization. Agarwal and Kempe [11] presented a clustering algorithm based on the rounding of the optimal solution to the linear programming (LP) clustering formulation. The objective function of the LP maximizes the modularity score expressed linearly. The highest network size solved by the LP approach had 235 nodes.

Martins [12] proposed a polynomial size integer LP model for the maximum edge weight k-plex partitioning problem, where each cluster of the partitioned graph is a k-plex, in which nodes belonging to each cluster hold a degree of at least $n - k$, where n is the number of nodes in the cluster. The model involves capacity constraints for the

clusters and upper bound constraints for the number of clusters. Other applications of clustering in networks can be found in [13–16].

The remainder of the paper is organized as follows: Sect. 2 describes our MILP model. Section 3 presents generalizations of the model, and Sect. 4 discusses and elaborates on the manipulation of an instrumental fraction f first defined in Sect. 3. Section 5 discusses how a network relaxation can act as a solution strategy, and example applications of the model are given in Sect. 6 to demonstrate its versatility and usefulness. Finally, concluding comments are provided in Sect. 7.

2 Our Mixed Integer Linear Programming Model

Our MILP model, presented below, was developed for clustering relational, or binary, networks. However, the model also applies to weighted networks when a threshold is applied to convert the weighted network to a binary network. In other words, given a network with arbitrary edge weights, preprocessing can be performed, which allows edges with weights less than a given threshold value to be removed from the network. The remaining edges have a weight of one, which thus results in a binary network.

As stated in Sect. 1, the model presented here is designed to create compact and separated clusters. In a binary network, the distance between any two objects is measured by the fewest number of edges that connect these two objects. To create compact clusters, the maximum of all cluster diameters should be minimized. Similarly, to create separate clusters, the maximum number of connections an object has with objects in other clusters should be minimized. A more detailed explanation is provided following the model's introduction. Our model employs a modified strong community structure defined in [17] as constraints and is formulated as follows:

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

subject to

$$d_m \geq d_{ij}(x_{il} + x_{jl} - 1) \quad \forall i, j, l, (i < j); \tag{1}$$

$$\sum_{l=1}^{c} x_{il} = 1 \quad \forall i; \tag{2}$$

$$\sum_{i=1}^{n} x_{il} \geq 1 \quad \forall l; \tag{3}$$

$$\sum_{j=1}^{n} (A_{ij} x_{jl}) \geq \left(\frac{\sum_{j=1}^{n} A_{ij}}{2} \right) x_{il} \quad \forall i, l; \tag{4}$$

$$\sum_{j=1}^{n} (A_{ij} x_{jl}) \geq \left(\sum_{j=1}^{n} A_{ij} \right) x_{il} - k_m^o \quad \forall i, l; \tag{5}$$

$$x_{il} \in \{0, 1\} \quad \forall i, l. \tag{6}$$

Model parameters include the number of objects (n), the number of clusters (c), the shortest path distances between objects i and j (d_{ij}), and the adjacency of objects i and j (A_{ij}), where A_{ij} is 1 if objects i and j are connected directly and 0 otherwise. The decision variables are x_{il}, d_m, and k_m^o defined as follows: x_{il} is 1 if object i is assigned to cluster l, 0 otherwise; d_m is the length of the longest diameter among all cluster diameters; k_m^o is the out connection number of the object with the maximum number of connections to objects outside its cluster. Constraint set (1) ensures that d_m is the maximum diameter, i.e., it is greater than the shortest distance between any two objects in any cluster as long as the two objects are in the same cluster. Constraint set (2) ensures that each object is assigned to exactly one cluster. Constraint set (3) ensures that a cluster has at least one object. Note that if $c < n$, then constraint set (3) cannot be satisfied. Therefore, a corresponding generalization of this constraint is discussed in Sect. 3.1. Constraint set (4) ensures that an object has at least as many connections with objects inside its cluster as outside, i.e., the term on the left is the number of connections object i has with other objects in the same cluster, and the term on the right is half the total number of connections for object i. Constraint set (5) ensures that k_m^o is greater than the number of connections any object has with other objects outside its own cluster. Constraint set (6) ensures that x_{il} are binary. The model has cn binary variables, two continuous variables, and $(\frac{cn^2}{2} + \frac{5cn}{2} + n + c)$ constraints.

3 Generalizations of the Model

An alternative way to express the MILP formulation for the network clustering problem is to represent the network as an undirected graph $G = (N, E)$, where $N = \{1, \ldots, n\}$ is the set of nodes, or objects to be clustered, and $E \subset N \times N$ is the set of edges, or connections between objects.

