Complete mixed integer linear programming formulations for modularity density based clustering

Alberto Costa[a,*], Tsan Sheng Ng[b], Lin Xuan Foo[b]

[a] *National University of Singapore and ETH Zurich, Future Resilient Systems, Singapore*
[b] *Department of Industrial and Systems Engineering, National University of Singapore, Singapore*

Abstract

Modularity density maximization is a clustering method that improves some issues of the commonly-used modularity maximization approach. Recently, some Mixed-Integer Linear Programming (MILP) reformulations have been proposed in the literature for the modularity density maximization problem, but they require as input the solution of a set of auxiliary binary Non-Linear Programs (NLPs). These can become computationally challenging when the size of the instances grows. In this paper we propose and compare some explicit MILP reformulations of these auxiliary binary NLPs, so that the modularity density maximization problem can be completely expressed as MILP. The resolution time is reduced by a factor up to two order of magnitude with respect to the one obtained with the binary NLPs.

Keywords: clustering, modularity density, mixed integer linear programming, reformulations

1. Introduction

Given an undirected unweighted graph $G = (V, E)$, where V is the vertex set and E is the edge set, cluster analysis refers to finding a partition of V into disjoint groups called clusters (or communities) such that vertices in the same cluster are densely connected to each other and less connected to those in other clusters. This is a very important problem with applications

*Corresponding author.
Email addresses: costa@lix.polytechnique.fr (Alberto Costa), isentsa@nus.edu.sg (Tsan Sheng Ng), foolinxuan@u.nus.edu (Lin Xuan Foo)

in many fields, e.g., social networks [1], recommender systems [2], biology and bioinformatics [3, 4].

To date, there have been numerous methods proposed to identify clusters in a graph. Some methods do not require a function to be optimized, for example the Girvan and Newman's heuristic [1] where the edge with highest 'betweenness' (i.e., number of shortest paths running along that edge) is iteratively removed. Other approaches can be based on some rules (usually related to the number of neighbors of vertices inside and outside their corresponding cluster) that each cluster must respect. In this category we can find the *strong* and *weak* definition of Radicchi et al. [5], the *semi-strong* and *extra-weak* definitions of Hu et al. [6], and the *almost-strong* definition of Cafieri et al. [7]. Alternatively, clustering can be expressed in terms of an objective function to optimize, and this is interesting from a mathematical programming point of view. One of the commonly-used objective functions is *modularity*, which is defined as the fraction of edges within clusters minus the expected fraction of such edges in a random graph with the same degree distribution. It is to be noted that the problem of clustering based on maximization of the modularity function is *NP*-hard [1, 8, 9].

Notwithstanding its *NP*-hardness, clustering solutions arising from modularity maximization also present some practical issues, in particular, *resolution limit* and *degeneracy*. The former refers to the possibility of small clusters being merged with other clusters and thus not detected [10]. The latter occurs when there are several high quality local optima and a global optimum cannot be easily found [11]. In order to overcome the resolution limit issue associated with modularity maximization, an alternative clustering measure, *modularity density*, was proposed in [12]. The main difference with respect to modularity is that modularity density takes into account the size of the clusters. More precisely, according to [13] the modularity density D can be defined as:

$$D = \sum_{c \in C} \left(\frac{2m_c - \bar{m}_c}{n_c} \right), \tag{1}$$

where C is the set of clusters (whose cardinality $|C|$ is not know in advance), m_c is the number of edges having both end vertices inside the cluster c (inner edges), $\bar{m}_c$ is the number of edges having one end vertex in c and the other one outside c (cut edges), and n_c is the number of vertices inside the cluster c (i.e., the size of the cluster).

Unfortunately, the optimization problem based on maximizing modularity density, termed as the *modularity density maximization* (MDM) problem, is not easy to solve. More precisely, MDM was formulated as a binary Non-Linear Programming (0-1 NLP) problem in [12], and in general non-linear, non-convex problems having integer variables are solved by means of MINLP (Mixed-Integer Non-Linear Programming) solvers, which may converge slowly to the optimal solution even for small size instances. Nevertheless, as pointed out in [14], it is still unclear if MDM is *NP*-hard, since the proof provided in [12] is wrong.

In [13] some Mixed-Integer Linear Programming (MILP) reformulations of MDM were proposed. However, to implement these MILP reformulations, the authors require the solution of a set of auxiliary 0-1 NLP problems. Albeit these auxiliary 0-1 NLPs are easier to solve than the MDM problem presented in [12], they can still be quite challenging when the size of the instances is large. Hence, it is important to find efficient way to solve them. In this paper, we derive several different exact MILP reformulations of the above-mentioned auxiliary NLP problems, thus obtaining complete MILP formulations of MDM. To do so, we employ some reformulation techniques, e.g., linearization of bilinear terms, expansion of integers in power of two, and reformulation of fractional programs. Finally, we perform several numerical studies to show that the proposed MILP reformulations are more computationally efficient, especially when problem instances scale up.

The rest of the paper is organized as follows: In Section 2 we introduce the techniques employed later to linearize the non-linear problems. In Section 3 we give a formal problem statement of the MDM and the related auxiliary 0-1 NLPs. The MILP reformulations of these problems are then presented in Sections 4 and 5. In Section 6 we also prove that the optimal solution of the MILPs and those obtained in [13] by solving the continuous relaxation of the 0-1 NLPs are the same, i.e., we can discard the integrality constraint from the 0-1 NLPs and the solution is still integral. Computational results showing the efficiency of our new models are reported in Section 7. Finally, conclusions are drawn in Section 8.

2. Preliminaries: reformulation techniques

In this section we introduce the techniques that will be employed later to derive the MILP reformulations of the 0-1 NLPs.

2.1. Linearization of multi-linear terms

Let $\mathbf{x} = [x_1, \ldots, x_k]$ be a vector of k variables defined on the hyperrectangle $[\mathbf{x}^L, \mathbf{x}^U] \subseteq \mathbb{R}^k$, where $\mathbf{x}^L$ and $\mathbf{x}^U$ are the vectors of lower and upper bounds, respectively. A multi-linear term is a function $\mu(\mathbf{x}) = x_1 \cdots x_k$ that represents the product of those k variables.

In general, optimization problems involving multi-linear functions are non-linear and non-convex. A common approach to deal with such terms is to linearize them, i.e., replace each term with a new variable and a set of linear constraints. The linearization of a multi-linear term produces a convex relaxation, i.e., the solution of the resulting problem provides an upper (lower) bound on the original maximization (minimization) problem. However, in some special cases the resulting linearized problem is an *exact* reformulation (i.e., its solution is the same as that of the original problem). This happens when, among the k variables of $\mu(\mathbf{x})$, at most one is continuous. The problems that we consider here involve mostly bilinear terms (i.e., products of two variables) where one or both the variables are binary, so we can obtain an exact linearization of these terms.

We review in the following two ways to linearize bilinear terms. The first one is based on the so-called McCormick's inequalities, which are constraints defining the convex envelope of the bilinear term. The second approach, termed as the *dual reformulation* in [15], is based on the vertex polyhedrality property of multi-linear terms, i.e., their convex envelope can be described as a convex combination of the vertices of the corresponding box [16, 17]. Note that both methods describe the same convex region, but may yield different solution efficiency, as shown in [15].