Let $N_i = \{j \in N : (i, j) \in E\}$ (i.e., the set of nodes adjacent to node i) and $n_i = |N_i|$ (i.e., the number of nodes adjacent to node i). Also, let $C = \{1, \ldots, c\}$ denote the set of clusters. Given that the binary variable $x_{il} = 1$ if and only if node i is assigned to cluster l, it follows that $\sum_{j \in N_i} x_{jl}$ = the number of nodes in cluster l adjacent to node i and $(n_i - \sum_{j \in N_i} x_{jl})$ = the number of nodes not in cluster l adjacent to node i. Constraint (4) now becomes

$$\sum_{j \in N_i} x_{jl} \geq \frac{1}{2} n_i x_{il} \quad i \in N, l \in C. \tag{7}$$

As in the original form of constraint set (4), this formulation is intended to model the requirement that, if node i is in cluster l ($x_{il} = 1$), the number of nodes in cluster l adjacent to node i must be at least as large as the number of nodes not in cluster l adjacent to node i. This requirement may be verified by noting that the final answer assures that $\sum_{j \in N_i} x_{jl} \geq (n_i - \sum_{j \in N_i} x_{jl})$. Hence, $\sum_{j \in N_i} x_{jl} \geq n_i/2$ given that $x_{il} = 1$, which constraint (7) appropriately accomplishes. Also, by rearranging the terms and letting z denote k_m^o from the previous formulation, constraint (5) can now be rewritten as

$$z \geq n_i x_{il} - \sum_{j \in N_i} x_{jl} \quad i \in N, l \in C. \tag{8}$$

Although not directly stated in the original formulation, constraint set (8) bounds z from below using the total number of nodes adjacent to node i minus the number of nodes in cluster l adjacent to node i, under the condition that node i belongs to cluster l. Under the minimization objective, the inequalities of constraint set (8) naturally ensure that only the maximum of the terms listed on the right affect the objective function. The new model formulation can now be described as follows, which we will refer to as MILP-C, where "C" stands for clustering.

$$\underset{d_m, z, x_{il}}{\text{minimize}} \quad (d_m + z)$$

subject to

$$d_m \geq d_{ij}(x_{il} + x_{jl} - 1) \quad (i, j) \in E, i < j, l \in C; \tag{9}$$

$$\sum_{l \in C} x_{il} = 1 \quad i \in N; \tag{10}$$

$$\sum_{i \in N} x_{il} \geq 1 \quad l \in C; \tag{11}$$

$$\sum_{j \in N_i} x_{jl} \geq \frac{1}{2} n_i x_{il} \quad i \in N, l \in C; \tag{12}$$

$$z \geq n_i x_{il} - \sum_{j \in N_i} x_{jl} \quad i \in N, l \in C; \tag{13}$$

$$x_{il} \in \{0, 1\} \quad i \in N, l \in C. \tag{14}$$

3.1 Generalization 1

As discussed above, the inequalities of constraint set (11) cannot be satisfied if $c < n$. However, note that for almost all realistic scenarios, c will be less than n, because $c = n$ implies each object will be in its own cluster by itself. Thus, to generate more meaningful clusters, we generalize constraint set (11) by replacing it with $\sum_{i \in N} x_{il} \geq L_l$. This generalization, which may have important practical implications for various applications, leads to different lower bounds on the number of objects in each cluster. Another reason to consider different L_l values is that constraints (10) and (11), together with (14), define a network, and a network-based optimization strategy can be introduced for solving MILP-C. Such a strategy, which includes additional elaborations of the network structure and its objective function, is discussed in Sect. 5.

3.2 Generalization 2

Another generalization can be achieved by replacing the coefficient of n_i in constraint set (12) by some fraction f. Note that when the coefficient is 1/2, as it currently

is, each object is forced to have at least half of its total connections with objects in its own cluster. Depending on the specific application, 1/2 may not be a desired ratio. Thus, using a different ratio f makes the model more flexible. Incorporating a selected fraction f has additional implications discussed in Sect. 3.3.

3.3 Generalization 3

Let α_{ij} be a measure of attractiveness or affinity between nodes i and j, for $(i, j) \in E$, to select pairs of objects with higher attractiveness values and group them in a common cluster. In such a case, if node i is assigned to a cluster l, then the sum of the attractiveness values for all nodes adjacent to i in cluster l should exceed the total attractiveness value for all nodes adjacent to i but not in cluster l. This value can be determined by introducing the following set of constraints, where $\alpha_i = \sum_{j \in N_i} \alpha_{ij}$

$$\sum_{j \in N_i} (\alpha_{ij} x_{jl}) \geq \frac{1}{2} \alpha_i x_{il} \quad i \in N, l \in C. \tag{15}$$

Note that this formulation is a direct generalization of constraint (12). The expression on the left identifies the total attractiveness value for cluster l from the perspective of node i. In other words, this part of the formulation is the sum of the attractiveness values for all nodes adjacent to i in cluster l. On the other hand, the total attractiveness value for all nodes adjacent to i but not in cluster l can be calculated by $\sum_{j \in N_i} \alpha_{ij}(1 - x_{jl})$. If node i is in cluster l, then the goal is for the following to hold $\sum_{j \in N_i} (\alpha_{ij} x_{jl}) \geq \sum_{j \in N_i} \alpha_{ij}(1 - x_{jl})$. This can be rewritten as $\sum_{j \in N_i} (\alpha_{ij} x_{jl}) \geq \sum_{j \in N_i} \alpha_{ij} - \sum_{j \in N_i} (\alpha_{ij} x_{jl})$ which then leads to $\sum_{j \in N_i} (\alpha_{ij} x_{jl}) \geq \alpha_i / 2$. The right side of the formulation is multiplied by x_{il} to ensure that the constraint is binding when $x_{il} = 1$, and leads to Eq. (15).

As discussed in Sect. 3.2, the coefficient 1/2 may be replaced by a different fraction f. However, rather than replacing formulation (12) with (15), both constraints may be included in the MILP formulation (perhaps for different fractions f), or Eq. (15) can embody a multi-objective theme by introducing constraints for different sets of attractiveness values α_{ij}. This type of multi-objective theme is amplified in Sect. 3.5.

3.4 Generalization 4

Analogous to Generalization 3, constraint (13) can be generalized to become

$$z \geq \alpha_i x_{il} - \sum_{j \in N_i} (\alpha_{ij} x_{jl}) \quad i \in N, l \in C \tag{16}$$

bounding z from below by the total sum of attractiveness values for all nodes adjacent to node i minus the corresponding sum of attractiveness values restricted to nodes in cluster l, under the condition that $x_{il} = 1$. Under the minimization objective, the maximum of the terms on the right side of (16) affects the objective function. Constraint

(16) can then be modified in a manner analogous to introducing the fraction f in Generalization 1 by multiplying the term $\sum_{j\in N_i}(\alpha_{ij}x_{jl})$ by a chosen fraction f.

3.5 Generalization 5

The multi-objective theme discussed in Generalization 3 can be handled in another way as follows: Denote the different sets of coefficients α_{ij} by α_{ij}^k for $k \in K = \{1, \ldots, k_o\}$, and let $\alpha_i^k = \sum_{j\in N_i} \alpha_{ij}^k$. Then constraint (16) can be expanded to become

$$z^k \geq \alpha_i^k x_{il} - \sum_{j\in N_i}(\alpha_{ij}^k x_{jl}) \quad i \in N, l \in C, k \in K. \tag{17}$$

Each z^k is stipulated to be a nonnegative continuous variable; likewise, z is replaced in the objective function by $\sum_{k\in K} z^k$. Alternatively, we may insert the maximum of the z^k values in the objective function by replacing z^k in constraint (17) with z, and keep z in the objective, as in the original formulation. Note that the α_{ij} values, and hence the α_{ij}^k values, should be normalized to ensure they are meaningful in the context of the problem solved, so that z implicitly receives a desired weight in the objective function relative to d_m.

3.6 Generalization 6

The treatment of Generalization 5 leads to a natural extension of the model by similarly forming different cases for inequality (9) that bound d_m. For example, these might include minimizing the maximum shortest path distance d_{ij} over nodes i and j that belong to a common cluster, and determining the sums of the d_{ij} distances over different clusters. We introduce a distance variable d_l applicable to cluster l, where d_l is continuous and bounded by $d_l \geq 0$, and the model replaces constraint set (9) by

$$d_l \geq d_{ij}(x_{il} + x_{jl} - 1) \quad (i, j) \in E, i < j, l \in C. \tag{18}$$

Then d_m in the objective can be replaced by $\sum_{l\in C} w_l d_l$ for different selected weights w_l. The effect of such weighting can also be implicitly accomplished by normalizing the d_{ij} coefficients, in which case the weights w_l can be disregarded. Finally, an extreme application might be to minimize the sum of all the effective d_{ij} values by introducing nonnegative continuous variables d_{ij}^e (the "e" stands for effective):

$$d_{ij}^e \geq d_{ij}(x_{il} + x_{jl} - 1) \quad (i, j) \in E, i < j, l \in C. \tag{19}$$

Then the variable d_m in the objective can be replaced by $\sum_{(i,j)\in E, i<j} d_{ij}^e$.

4 Interpreting and Manipulating the Fraction f

The fraction f, discussed in Generalization 2 as a replacement for the coefficient 1/2 of n_i in constraint (12), can be experimentally manipulated to determine good values in various contexts. As a basis for this manipulation, interpreting the meaning of f helps identify which values may reasonably be assigned to it. Specifically, if constraint (12) is replaced by the constraint

$$\sum_{j \in N_i} x_{jl} \geq f n_i x_{il} \quad i \in N, l \in C \tag{20}$$

then f can be interpreted as receiving the value $v/(v+1)$, where v is chosen so that the number of nodes in cluster l adjacent to node i, when node i itself belongs to cluster l, must be at least v times as great as the number of nodes not in cluster l adjacent to node i. This follows by simple extension of the logic that justifies constraint (12). Note that the stated goal for selecting v can be formulated as requiring $\sum_{j \in N_i} x_{jl} \geq v(n_i - \sum_{j \in N_i} x_{jl})$ when $x_{il} = 1$, which yields $\sum_{j \in N_i} x_{jl} \geq (v/(v+1))n_i$. Hence to make the inequality depend on $x_{il} = 1$ we have

$$\sum_{j \in N_i} x_{jl} \geq \frac{v}{v+1} n_i x_{il} \quad i \in N, l \in C. \tag{21}$$