2.1.1. McCormick's inequalities

Let $\mu(\mathbf{x}) = x_1 x_2$ be a bilinear term, where $x_1 \in [L_{x_1}, U_{x_1}]$ and $x_2 \in [L_{x_2}, U_{x_2}]$. The convex envelope of the bilinear term is defined by the following McCormick's inequalities [18]:

$$
\begin{aligned}
w_{1,2} &\geq L_{x_1} x_2 + L_{x_2} x_1 - L_{x_1} L_{x_2} \\
w_{1,2} &\geq U_{x_1} x_2 + U_{x_2} x_1 - U_{x_1} U_{x_2} \\
w_{1,2} &\leq L_{x_1} x_2 + U_{x_2} x_1 - L_{x_1} U_{x_2} \\
w_{1,2} &\leq U_{x_1} x_2 + L_{x_2} x_1 - U_{x_1} L_{x_2},
\end{aligned}
$$

where the term $x_1 x_2$ is replaced by the new variable $w_{1,2}$. As stated earlier, in general $w_{1,2} \neq x_1 x_2$ but if at least one variable is integer we can obtain the equivalence. In the special case where both variables are binary, the McCormick's inequalities assume the form of the *Fortet's inequalities* [19]:

$$
\begin{aligned}
&w_{1,2} \geq 0 \\
&w_{1,2} \geq x_1 + x_2 - 1 \\
&w_{1,2} \leq x_1 \\
&w_{1,2} \leq x_2.
\end{aligned}
$$

These inequalities can be obtained by the McCormick's ones by replacing each lower and upper bound with 0 and 1, respectively. It is easy to check that if $x_1 \in \{0, 1\}$ and $x_2 \in \{0, 1\}$ the Fortet's inequalities force $w_{1,2}$ to be equal to $x_1 x_2$.

2.1.2. Dual reformulation

Multi-linear terms are vertex polyhedral, i.e., their convex envelopes are the convex envelopes of their restriction to the vertices of the corresponding box, as shown in [16, 17]. Hence, we can derive an alternative way to describe the convex envelope of a multi-linear term. More precisely, let $w(\mathbf{x})$ be the value of the convex envelope of $\mu(\mathbf{x})$ and $\mathbf{p}_1, \ldots, \mathbf{p}_{2^k} \in \mathbb{R}^k$ the vertices of the box $[\mathbf{x}^L, \mathbf{x}^U] \subseteq \mathbb{R}^k$. Since this is a polyhedron, we can express any point $\mathbf{x} \in [\mathbf{x}^L, \mathbf{x}^U]$ as a convex combination of the vertices $\mathbf{p}_1, \ldots, \mathbf{p}_{2^k}$:

$$
\begin{aligned}
&\mathbf{x} = \sum_{i=1}^{2^k} \lambda_i \mathbf{p}_i \\
&\sum_{i=1}^{2^k} \lambda_i = 1 \\
&\forall i \in \{1, \ldots, 2^k\}, \quad \lambda_i \geq 0.
\end{aligned}
$$

Since the value of the convex envelope of $\mu(\mathbf{x})$ on the vertex $\mathbf{p}_i$ is equal to $\prod_{j=1}^{k} p_{i,j}$, we can express $w(\mathbf{x})$ as:

$$
w(\mathbf{x}) = \sum_{i=1}^{2^k} \lambda_i \prod_{j=1}^{k} p_{i,j}.
$$

To make a comparison with the McCormick's inequalities, consider a bilinear term $x_1 x_2 \in [\mathbf{x}^L, \mathbf{x}^U]$. Hence, $k = 2$, and the 4 vertices of the box are

$\mathbf{p}_1 = (L_{x_1}, L_{x_2})$, $\mathbf{p}_2 = (L_{x_1}, U_{x_2})$, $\mathbf{p}_3 = (U_{x_1}, L_{x_2})$, $\mathbf{p}_4 = (U_{x_1}, U_{x_2})$. The dual approach defines the convex envelope of the bilinear term as:

$$
\begin{aligned}
& x_1 = (\lambda_1 + \lambda_2)L_{x_1} + (\lambda_3 + \lambda_4)U_{x_1} \\
& x_2 = (\lambda_1 + \lambda_3)L_{x_2} + (\lambda_2 + \lambda_4)U_{x_2} \\
& \sum_{i=1}^{4} \lambda_i = 1 \\
& w_{1,2} = \lambda_1 L_{x_1} L_{x_2} + \lambda_2 L_{x_1} U_{x_2} + \lambda_3 U_{x_1} L_{x_2} + \lambda_4 U_{x_1} U_{x_2} \\
& \forall i \in \{1, \ldots, 4\}, \quad \lambda_i \geq 0.
\end{aligned}
$$

Notice that, as reported in [15], one advantage of the dual approach is that it is suitable to express any multi-linear term, unlike a *primal* method (i.e., based on defining the convex envelope using linear inequalities, for example the McCormick's inequalities for a product of two variables). As a matter of fact, other than the bilinear case, inequalities defining the convex envelopes of multi-linear terms are known explicitly only for trilinear terms [20, 21] and partly for quadrilinear terms [22].

2.2. Binary expansion of integers

Consider an integer variable $y \in \{0, \ldots, n\}$. This can be expressed as $y = \sum_{i=1}^{n} b_i$, where b_i are binary variables. Alternatively, we can express it by means of a binary expansion:

$$
y = \sum_{i=0}^{u} 2^i b_i, \quad b_i \in \{0, 1\},
$$

where $u = \lceil \log_2(n+1) \rceil - 1$. This is because the maximum value that can be represented by the term $\sum_{i=0}^{u} 2^i b_i$ is where all $b_i = 1$. In order to be able to represent all values of y, u can be computed as such:

$$
\begin{aligned}
\sum_{i=0}^{u} 2^i = 2^{u+1} - 1 &\geq n \\
u &\geq \log_2(n+1) - 1 \\
u &= \lceil \log_2(n+1) \rceil - 1.
\end{aligned}
\tag{2}
$$

There are two advantages of using the binary representation in a mathematical program. The first advantage is that less binary variables are required. While n binary variables are required originally, the binary representation requires only $O(\log(n))$ binary variables. The second advantage is that by using binary representation, we can avoid the symmetries arising in the original representation. For example, consider the set of values of b_i such that $\sum_{i=1}^{n} b_i = k$, where $k \leq n$ is an integer number. Using the original representation, there can be $\binom{n}{k}$ symmetric solutions. However, with the binary representation, there is only one way to represent k. Often it is better to avoid symmetries as they increase the size of the search space, and this can impair the performance of Branch-and-Bound algorithms [23].