For example, if $f = 2/3$, inequality (20) implies that the number of nodes adjacent to node i in cluster l must be at least twice the number of such nodes outside of l. Thus, selecting $f = 2/3$ is already close to the boundary of the largest value of f that may be desirable to investigate, and selecting f as large as 3/4 is exceedingly ambitious. Nevertheless, with this interpretation as a guideline, experiments may be performed with different values of f as a way to generate different sets of clusters, and the outcomes can be compared to determine which sets of clusters (and hence which values of f) have the most desirable features in a given application.

We can also go a step further, however, and allow f to be a variable, bounded, for example, by $\overline{f} \geq f \geq \underline{f}$. Term $-pf$ must be added to handle the objective function, in which p is a positive penalty to induce f to exceed $\underline{f}$. For p large enough, f will be driven to be as close to $\overline{f}$ as feasible. (If $p = 0$, then the optimum value for f will be $f = \underline{f}$, assuming that $\underline{f}$ is small enough to admit a feasible solution.

Allowing f to be variable in this manner causes our model to become a quadratic mixed integer formulation. A linear mixed integer formulation may be obtained by replacing $f x_{il}$ with a continuous variable f_{il}, which is assured to receive the correct value by adding the constraints $f \geq f_{il} \geq f - \overline{f}(1 - x_{il})$ and $\overline{f} x_{il} \geq f_{il} \geq 0$.

To avoid the expense of exploring the effects of this transformation, by using different penalty values p, using the variable f formulation just once with large p is easier. Using the variable f formulation with large p will allow us to identify the largest feasible value f^* for f. Then, the constant f formulation can be used to explore different constant values for f satisfying $f \leq f^*$. The computational time when solving for f^* can be reduced by temporarily dropping constraints (9) and (13), since they do not

limit feasibility and will not affect the maximum value of f. By significantly reducing the number of constraining inequalities in the MILP formulation, the problem will likely be solved more quickly and with less computational effort.

5 Network Relaxation as a Solution Strategy

A network optimization problem can typically be solved significantly faster than most other classes of LP problems, and if a specialized network solution routine is used in place of a general LP solution routine, then the speedup is appreciably increased. Considering also that the solution values of the network problem's variables are automatically integers, using network relaxation as a strategy for solving MILP-C should be considered.

As noted in Generalization 1, constraints (10), (11) and (14) by themselves provide a network structure. Such a structure can more generally be produced by a further generalization that replaces constraint (11) with

$$\sum_{i \in N} x_{il} + u_l - v_l = L_l \quad l \in C, \tag{22}$$

where u_l is a nonnegative slack variable and v_l is a nonnegative surplus variable. The variable v_l is implicit in constraint (11), though not normally identified. In a network setting, u_l is a slack arc and v_l is a surplus arc. Here, L_l can be considered as a target value for $\sum_{i \in N} x_{il}$, and u_l and v_l, respectively, allow the target to be under or over satisfied. Placing upper bounds on u_l and v_l can limit the amount by which $\sum_{i \in N} x_{il}$ deviates from the target. For example, setting $u_l \leq 0$, compelling $u_l = 0$, causes constraint (22) to correspond to constraint (11) with a right side of L_l.

These slack and surplus variables u_l and v_l can then be given nonnegative, typically positive, coefficients p_l and q_l in the objective function, where p_l penalizes the amount by which $\sum_{i \in N} x_{il}$ falls short of L_l and q_l penalizes the amount by which $\sum_{i \in N} x_{il}$ exceeds L_l. This approach, for example, can act as a special case where all clusters are targeted to contain roughly the same number of elements (e.g., L_l can be set, approximately, to equal to the "average" value n/c). Another special case could target different clusters to contain different numbers of elements, with varying penalties for deviations. In short, constraint (22) introduces a goal programming component into the network model. This approach may be accompanied by assigning nonnegative cost coefficients a_{il} to variables x_{il} to produce a network objective function of

$$\underset{u_l, v_l, x_{il}}{\text{minimize}} \quad \left(\sum_{l \in C} p_l u_l + q_l v_l - \sum_{i \in N, l \in C} a_{il} x_{il} \right).$$

Selecting large a_{il} values increase the inducement for $x_{il} = 1$. In this way, to induce two elements i and j to belong to the same cluster l, larger values can be assigned to both a_{il} and a_{jl}. These inducements can be manipulated by locally evaluating a given cluster assignment and adjusting coefficients according to the attractiveness,

by reference to the more complex objective for MILP-C, of shifting elements from their current clusters to other clusters. Because such a network relaxation strategy can likewise be employed for a variety of other clustering formulations, we are presenting our investigation into such a strategy in a sequel to this paper.