2.3. Linearization of fractional terms

Consider a 0-1 NLP where the objective function is a ratio of two functions (the problems we study in the following belong to this class):

$$\begin{aligned} \max \quad & \left(\frac{\sum_{j \in J \subseteq I} c_j y_j}{\sum_{i \in I} y_i} \right) \\ \text{s.t.} \quad & L \leq \sum_{i \in I} y_i \leq U \\ & \forall i \in I, \quad y_i \in \{0, 1\}, \end{aligned}$$

where the vector $\mathbf{c}$ is a parameter and $L \geq 0$ and $U \leq |I|$ are known bounds of $\sum_{i \in I} y_i$. By introducing an additional variable h, the above fractional program can be reformulated in a similar fashion to the Charnes-Cooper transformation [24] as follows:

$$\begin{aligned} \max \quad & \sum_{j \in J} c_j y_j h \\ \text{s.t.} \quad & L \leq \sum_{i \in I} y_i \leq U \\ & \sum_{i \in I} y_i h = 1 \\ & \forall i \in I, \quad y_i \in \{0, 1\}. \end{aligned}$$

It can be seen that the resulting formulation is a bilinear program due to the products between the binary variables y and the continuous variable h. The bilinear program can then be linearized exactly using the techniques discussed earlier. The McCormick's inequalities and dual reformulation requires us to know the bounds of h, which can be derived as such:

$$\left.\begin{array}{r} L \leq \sum_{i \in I} y_i \leq U \\ \sum_{i \in I} y_i h = 1 \end{array}\right\} \quad \Longrightarrow \quad \tfrac{1}{U} \leq h \leq \tfrac{1}{L}.$$

3. Problem definition

We provide now a more detailed statement of the modularity density maximization problem. Let $y_{i,c}$ be a binary variable equal to 1 if node $v_i \in V$ belongs to cluster c, and 0 otherwise. Moreover, let k_i be the degree of vertex v_i, i.e., the number of its neighbors. Starting from Definition (1), we can express the number of inner edges (m_c) as $\sum_{\{v_i,v_j\}\in E} y_{i,c}y_{j,c}$ and the number of cut edges ($\bar{m}_c$) as $\sum_{v_i \in V} k_i y_{i,c} - 2m_c$. Hence, modularity density D in (1) can be equivalently expressed as:

$$D = \sum_{c \in C} \left(\frac{4 \sum\limits_{\{v_i,v_j\}\in E} y_{i,c}y_{j,c} - \sum\limits_{v_i \in V} k_i y_{i,c}}{\sum\limits_{v_i \in V} y_{i,c}} \right). \tag{3}$$

Using expression (3) and according to [13], the modularity density maximization problem (MDM) can be formulated as follows:

$$\max \quad \sum_{c \in C} \left(\frac{4 \sum\limits_{\{v_i,v_j\}\in E} y_{i,c}y_{j,c} - \sum\limits_{v_i \in V} k_i y_{i,c}}{\sum\limits_{v_i \in V} y_{i,c}} \right) \tag{4}$$

$$\text{s.t.} \quad \forall c \in C, \quad 2 \leq \sum_{v_i \in V} y_{i,c} \leq |V| - 2(|C| - 1) \tag{5}$$

$$\forall v_i \in V, \quad \sum_{c \in C} y_{i,c} = 1 \tag{6}$$

$$\forall c \in C, \forall v_i \in V, \quad y_{i,c} \in \{0, 1\}, \tag{7}$$

where Constraint (5) imposes a minimum and maximum size for each cluster and Constraint (6) is used to assign each vertex to one cluster (even though extensions of clustering methods to overlapping communities have been studied in the literature [25, 26, 27]). We note that in the original formulation of

[12] Constraint (5) was actually expressed as $0 < \sum_{v_i \in V} y_{i,c} < |V|$, but it has been proved [13] that the tightest form (5) is valid.

Clearly, a source of non-linearity in MDM is due to the linear-fractional terms in the objective function. Some MILP reformulations of MDM have been proposed in [13]. However, they involve the linearization of the product between the objective function and binary variables. As explained in Section 2.1, this requires the knowledge of upper and lower bound values for both terms involved in the product. More precisely, two types of model were proposed in [13, Section 4]. One of them includes bilinear terms representing products between binary variables and the left-hand side of the objective function (3), i.e., $\frac{\sum_{\{v_i,v_j\} \in E} y_{i,c} y_{j,c}}{\sum_{v_i \in V} y_{i,c}}$. While the lower and upper bounds for the binary variable are clearly 0 and 1, respectively, those related to the latter expression are not so easy to derive. A theoretical lower bound for the latter expression was proposed in [13]. On the other hand, evaluating an upper bound is not straightforward, and involves solving the following auxiliary optimization problem, termed as A_β, defined as follow:

$$U_\beta = \max \quad \left(\frac{\sum\limits_{\{v_i,v_j\} \in E} y_i y_j}{\sum_{v_i \in V} y_i} \right) \tag{8}$$

$$\text{s.t.} \quad 2 \leq \sum_{v_i \in V} y_i \leq |V| - 2(|C| - 1) \tag{9}$$

$$\forall v_i \in V, \quad y_i \in \{0, 1\}. \tag{10}$$

Another reformulation presented in [13] involves the product between the objective function (3) and binary decision variables. Similarly, an upper bound for the objective function was required, and the corresponding auxiliary optimization problem (A_α) was defined as follows:

$$U_\alpha = \max \quad \left(\frac{4 \sum_{\{v_i,v_j\} \in E} y_i y_j - \sum_{v_i \in V} k_i y_i}{\sum_{v_i \in V} y_i} \right) \tag{11}$$

$$\text{s.t.} \quad 2 \leq \sum_{v_i \in V} y_i \leq |V| - 2(|C| - 1) \tag{12}$$

$$\forall v_i \in V, \quad y_i \in \{0, 1\}. \tag{13}$$

In [13], the continuous relaxation of A_β and A_α were solved. This because on the one hand, we need a possibly tight upper bound for the linearization of the bilinear term, but on the other hand we do not want to spend too much time to find it. We will show in Section 6 that the optimal solution of the continuous relaxation of A_β and A_α are in fact integer solutions, hence solving the continuous relaxation did not impair the tightness of the linearization of the corresponding bilinear term. However, solving A_β and A_α via their explicit MILP reformulations can improve the computational efficiency significantly. Notice that we also tried to apply the technique of Section 2.3 to reformulate directly MDM. However, preliminary tests showed that the resulting formulation was inefficient from the computational point of view, hence we did not pursue this direction.

4. MILP reformulations of A_β

In this section we derive MILP reformulations for the problems A_β defined in Section 3, based on the methods presented in Section 2. After applying the technique of Section 2.3 to problem A_β, we obtain the following formulation:

$$U_\beta = \max \quad \sum_{\{v_i,v_j\} \in E} y_i y_j h \tag{14}$$

$$\text{s.t.} \quad 2 \leq \sum_{v_i \in V} y_i \leq |V| - 2(|C| - 1) \tag{15}$$

$$\sum_{v_i \in V} y_i h = 1 \tag{16}$$

$$\forall v_i \in V, \quad y_i \in \{0, 1\}. \tag{17}$$

We present now four MILP reformulations of the above.

4.1. Formulation A^1_β

In the first model, the McCormick's (and Fortet's) inequalities are used to linearize the objective function and Constraint (16) in the following steps:

1. To linearize the objective function, we first applied Fortet's inequalities to the product $y_i y_j$ by replacing it with a new variable $s_{i,j}$. However, the lower envelopes for $s_{i,j}$ are redundant as these variables are maximized and thus, left out in the formulation of the model.
2. The resulting term $s_{i,j}h$ is then linearized by applying McCormick's inequalities and replaced with a new variable $q_{i,j}$. The lower envelopes for $q_{i,j}$ are also left out due to the same reasons stated for that of $s_{i,j}$.
3. To linearize Constraint (16), we applied the McCormick's inequalities and replaced each product $y_i h$ with a new variable r_i.