6 Application of the Model

The model was applied to small binary networks, several real social networks cited in [18], and a relatively large processed human gene co-expression network cited in [19]. Small size data sets were chosen since finding an optimal solution for larger data sets is hard. Once a model demonstrates that it performs well, heuristic algorithms may be used to solve the model. CPLEX [20] was utilized to solve the model using the data sets referenced above. The R [21] igraph library [22] was used to run the community structure-finding algorithm and calculate the modularity values. The Cluster package [23] was used to calculate Silhouette values. The clValid package [24] was used to calculate Dunn index values.

Figure 1 shows a complete network with six nodes and a tree with three nodes. The model does not have a feasible solution for these networks if $c > 1$. Since any division will leave at least one isolated object, the nodes for both networks in Fig. 1 should be in a single cluster.

Figure 2 illustrates the clusters found by the model on two different networks. The first is divided into $2, 3, \ldots, 8$ clusters, and the second is divided into $2, 3, \ldots, 9$ clusters. The first network has only one feasible solution with two clusters. The second network has feasible solutions with two and three clusters. The feasible solution with two clusters for the second network is obtained by merging any two of the three clusters.

The model was also applied on the real social network of 62 bottle nose dolphins in Doubtful Sound, New Zealand. The network is compiled by Lusseau [25] and its natural split into two is described in [26]. Our model detects the natural split of the dolphins into two groups as illustrated in Fig. 3.

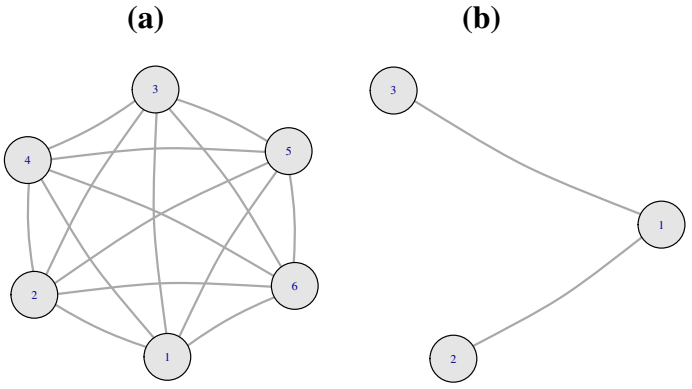

Fig. 1 Two different network structures for which our model has no feasible solution

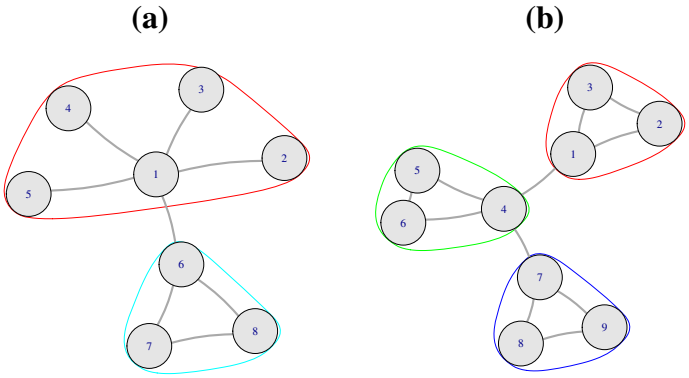

Fig. 2 Model solutions for **a** cluster size two and **b** cluster size three

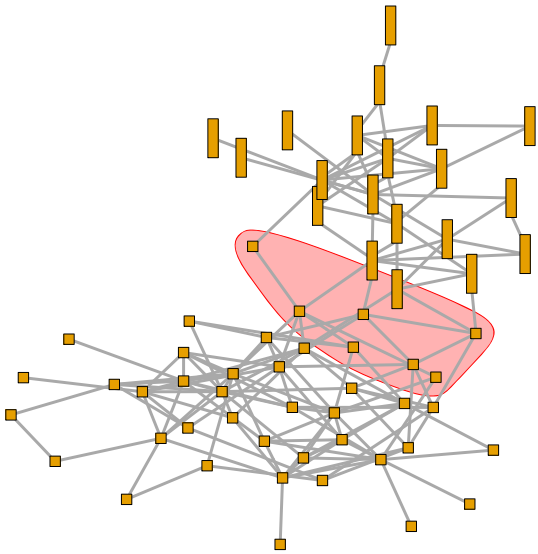

Fig. 3 The dolphins network: the network illustrates the two clusters having members with square and rectangular nodes found by the proposed model. However, the compared model assigns shaded nodes to the cluster with rectangular node members

Square nodes and rectangular nodes represent members of different groups. Our optimal solution is compared with a commonly used clustering model applied in [13]. This commonly used model inaccurately assigns shaded nodes to the cluster of rectangular nodes. Moreover, our proposed model found the optimal solution in seconds, while the commonly used clustering model ran for almost one day to find an optimal solution for its formulation, which produced a partition that deviated from the actual partition. Figures 4 and 5 illustrate the partitions of the data sets "books", "football", and "karate" by our model.