The resulting model, A^1_β, is then:

$$
\begin{aligned}
U_\beta = \max \quad & \sum_{\{v_i,v_j\}\in E} q_{i,j} \\
\text{s.t.} \quad & 2 \le \sum_{v_i\in V} y_i \le |V| - 2(|C|-1) \\
& \forall\{v_i,v_j\}\in E, \quad s_{i,j} \le y_i, \quad s_{i,j} \le y_j \\
& \forall\{v_i,v_j\}\in E, \quad q_{i,j} \le h - L_h(1-s_{i,j}), \quad q_{i,j} \le U_h s_{i,j} \\
& \forall v_i \in V, \quad L_h y_i \le r_i \le U_h y_i \\
& \forall v_i \in V, \quad h - U_h(1-y_i) \le r_i \le h - L_h(1-y_i) \\
& \sum_{v_i\in V} r_i = 1 \\
& \forall v_i \in V, \quad y_i \in \{0,1\}.
\end{aligned}
$$

4.2. Formulation A^2_β

In the second model, binary expansion and the McCormick's (and Fortet's) inequalities are used to linearize the objective function and Constraint (16) in the following steps:

1. The first step of linearizing the objective function is the same as A^1_β, which is the application of Fortet's inequalities such that $s_{i,j} = y_i y_j$.

2. Then, we applied binary expansion such that $\sum_{\{v_i,v_j\}\in E} s_{i,j} = \sum_{t\in T} 2^t a_t$. The maximum value of $\sum_{\{v_i,v_j\}\in E} s_{i,j}$ is bounded above by the number of edges (i.e. $|E|$). Therefore, we can derive that $T = \{0, \dots, \lceil \log_2(|E|+1) \rceil - 1\}$ using Formula (2).
3. The resulting term $\sum_{t\in T} 2^t a_t h$ is then linearized using the McCormick's inequalities, replacing the product $a_t h$ with a new variable q_t. Similar to Step 2 of A^2_β, the lower envelopes are not needed.
4. To linearize the constraint, we first applied binary expansion such that $\sum_{v_i\in V} y_i = \sum_{u\in U} 2^u b_u$, where $U = \{0, \dots, \lceil \log_2(|V| - 2|C| + 3) \rceil - 1\}$ (derived from Constraint (15) and Formula (2)).
5. The resulting term $\sum_{u\in U} 2^u b_u h$ is then linearized using McCormick's inequalities, replacing the product $b_u h$ by r_u.

The resulting model, A^2_β, is then:

$$
\begin{aligned}
U_\beta = \max \quad & \sum_{t\in T} 2^t q_t \\
\text{s.t.} \quad & 2 \le \sum_{v_i\in V} y_i \le |V| - 2(|C| - 1) \\
& \forall \{v_i, v_j\} \in E, \quad s_{i,j} \le y_i, \quad s_{i,j} \le y_j \\
& \sum_{\{v_i,v_j\}\in E} s_{i,j} = \sum_{t\in T} 2^t a_t \\
& \sum_{v_i\in V} y_i = \sum_{u\in U} 2^u b_u \\
& \forall t \in T, \quad q_t \le h - L_h(1 - a_t), \quad q_t \le U_h a_t \\
& \forall u \in U, \quad L_h b_u \le r_u \le U_h b_u \\
& \forall u \in U, \quad h - U_h(1 - b_u) \le r_u \le h - L_h(1 - b_u) \\
& \sum_{u\in U} 2^u r_u = 1 \\
& \forall t \in T, \quad a_t \in \{0, 1\} \\
& \forall u \in U, \quad b_u \in \{0, 1\} \\
& \forall v_i \in V, \quad y_i \in \{0, 1\}.
\end{aligned}
$$

4.3. Formulation A^3_β

In the third model, Fortet's inequalities, binary expansion and dual reformulation are used to linearize the objective function while binary expansion and dual reformulation are used to linearize Constraint (16). The steps taken are described as follows:

1. The first two steps of linearizing the objective function is the same as that of the second model, A^2_β. That is, using Fortet's inequalities to linearize the product y_iy_j, followed by binary expansion of $\sum_{\{v_i,v_j\}\in E} s_{i,j}$.
2. The term $\sum_{t\in T} 2^t a_t h$ is then linearized using dual reformulation with variables λ_A, and replacing the product $a_t h$ with the expression $(L_h\lambda_{A_{3,t}} + U_h\lambda_{A_{4,t}})$.
3. The first step of linearizing the constraint is the same as Step 4 of A^2_β, which is the binary expansion of $\sum_{v_i\in V} y_i$.
4. The resulting term $\sum_{u\in U} 2^u b_u h$ is then linearized using dual reformulation with variables λ_B and replacing the product $b_u h$ with the expression $(L_h\lambda_{B_{3,u}} + U_h\lambda_{B_{4,u}})$.

The resulting model, A^3_β, is then:

$$
\begin{aligned}
U_\beta = \max \quad & \sum_{t\in T} 2^t (L_h \lambda_{A_{3,t}} + U_h \lambda_{A_{4,t}}) \\
\text{s.t.} \quad & 2 \le \sum_{v_i \in V} y_i \le |V| - 2(|C|-1) \\
& \forall \{v_i, v_j\} \in E, \quad s_{i,j} \le y_i, \quad s_{i,j} \le y_j \\
& \sum_{\{v_i,v_j\}\in E} s_{i,j} = \sum_{t\in T} 2^t a_t \\
& \forall t \in T, \quad a_t = \lambda_{A_{3,t}} + \lambda_{A_{4,t}} \\
& \forall t \in T, \quad h = L_h(\lambda_{A_{1,t}} + \lambda_{A_{3,t}}) + U_h(\lambda_{A_{2,t}} + \lambda_{A_{4,t}}) \\
& \sum_{v_i\in V} y_i = \sum_{u\in U} 2^u b_u \\
& \forall u \in U, \quad b_u = \lambda_{B_{3,u}} + \lambda_{B_{4,u}} \\
& \forall u \in U, \quad h = L_h(\lambda_{B_{1,u}} + \lambda_{B_{3,u}}) + U_h(\lambda_{B_{2,u}} + \lambda_{B_{4,u}}) \\
& \sum_{u\in U} 2^u (L_h \lambda_{B_{3,u}} + U_h \lambda_{B_{4,u}}) = 1 \\
& \forall t \in T, \quad \sum_{l=1}^{4} \lambda_{A_{l,t}} = 1 \\
& \forall u \in U, \quad \sum_{l=1}^{4} \lambda_{B_{l,u}} = 1 \\
& \forall t \in T, \quad \forall l \in \{1,\dots,4\}, \quad \lambda_{A_{l,t}} \ge 0 \\
& \forall u \in U, \quad \forall l \in \{1,\dots,4\}, \quad \lambda_{B_{l,u}} \ge 0 \\
& \forall t \in T, \quad a_t \in \{0,1\} \\
& \forall u \in U, \quad b_u \in \{0,1\} \\
& \forall v_i \in V, \quad y_i \in \{0,1\}.
\end{aligned}
$$

4.4. Formulation A^4_β

In the fourth model, dual reformulation is used to linearize the objective function while binary expansion and dual reformulation are used to linearize Constraint (16). The steps taken are described as follows:

1. Each term $y_i y_j h$ is linearized using dual reformulation with variables

λ_Q and replaced by the expression $(L_h\lambda_{Q_{7,i,j}} + U_h\lambda_{Q_{8,i,j}})$. This is an application to trilinear terms of the technique shown in Section 2.1.2.

2. The steps taken to linearize the constraint are the same as that of A^3_β, which is the binary expansion of $\sum_{v_i\in V} y_i$ and linearization of the resulting term $b_u h$ using dual reformulation.

The resulting model, A^4_β, is then:

$$
\begin{aligned}
U_\beta = \max \quad & \sum_{\{v_i,v_j\}\in E} \left(L_h\lambda_{Q_{7,i,j}} + U_h\lambda_{Q_{8,i,j}}\right) \\
\text{s.t.} \quad & 2 \le \sum_{v_i\in V} y_i \le |V| - 2(|C|-1) \\
& \forall\{v_i,v_j\}\in E, \quad \sum_{l=1}^{8} \lambda_{Q_{l,i,j}} = 1 \\
& \forall\{v_i,v_j\}\in E, \quad y_i = \sum_{l=5}^{8} \lambda_{Q_{l,i,j}} \\
& \forall\{v_i,v_j\}\in E, \quad y_j = \lambda_{Q_{3,i,j}} + \lambda_{Q_{4,i,j}} + \lambda_{Q_{7,i,j}} + \lambda_{Q_{8,i,j}} \\
& \forall\{v_i,v_j\}\in E, \quad h = \sum_{l\in\{1,3,5,7\}} L_h\lambda_{Q_{l,i,j}} + \sum_{l\in\{2,4,6,8\}} U_h\lambda_{Q_{l,i,j}} \\
& \sum_{v_i\in V} y_i = \sum_{u\in U} 2^u b_u \\
& \forall u\in U, \quad b_u = \lambda_{B_{3,u}} + \lambda_{B_{4,u}} \\
& \forall u\in U, \quad h = L_h(\lambda_{B_{1,u}} + \lambda_{B_{3,u}}) + U_h(\lambda_{B_{2,u}} + \lambda_{B_{4,u}}) \\
& \sum_{u\in U} 2^u(L_h\lambda_{B_{3,u}} + U_h\lambda_{B_{4,u}}) = 1 \\
& \forall\{v_i,v_j\}\in E, \quad \forall l\in\{1,\ldots,8\}, \quad \lambda_{Q_{l,i,j}} \ge 0 \\
& \forall u\in U, \quad \forall l\in\{1,\ldots,4\}, \quad \lambda_{B_{l,u}} \ge 0 \\
& \forall u\in U, \quad b_u\in\{0,1\} \\
& \forall v_i\in V, \quad y_i\in\{0,1\}.
\end{aligned}
$$

5. MILP reformulations of A_α

In this section we derive MILP reformulations for the problems A_α. First, using the technique presented in Section 2.3, we obtain the following model after transforming the objective function:

$$U_\alpha = \max \quad \left(4 \sum_{\{v_i,v_j\}\in E} y_i y_j h - \sum_{v_i \in V} k_i y_i h \right) \tag{18}$$

$$\text{s.t.} \quad 2 \le \sum_{v_i\in V} y_i \le |V| - 2(|C| - 1) \tag{19}$$

$$\sum_{v_i\in V} y_i h = 1 \tag{20}$$

$$\forall v_i \in V, \quad y_i \in \{0,1\}. \tag{21}$$

It can be seen that the terms requiring linearization are in the objective function (18) and Constraint (20). The MILP reformulations of this problem are based on the best results obtained for problem A_β. More precisely, computational experiments presented in Section 7 show that the best results are obtained with models A_β^2 and A_β^3. Based on the techniques employed to derive those models, two MILP reformulations are proposed in the following subsections.

5.1. Formulation A_α^1

The first model, based on A_β^2, uses the McCormick's (and Fortet's) inequalities and binary expansion to linearize the objective function and Constraint (20). In this model, the steps described in Section 4.2 are used to linearize the terms $\sum_{\{v_i,v_j\}\in E} y_i y_j h$ and $\sum_{v_i\in V} y_i h = 1$. In addition, the following steps are taken to linearize the term $\sum_{v_i\in V} k_i y_i h$:

1. We first applied binary expansion such that $\sum_{v_i\in V} k_i y_i = \sum_{w\in W} 2^w c_w$. We can easily derive an upper bound of the term $\sum_{v_i\in V} k_i y_i$ by considering the extreme case where all the vertices belong to the cluster. This way we set all the variables y to 1, and we obtain the sum of the degrees of the vertices of the graph that is $2|E|$. We can then use this result in Formula (2) to derive that $W = \{0, \ldots, \lceil \log_2 (2|E| + 1) \rceil - 1\}$.
2. The resulting term $\sum_{w\in W} 2^w c_w h$ is then linearized using the McCormick's inequalities, replacing the product $c_w h$ with a new variable p_w. The upper envelopes are left out since we want to minimize the term (i.e., maximize it with negative sign).

The resulting model, A^1_α, is then:

$$
\begin{aligned}
U_\alpha = \max \quad & \left(4\sum_{t\in T} 2^t q_t - \sum_{w\in W} 2^w p_w\right) \\
\text{s.t.} \quad & 2 \le \sum_{v_i\in V} y_i \le |V| - 2(|C|-1) \\
& \forall \{v_i, v_j\} \in E, \quad s_{i,j} \le y_i, \quad s_{i,j} \le y_j \\
& \sum_{\{v_i,v_j\}\in E} s_{i,j} = \sum_{t\in T} 2^t a_t \\
& \forall t \in T, \quad q_t \le h - L_h(1-a_t), \quad q_t \le U_h a_t \\
& \sum_{v_i\in V} y_i = \sum_{u\in U} 2^u b_u \\
& \forall u \in U, \quad L_h b_u \le r_u \le U_h b_u \\
& \forall u \in U, \quad h - U_h(1-b_u) \le r_u \le h - L_h(1-b_u) \\
& \sum_{u\in U} 2^u r_u = 1 \\
& \sum_{v_i\in V} k_i y_i = \sum_{w\in W} 2^w c_w \\
& \forall w \in W, \quad p_w \ge h - U_h(1-c_w), \quad p_w \ge L_h c_w \\
& \forall t \in T, \quad a_t \in \{0,1\} \\
& \forall u \in U, \quad b_u \in \{0,1\} \\
& \forall w \in W, \quad c_w \in \{0,1\} \\
& \forall v_i \in V, \quad y_i \in \{0,1\}.
\end{aligned}
$$

5.2. Formulation A^2_α

The second model, based on A^3_β, uses Fortet's inequalities, binary expansion and dual reformulation to linearize the objective function. On the other hand, binary expansion and dual reformulation are used to linearize Constraint (20). In this model, the steps described in Section 4.3 are used to linearize the terms $\sum_{\{v_i,v_j\}\in E} y_i y_j h$ and $\sum_{v_i\in V} y_i h = 1$. In addition, the following steps are taken to linearize the term $\sum_{v_i\in V} k_i y_i h$:

1. The first step to linearize the term is the same as Step 1 in A^1_α. That is, to apply binary expansion such that $\sum_{v_i\in V} k_i y_i = \sum_{w\in W} 2^w c_w$.