Computational experiments were run on a workstation with 4GB ram, Core i5 2.5 Ghz processor, and 32 bit Windows 7 operating system. Although this study's main focus is to develop a mathematical model for clustering relational networks and detecting their natural splits, modularity values were calculated because maximizing

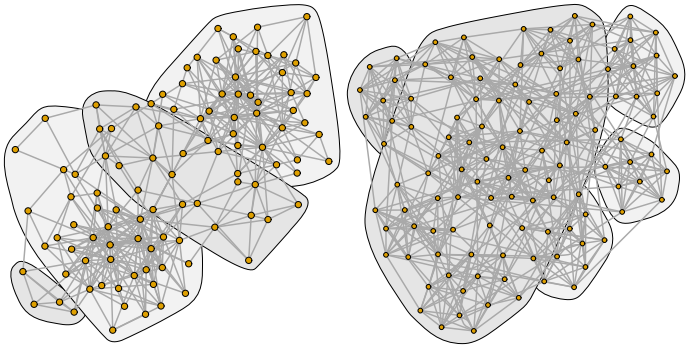

Fig. 4 Partitions of books (on the left) and football networks by the proposed model

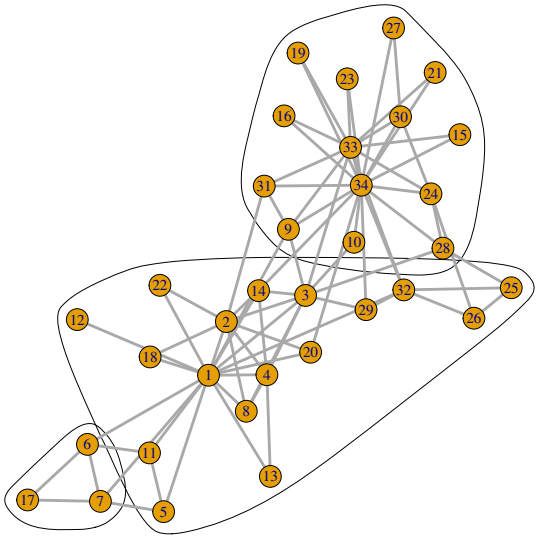

Fig. 5 Partition of the karate network into three clusters by the proposed model

modularity is a commonly used objective function for algorithms in network research. Modularity is defined as improvement on random connectivity. The higher the value of modularity, the better the partition is, so forming clusters, or communities, that maximize the modularity is desirable. Table 1 summarizes the modularity values found by our MILP model, compared to a popular community structure-finding algorithm based on the betweenness results reported in [26]. Modularity values found by our proposed model are represented by the column Modularity M. Not surprisingly, the model does not produce better modularity values than Newman's method [26], shown as Modularity N in Table 1, since Newman's method is designed to maximize modularity. However, without being designed to maximize modularity, our model finds competitive modularity values.

Although modularity is used in community structure-finding algorithms, other internal validation indices are more commonly used. A recent review on cluster validation indices can be found in [27], employing 30 different indices and reporting three ranked

Table 1 Modularity values

Data set	Nodes	Edges	Clusters	Modularity M	Modularity N
Karate	34	78	3	0.335	0.401
Dolphins	62	159	5	0.450	0.519
Books	105	441	4	0.511	0.517
Football	115	613	5	0.460	0.601
Human	349	1418	3	0.271	0.651

Table 2 Silhouette values

Data set	Nodes	Edges	Clusters	Silhouette M	Silhouette N
Karate	34	78	3	0.222	0.166
Dolphins	62	159	3	0.302	0.288
Books	105	441	3	0.288	0.257
Football	115	613	6	0.141	0.301
Human	349	1418	3	0.523	0.302

Table 3 Dunn values

Data set	Nodes	Edges	Clusters	Dunn M	Dunn N
Karate	34	78	3	0.111	0.330
Dolphins	62	159	5	0.250	0.200
Books	105	441	3	0.250	0.250
Football	115	613	6	0.250	0.330
Human	349	1418	3	0.077	0.125

groups of validation indices. Both the Silhouette index [28] and the Dunn index [29] are in the first ranked group. Moreover, the Silhouette index is the first index in the first ranked group. These indices were used for the data sets above, and the results, obtained by our model and with Newman's community structure-finding algorithm, were compared based on betweenness [26]. The results are summarized in Tables 2 and 3. The Silhouette M and Dunn M columns list the Silhouette and Dunn index values determined by our model.