2. The resulting term $\sum_{w \in W} 2^w c_w h$ is then linearized using dual reformulation on the product $c_w h$.

The resulting model, A^2_α, is then:

$$
\begin{aligned}
U_\alpha = \max \quad & \left(4 \sum_{t \in T} 2^t (L_h \lambda_{A_{3,t}} + U_h \lambda_{A_{4,t}}) - \sum_{w \in W} 2^w (L_h \lambda_{C_{3,w}} + U_h \lambda_{C_{4,w}}) \right) \\
\text{s.t.} \quad & \forall \{v_i, v_j\} \in E, \quad s_{i,j} \leq y_i, \quad s_{i,j} \leq y_j \\
& \sum_{\{v_i, v_j\} \in E} s_{i,j} = \sum_{t \in T} 2^t a_t \\
& \forall t \in T, \quad a_t = \lambda_{A_{3,t}} + \lambda_{A_{4,t}} \\
& \forall t \in T, \quad h = L_h (\lambda_{A_{1,t}} + \lambda_{A_{3,t}}) + U_h (\lambda_{A_{2,t}} + \lambda_{A_{4,t}}) \\
& \sum_{v_i \in V} y_i = \sum_{u \in U} 2^u b_u \\
& \forall u \in U, \quad b_u = \lambda_{B_{3,u}} + \lambda_{B_{4,u}} \\
& \forall u \in U, \quad h = L_h (\lambda_{B_{1,u}} + \lambda_{B_{3,u}}) + U_h (\lambda_{B_{2,u}} + \lambda_{B_{4,u}}) \\
& \sum_{u \in U} 2^u (L_h \lambda_{B_{3,u}} + U_h \lambda_{B_{4,u}}) = 1 \\
& \sum_{v_i \in V} k_i y_i = \sum_{w \in W} 2^w c_w \\
& \forall w \in W, \quad c_w = \lambda_{C_{3,w}} + \lambda_{C_{4,w}} \\
& \forall w \in W, \quad h = L_h (\lambda_{C_{1,w}} + \lambda_{C_{3,w}}) + U_h (\lambda_{C_{2,w}} + \lambda_{C_{4,w}}) \\
& \forall t \in T, \quad \forall l \in \{1, \ldots, 4\}, \quad \lambda_{A_{l,t}} \geq 0 \\
& \forall u \in U, \quad \forall l \in \{1, \ldots, 4\}, \quad \lambda_{B_{l,u}} \geq 0 \\
& \forall w \in W, \quad \forall l \in \{1, \ldots, 4\}, \quad \lambda_{C_{l,w}} \geq 0 \\
& \forall t \in T, \quad a_t \in \{0, 1\}, \quad \sum_{l=1}^{4} \lambda_{A_{l,t}} = 1 \\
& \forall u \in U, \quad b_u \in \{0, 1\}, \quad \sum_{l=1}^{4} \lambda_{B_{l,u}} = 1 \\
& \forall w \in W, \quad c_w \in \{0, 1\}, \quad \sum_{l=1}^{4} \lambda_{C_{l,w}} = 1 \\
& \forall v_i \in V, \quad y_i \in \{0, 1\}.
\end{aligned}
$$

6. Bound tightness of NLP continuous relaxations

In [13] the continuous relaxations of (8)-(10) (A_β) and (11)-(13) (A_α) were solved to find the bounds needed to define the McCormick's inequalities. A question that was not investigated in the previous work is how tight these bounds are. We address this question in the following. In particular, we show that the optimal solutions of A_β and A_α remain integer even when the integrality constraints are relaxed. This implies that the bounds achieved by NLP relaxations, provided that they are solved to optimality, are in fact the same as those achieved by solving the exact MILP reformulations. We emphasize, however, that solving the NLP relaxations may still be more computationally challenging compared to solving the MILP reformulations. As a matter of fact, despite the fact that the domain of the variables is relaxed, the problem is still a non-linear non-convex one. Hence, a convex non-linear solver may not achieve the global optimal solution. On the other hand, the MILP reformulation can be solved by more efficient solvers as CPLEX and the global optimality of the solution can thus be guaranteed.

Theorem 1. *Let $\mathbf{F}_\beta$ be the set of symmetric optimal solutions of the continuous relaxation of A_β (where $|\mathbf{F}_\beta| = 1$ if the global optimum is unique). Then, it is always possible to find $\mathbf{y}^* \in \mathbf{F}_\beta : \mathbf{y}^* \in \{0,1\}^{|V|}$.*

Proof. Suppose that $\exists \mathbf{y}^* \in \mathbf{F}_\beta$ such that $\exists v_p \in V : Y_p^* \in (0,1)$, and the corresponding optimal solution value is U_β^*. Starting from $\mathbf{y}^*$, consider the following two alternative solutions: in the first one we set $y_p^* = 0$, and the corresponding objective value is U_β^-. In the second one we set $y_p^* = 1$, and the corresponding objective value is U_β^+. We show that either $U_\beta^+ = U_\beta^* = U_\beta^-$ or at least one among U_β^+ and U_β^- is larger than U_β^*. In other words, we can always find an integer solution at least as good as the fractional one. Note that the integer solutions obtained this way are feasible with respect to Constraint (9). When there are at least two clusters the original constraint proposed in [12] was more loose, i.e., $0 < \sum_{v_i \in V} y_i < |V|$, but it was shown in [13] that the integer optimal solution respect Constraint (9). As the need arises, let us define the following terms:

$$\eta = \sum_{\{v_i, v_j\} \in E} y_i^* y_j^* \tag{22}$$

$$\xi = \sum_{v_i \in V} y_i^* \tag{23}$$

$$\delta = \sum_{\{v_p, v_j\} \in E} y_j^*. \tag{24}$$

We can express U_β^*, U_β^+, and U_β^- using the notation introduced above respectively as:

$$U_\beta^* = \frac{\eta}{\xi} \tag{25}$$

$$U_\beta^+ = \frac{\eta + (1 - y_p^*)\delta}{\xi + 1 - y_p^*} \tag{26}$$

$$U_\beta^- = \frac{\eta - y_p^* \delta}{\xi - y_p^*}. \tag{27}$$

We can now write the difference in terms of objective function value corresponding to setting the variable y_p^* to 0 and 1, respectively, as follows:

$$\Delta_\beta^+ = U_\beta^+ - U_\beta^* = \frac{\eta + (1 - y_p^*)\delta}{\xi + 1 - y_p^*} - \frac{\eta}{\xi} = \frac{(1 - y_p^*)(\xi\delta - \eta)}{\xi(\xi + 1 - y_p^*)} \tag{28}$$

$$\Delta_\beta^- = U_\beta^- - U_\beta^* = \frac{\eta - y_p^*\delta}{\xi - y_p^*} - \frac{\eta}{\xi} = \frac{-y_p^*(\xi\delta - \eta)}{\xi(\xi - y_p^*)}. \tag{29}$$

As a consequence of Constraint (9) we have that $\xi \geq 2$. Combining this with the condition $y_p^* \in (0, 1)$ we can conclude that the denominators of both Δ_β^- and Δ_β^+ are strictly greater than 0. We have now two cases.