The Silhouette index measures both cohesion and separateness, features incorporated into our model's objective function. This index can take values between -1 and 1, and being closer to 1 indicates a better clustering. The Dunn index is the ratio of the minimum distance among objects of different clusters to the maximum distance between objects in the same cluster. The index can take values between zero and infinity.

Our model finds better Silhouette values than the community structure-finding algorithm, except for the Football data set. One reason is the increased difficulty of solving the model for greater numbers of divisions. In other words, the model may find better values for greater numbers of divisions, but it requires disproportionately

more effort to do so. The proposed model finds similar Dunn index values compared to values found by the community structure-finding algorithm.

The approach of using the fraction f, elaborated in Sect. 4, was applied to the trivial network shown in Fig. 2b. The optimum values of f correspond to different numbers of clusters. The model to find the optimum f value is formulated as

$$\underset{d_m, k_m^o, f, f_{il}, x_{il}}{\text{minimize}} \quad (d_m + k_m^o - pf)$$

subject to

$$\sum_{l=1}^{c} x_{il} = 1 \quad \forall i; \tag{23}$$

$$\sum_{i=1}^{n} x_{il} \geq 1 \quad \forall l; \tag{24}$$

$$\sum_{j=1}^{n} A_{ij} x_{jl} \geq \sum_{j=1}^{n} A_{ij} f_{il} \quad \forall i, l; \tag{25}$$

$$f \geq f_{il} \quad \forall i, l; \tag{26}$$

$$f_{il} \geq f - \overline{f}(1 - x_{il}) \quad \forall i, l; \tag{27}$$

$$\overline{f} x_{il} \geq f_{il} \quad \forall i, l; \tag{28}$$

$$x_{il} \in \{0, 1\} \quad \forall i, l; \tag{29}$$

$$f_{il}, d_m, k_m^o \geq 0 \quad \forall i, l. \tag{30}$$

The optimal solution for the nine-node network present in Fig. 2b is $f = 0.5$ when the number of clusters is specified to be three. The optimal f value decreases to 0.25 and 0 when the number of clusters is specified to be four and five, respectively. The optimal f values for the Karate data set are 0.66, 0.5, 0.5, 0.41, and 0.33 with number of clusters 2, 3, 4, 5, and 6, respectively. The optimal f value for the Dolphins data set with two clusters is 0.57. The optimal f value for the Books data set with three clusters is 0.53.

7 Conclusions

We have introduced a novel mixed integer linear programming (MILP) model for clustering relational networks that detects natural partitions in the data defining a social network. Our model additionally encompasses numerous generalizations to handle an exceedingly broad range of clustering goals. Some of these goals do not have to be explicitly identified in the model. For example, a benchmark test shows that our method yields a competitive set of modularity values without making reference to a goal of maximizing modularity. Moreover, except for a single data set, the model also determined better Silhouette index values than the community structure-finding

algorithm designed to yield such values. These outcomes suggest the utility of our model as a guide for creating novel heuristic clustering algorithms for larger data sets.

Similarly, in tests conducted on the Dolphins data set, our model was demonstrated to yield outcomes superior to those obtained by the widely used Newman clustering model used as a reference for comparison. The optimum solution of our proposed model matched the actual partition exactly, while the optimum solution of the Newman model deviated from the actual partition. Unlike the Newman model, which required hours to determine an optimum solution by its criterion, our model determined the optimum solution in mere seconds. Hence, we conclude our model is a more suitable guide to developing heuristic algorithms for larger data sets.

It can be worthwhile to incorporate additional elements within our model. The proposed MILP formulation uses the number of clusters as a parameter. However, Silhouette plots, based on different numbers of clusters, provides a useful method for identifying a desired number of clusters by choosing the partition that gives the highest Silhouette value. Different network topologies will affect the objective function value of the model, since the objective function assumes a binary network. For example, if the network is dense, clusters will have small diameters enclosing the objects within them and will possess a large number of connections with other clusters. Furthermore, a dense binary network means that k_m^o will dominate the d_m term in the objective. Therefore, this phenomenon should be the subject of future parametric optimization studies involving the use of different coefficients for both d_m and k_m^o. The proposed model is especially appropriate when the binary network is sparse.

The main focus and the contribution of our study has been to determine the efficacy of the proposed model to derive important structures with small- to medium-sized networks, broadening the focus of earlier studies. Our model generalizations and our associated network relaxation strategy provide additional foundation for future studies involving networks of even greater complexity.

Acknowledgements This work is partially supported by KFUPM DSR project JF121001. We would like to thank the two anonymous referees and the Associate Editor for the feedback they provided which improved the quality of the paper significantly.

References

1. Do, J.H., Choi, D.K.: Clustering approaches to identifying gene expression patterns from DNA microarray data. Mol. Cells **25**(2), 279–288 (2008)
2. Glover, F.W., Kochenberger, G.: New optimization models for data mining. Int. J. Inf. Technol. Decis. Mak. **05**(04), 605–609 (2006)
3. Shamir, R., Sharan, R.: Algorithmic approaches to clustering gene expression data. In: Jiang, T., Smith, T., Xu, Y., Zhang, M. (eds.) Current Topics in Computational Biology, pp. 269–299. MIT Press, Cambridge (2002)
4. Pirim, H., Eksioglu, B., Perkins, A.D., Yuceer, C.: Clustering of high throughput gene expression data. Comput. Oper. Res. **39**(12), 3046–3061 (2012)
5. Rao, M.R.: Cluster analysis and mathematical programming. J. Am. Stat. Assoc. **66**(335), 622–626 (1971)
6. Kusiak, A.: Analysis of integer programming formulations of clustering problems. Image Vis. Comput. **2**(1), 35–40 (1984)
7. Saglam, B., Salman, F.S., Sayin, S., Turkay, M.: A mixed-integer programming approach to the clustering problem with an application in customer segmentation. Eur. J. Oper. Res. **173**(3), 866–879 (2006)

8. Mehrotra, A., Trick, M.A.: Cliques and clustering: a combinatorial approach. Oper. Res. Lett. **22**(1), 1–12 (1998)
9. Xu, G., Tsoka, S., Papageorgiou, L.G.: Finding community structures in complex networks using mixed integer optimisation. Eur. Phys. J. B **60**(2), 231–239 (2007)
10. Cafieri, S., Hansen, P.: Using mathematical programming to refine heuristic solutions for network clustering. In: Batsyn, M., Kalyagin, V., Pardalos, P. (eds.) Models, Algorithms and Technologies for Network Analysis, Proceedings in Mathematics & Statistics, vol. 104. Springer, Switzerland (2014)
11. Agarwal, G., Kempe, D.: Modularity-maximizing graph communities via mathematical programming. Eur. Phys. J. B **66**(3), 409–418 (2008)
12. Martins, P.: Modeling the maximum edge-weight k-plex partitioning problem. Cornell University. arxiv:1612.06243 [math.co] (2016)
13. Nascimento, M., Toledo, F., de Carvalho, A.: Investigation of a new GRASP-based clustering algorithm applied to biological data. Comput. Oper. Res. **37**(8), 1381–1388 (2010)
14. Pirim, H., Eksioglu, B., Perkins, A.D.: Clustering high throughput biological data with B-MST, a minimum spanning tree based heuristic. Comput. Biol. Med. **62**, 94–102 (2015)
15. Pirim, H., Gautam, D., Bhowmik, T., Perkins, A.D., Eksioglu, B., Alkan, A.: Performance of an ensemble clustering algorithm on biological data sets. Math. Comput. Appl. **16**(1), 87–96 (2011)
16. Tan, M.P., Smith, E.N., Broach, J.R., Floudas, C.A.: Microarray data mining: a novel optimization-based approach to uncover biologically coherent structures. BMC Bioinform. **9**, 1–21 (2008)
17. Radicchi, F., Castellano, C., Cecconi, F., Loreto, V., Parisi, D.: Defining and identifying communities in networks. Proc. Natl. Acad. Sci. USA **101**(9), 2658–2663 (2004)
18. Cafieri, S., Hansen, P., Liberti, L.: Locally optimal heuristic for modularity maximization of networks. Phys. Rev. E **83**, 1–8 (2011)
19. Prieto, C., Risueno, A., Fontanillo, C., Rivas, J.D.L.: Human gene coexpression landscape: confident network derived from tissue transcriptomic profiles. PLoS ONE **3**(12), 1–14 (2008)
20. IBM ILOG CPLEX 12.6 (2014)
21. R Core Team: R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria (2014). http://www.R-project.org/
22. Csardi, G., Nepusz, T.: The igraph software package for complex network research. Int. Complex Syst. **1695**, 1–9 (2006)
23. Maechler, M., Rousseeuw, P., Struyf, A., Hubert, M., Hornik, K.: Cluster: Cluster Analysis Basics and Extensions, R package version 2.0.6 edn. (2017)
24. Brock, G., Pihur, V., Datta, S., Datta, S.: clValid: Validation of Clustering Results, R package version 0.6-6 edn. (2014)
25. Lusseau, D.: The emergent properties of a dolphin social network. Proc. R. Soc. Lond. B Biol. Sci. **270**(2), 186–188 (2003)
26. Newman, M.E.J., Girvan, M.: Finding and evaluating community structure in networks. Phys. Rev. E **69**, 026113 (2004)
27. Arbelaitz, O., Gurrutxaga, I., Muguerza, J., Perez, J.M., Perona, I.: An extensive comparative study of cluster validity indices. Pattern Recognit. **46**(1), 243–256 (2013)
28. Rousseeuw, P.J.: Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. J. Comput. Appl. Math. **20**, 53–65 (1987)
29. Dunn, J.C.: Well-separated clusters and optimal fuzzy partitions. J. Cybernet. **4**(1), 95–104 (1974)