1. If $\delta = \frac{\eta}{\xi}$ then $\Delta_\beta^- = \Delta_\beta^+ = 0$. Hence, we can set y_p^* to 0 or 1 and we find an integer solution having cost U_β^*.
2. If $\delta \neq \frac{\eta}{\xi}$ we need to show that it is impossible to have both Δ_β^+ and Δ_β^- negative. If $\Delta_\beta^+ < 0$ then we have $(1 - y_p^*)(\xi\delta - \eta) < 0$ (since the denominator is strictly greater than 0). Similarly, if $\Delta_\beta^- < 0$ we have $-y_p^*(\xi\delta - \eta) < 0$. This means that $-(1 - y_p^*)y_p^* > 0$, that is impossible if $y_p^* \in (0, 1)$. Hence, at least one of the solutions obtained by setting y_p^* to 0 or 1 is better than the fractional solution.

It can be easily verified that the same reasoning can be extended to the case where more than one variable is fractional in the assumed optimal solution. □

Theorem 2. *Let* $\mathbf{F}_\alpha$ *be the set of symmetric optimal solutions of the continuous relaxation of* A_α *(where* $|\mathbf{F}_\alpha| = 1$ *if the global optimum is unique). Then, it is always possible to find* $\mathbf{y}^* \in \mathbf{F}_\alpha : \mathbf{y}^* \in \{0,1\}^{|V|}$.

Proof. This can be proved similarly to Theorem 1. Starting again from a fractional solution $\mathbf{y}^*$, let us define the following terms:

$$\eta = 4 \sum_{\{v_i,v_j\}\in E} y_i^* y_j^* - \sum_{v_i \in V} k_i y_i^* \tag{30}$$

$$\xi = \sum_{v_i \in V} y_i^* \tag{31}$$

$$\gamma = 4 \sum_{\{v_p,v_j\}\in E} y_j^*, \tag{32}$$

and the objective function values:

$$U_\alpha^* = \frac{\eta}{\xi} \tag{33}$$

$$U_\alpha^+ = \frac{\eta + (1-y_p^*)\gamma - (1-y_p^*)k_p}{\xi + 1 - y_p^*} \tag{34}$$

$$U_\alpha^- = \frac{\eta - y_p^*\gamma + y_p^* k_p}{\xi - y_p^*}. \tag{35}$$

By defining $\delta = \gamma - k_p$ we can prove the theorem as for Theorem 1. □

7. Computational experiments

In this section we present the results obtained by solving some problem instances of the literature using the MILP reformulations presented in Sections 4 and 5. The computational experiments were carried out using a workstation with 4 Intel Xeon E5-4620 CPU at 2.20 gigahertz (8 cores each, Hyper Threading and Turbo Boost disabled), 128 gigabytes RAM (32 gigabytes for each processor) running Linux. The same workstation and instances, whose details are reported in Table 1, were also used in [13]. We can, therefore, make a fair comparison with the results obtained by solving the continuous relaxation of A_β and A_α in [13]. Note, however, that the continuous relaxation of A_β and A_α, being non-convex and non-linear, were solved using the MINLP solver SCIP [28], while the corresponding MILP reformulations were solved using the solver CPLEX [29].

Graph	$\|V\|$	$\|E\|$
strike [30]	24	38
Galesburg F [31]	31	63
Galesburg D [31]	31	67
karate [32]	34	78
Korea1 [33]	35	69
Korea2 [33]	35	84
Mexico [34]	35	117
sawmill [35]	36	62
dolphins small [36]	40	70
journal index [37]	40	189

Table 1: Details of the instances used for the computational experiments.

Table 2 shows the time needed to solve the MILP reformulations of problem A_β, compared with the continuous relaxation of A_β ($c(A_\beta)$). If the time needed exceeds five minutes, the experiment is stopped and recorded as *t.l.* (i.e., time limit) in the table. From the table, we can see that the models A^2_β and A^3_β appear more computationally efficient than the other models. Recall that models A^2_β and A^3_β are obtained by utilizing binary decomposition, in conjunction with McCormick's inequalities or dual reformulation. MILP reformulations are then proposed based on these two models, as mentioned previously. The improvement in solution time over $c(A_\beta)$ is especially significant in the problem based on the graph "journal index". In fact, the results indicate that the computational time improvement becomes remarkable when the problem size increases.

Table 3 shows the time needed to solve the MILP reformulations of A_α, compared with the continuous relaxation of A_α ($c(A_\alpha)$). From the table, we can see that both models A^1_α and A^2_α are more efficient than $c(A_\alpha)$, with the exception of the problem based on the smallest instance *strike*. Overall, A^1_α seems to be more efficient than A^2_α. A^1_α is the MILP reformulation of A_α

Graph	Time (seconds)				
	$c(A_\beta)$	A^1_β	A^2_β	A^3_β	A^4_β
strike	0.81	0.91	0.75	**0.64**	0.95
Galesburg F	1.68	2.12	0.53	**0.42**	1.98
Galesburg D	1.49	2.06	1.27	**0.43**	2.16
karate	1.93	2.70	**1.00**	1.06	2.91
Korea1	1.76	1.91	**0.37**	0.72	1.62
Korea2	1.78	2.58	0.53	**0.46**	1.83
Mexico	4.25	16.95	0.69	**0.53**	83.06
sawmill	3.64	2.50	1.05	**0.73**	5.10
dolphins small	6.41	4.65	1.06	**1.04**	11.36
journal index	122.0	*t.l.*	**0.41**	0.57	*t.l.*

Table 2: Results obtained for MILP reformulations of A_β, using the solver CPLEX. Shortest time taken to solve for each graph is bolded. Time recorded as *t.l.* when time limit of 5 minutes is reached. Results for $c(A_\beta)$ are taken from [13].

obtained by utilizing binary decomposition and McCormick's inequalities.

It is important to point out that the computational times involved in solving these auxiliary problems can indeed become significant. First, as shown in [13], for large instances the time needed to solve those auxiliary problems can be more than the time needed to solve the MILP reformulation of MDM. Moreover, for large instances we may need to execute these problems repeatedly for several times when we try different number of clusters.

8. Conclusions

In this paper we have provided a complete MILP characterization of the modularity density maximization problem by reformulating the 0-1 NLP auxiliary problems of [13]. To do so we have combined and compared different reformulation techniques and identified the most effective ones. Computational experiments show that solving those MILP reformulations can be up to two order of magnitude faster than solving the continuous relaxation of the 0-1 NLP. We have also proved that the optimal solution of the continuous relaxation of the 0-1 NLP provides an integer solution, thus answering to an

Graph	Time (seconds)		
	$c(A_\alpha)$	A^1_α	A^2_α
strike	**0.19**	0.48	0.83
Galesburg F	2.41	0.65	**0.55**
Galesburg D	1.71	**0.37**	0.89
karate	3.43	0.81	**0.36**
Korea1	1.37	**0.36**	0.82
Korea2	1.07	**0.35**	0.67
Mexico	55.87	**0.39**	0.48
sawmill	5.84	**0.37**	0.57
dolphins small	1.98	**0.38**	0.62
journal index	25.08	**0.59**	0.67

Table 3: Results obtained for MILP reformulations of A_α, using the solver CPLEX. Shortest time taken to solve for each graph is bolded. Results for $c(A_\alpha)$ are taken from [13].

open question from [13]. The future work has two main directions: first, to use the results obtained in order to improve other mathematical programming based approaches to modularity density maximization, for example the heuristic proposed in [38] and the SDP relaxation of [39]. Second, to develop an heuristic algorithm based on the resolution of the MILP reformulations of the auxiliary problems, since they can be solved efficiently.

Acknowledgements

This research was partially conducted at the Future Resilient Systems at the Singapore-ETH Centre (SEC). The SEC was established as a collaboration between ETH Zurich and National Research Foundation (NRF) Singapore (FI 370074011) under the auspices of the NRF's Campus for Research Excellence and Technological Enterprise (CREATE) programme.

References

[1] M. Girvan, M. E. J. Newman, Community structure in social and biological networks, Proceedings of the National Academy of Sciences of the USA 99 (12) (2002) 7821–7826.

[2] G. Adomavicius, A. Tuzhilin, Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions, IEEE Transactions on Knowledge and Data Engineering 17 (6) (2005) 734–749.

[3] R. Guimerà, L. A. N. Amaral, Functional cartography of complex metabolic networks, Nature 433 (2005) 895–900.

[4] G. Palla, I. Dernyi, I. Farkas, T. Vicsek, Uncovering the overlapping community structure of complex networks in nature and society, Nature 435 (7043) (2005) 814–818.

[5] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, D. Parisi, Defining and identifying communities in networks, Proceedings of the National Academy of Sciences of the USA 101 (9) (2004) 2658–2663.

[6] Y. Hu, H. Chen, P. Zhang, M. Li, Z. Di, Y. Fan, Comparative definition of community and corresponding identifying algorithm, Physical Review E 78 (2) (2008) 026121.

[7] S. Cafieri, G. Caporossi, P. Hansen, S. Perron, A. Costa, Finding communities in networks in the strong and almost-strong sense, Physical Review E 85 (4) (2012) 046113.

[8] M. Newman, M. E. J. Girvan, Finding and evaluating community structure in networks, Physical Review E 69 (2) (2004) 026113.

[9] U. Brandes, D. Delling, M. Gaertler, R. Görke, M. Hoefer, Z. Nikoloski, D. Wagner, On modularity clustering, IEEE Transactions on Knowledge and Data Engineering 20 (2) (2008) 172–188.

[10] S. Fortunato, M. Barthélemy, Resolution limit in community detection, Proceedings of the National Academy of Sciences of the USA 104 (1) (2007) 36–41.

[11] B. H. Good, Y.-A. de Montjoye, A. Clauset, Performance of modularity maximization in practical contexts, Physical Review E 81 (2010) 046106.

[12] Z. Li, S. Zhang, R.-S. Wang, X.-S. Zhang, L. Chen, Quantitative function for community detection, Physical Review E 77 (3) (2008) 036109.

[13] A. Costa, MILP formulations for the modularity density maximization problem, European Journal of Operational Research 245 (1) (2015) 14 – 21.

[14] A. Costa, Some remarks on modularity density, Tech. Rep. 1409.4063, arXiv (2014).

[15] A. Costa, L. Liberti, Relaxations of multilinear convex envelopes: Dual is better than primal, in: R. Klasing (Ed.), Experimental Algorithms, Vol. 7276 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2012, pp. 87–98.

[16] A. Rikun, A convex envelope formula for multilinear functions, Journal of Global Optimization 10 (4) (1997) 425–437.

[17] F. Tardella, Existence and sum decomposition of vertex polyhedral convex envelopes, Optimization Letters 2 (2008) 363–375.

[18] G. P. McCormick, Computability of global solutions to factorable nonconvex programs: Part I — Convex underestimating problems, Mathematical Programming 10 (1976) 146–175.

[19] R. Fortet, Applications de l'algèbre de Boole en recherche opérationelle, Revue Française de Recherche Opérationelle 4 (14) (1960) 17–26.

[20] C. A. Meyer, C. A. Floudas, Frontiers in Global Optimization, Springer US, 2004, Ch. Trilinear Monomials with Positive or Negative Domains: Facets of the Convex and Concave Envelopes, pp. 327–352.

[21] C. A. Meyer, C. A. Floudas, Trilinear monomials with mixed sign domains: Facets of the convex and concave envelopes, Journal of Global Optimization 29 (2) 125–155.

[22] S. Balram, Crude transshipment via floating, production, storage and offloading platforms, Master's thesis, Dept. of Chemical and Biomolecular Engineering, National University of Singapore, Singapore (2010).

[23] A. Costa, P. Hansen, L. Liberti, On the impact of symmetry-breaking constraints on spatial branch-and-bound for circle packing in a square, Discrete Applied Mathematics 161 (1) (2013) 96–106.

[24] A. Charnes, W. W. Cooper, Programming with linear fractional functionals, Naval Research Logistics Quarterly 10 (1) (1963) 273–274.

[25] J. Xie, S. Kelley, B. K. Szymanski, Overlapping community detection in networks: The state-of-the-art and comparative study, ACM Computing Surveys 45 (4) (2013) 43:1–43:35.

[26] M. R. Fellows, J. Guo, C. Komusiewicz, R. Niedermeier, J. Uhlmann, Graph-based data clustering with overlaps, Discrete Optimization 8 (1) (2011) 2 – 17.

[27] A. Lancichinetti, S. Fortunato, J. Kertsz, Detecting the overlapping and hierarchical community structure in complex networks, New Journal of Physics 11 (3) (2009) 033015.

[28] T. Achterberg, SCIP: Solving constraint integer programs, Mathematical Programming Computation 1 (1) (2009) 1–41.

[29] IBM, ILOG CPLEX 12.6 User's Manual, IBM (2013).

[30] J. H. Michael, Labor dispute reconciliation in a forest products manufacturing facility, Forest Products Journal 47 (11/12) (1997) 41.

[31] J. S. Coleman, E. Katz, H. Menzel, et al., Medical innovation: A diffusion study, Bobbs-Merrill Indianapolis, 1966.

[32] W. W. Zachary, An information flow model for conflict and fission in small groups, Journal of Anthropological Research (1977) 452–473.

[33] E. M. Rogers, D. L. Kincaid, Communication networks: toward a new paradigm for research.

[34] J. Gil-Mendieta, S. Schmidt, The political network in Mexico, Social Networks 18 (4) (1996) 355 – 381.

[35] J. H. Michael, J. G. Massey, Modeling the communication network in a sawmill, Forest Products Journal 47 (9) (1997) 25.

[36] D. Lusseau, K. Schneider, O. J. Boisseau, P. Haase, E. Slooten, S. M. Dawson, The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations, Behavioral Ecology and Sociobiology 54 (4) (2003) 396–405.

[37] M. Rosvall, C. T. Bergstrom, An information-theoretic framework for resolving community structure in complex networks, Proceedings of the National Academy of Sciences 104 (18) (2007) 7327–7331.

[38] A. Costa, S. Kushnarev, L. Liberti, Z. Sun, Divisive heuristic for modularity density maximization, Computers & Operations Research 71 (2016) 100 – 109.

[39] Y. Izunaga, T. Matsui, Y. Yamamoto, A doubly nonnegative relaxation for modularity density maximization, Tech. Rep. 2016-03-5368, Optimization Online (2016).