Proceedings of the 16th Conference on Computer Science and Intelligence Systems pp. 329–335
DOI: 10.15439/2021F65
ISSN 2300-5963 ACSIS, Vol. 25

Discovering Communities in Networks: A Linear Programming Approach Using Max-Min Modularity

Arman Ferdowsi
Vienna University of Technology
Institute of Computer Engineering
Embedded Computing Systems
Email: aferdowsi@ecs.tuwien.ac.at

Alireza Khanteymoori
University of Freiburg
Department of Computer Science
Bioinformatics Group
Email: alireza@informatik.uni-freiburg.de

***Abstract*—Community detection is a fundamental challenge in network science and graph theory that aims to reveal nodes' structures. While most methods consider Modularity as a community quality measure, Max-Min Modularity improves the accuracy of the measure by penalizing the Modularity quantity when unrelated nodes are in the same community. In this paper, we propose a community detection approach based on linear programming using Max-Min Modularity. The experimental results show that our algorithm has a better performance than the previously known algorithms on some well-known instances.**

I. Introduction

IN MANY (complex) networks, there are sets of nodes with some common characteristics. More specifically, there are sets of highly interactive vertices that are likely to yield and share common relationships and properties among themselves. These sets are called *communities*. Detecting communities has become one of the fundamental subjects in the field of network science and graph theory and has numerous applications in a wide range of areas, including the analysis of Social Network [1], [2], Biological Networks [3], Cosmological Networks [4], and WEB [5]. It also plays a crucial role in the domain of Signal Processing [6], Image Segmentation [7], Pattern Recognition [8], and Data Clustering [9].

A network is basically given as a graph $G = (V, E)$ with the set of vertices V and edges E. A *community* in the network can then be contemplated as a subset of vertices $C \subseteq V$ with a high density of edges between nodes inside the subset and a low density of edges connecting this subset to the others. Accordingly, one can define the community detection problem as *partitioning* V into a set of disjoint communities $\mathbf{C} = \{C_1, C_2, \ldots, C_k\}$. In the literature, several quality measures can be used to qualify the goodness of a partitioning. One of the most widely used and well-known quality measures is *Modularity*, introduced by Newman [10]: Let $A = (a_{i,j})$ be the adjacency matrix of G, where $a_{i,j}$ is one when there is an edge between node i and node j, and zero otherwise; $d_i = \Sigma_{l=1}^{n} a_{i,l}$ be the degree of node i; m be the number of edges and n be the number of vertices in G. Modularity Q of a given partitioning $\mathbf{C}$ is defined as:

$$Q(\mathbf{C}) = \frac{1}{2m} \sum_{i,j \in V} [a_{i,j} - \frac{d_i d_j}{2m}] \sigma(i,j) \qquad (1)$$

where $\sigma(i,j)$ is one if i and j are in the same community and zero otherwise.

Intuitively, for a community C, Modularity is the number of edges within C minus the expected number of such edges. So, the high-quality communities can be determined as the ones with the high value of Modularity. We refer to the problem of finding a partition of the network that maximizes Modularity as the *Modularity Maximization* problem. The Modularity Maximization problem is NP-hard [11]. Nevertheless, many algorithms, both heuristics (e.g., [12], [13], [14], [15], [16]) and exact methods (e.g., [11], [17], [18]) have been proposed to solve this problem (approximately).

It is known that the Modularity measure suffers from some limitations (see [19], [20] for more details). In particular, as pointed out in [21] and [22], one of the major limitations of Modularity is that it only takes the existing edges of the network into consideration. In other words, Modularity qualifies the goodness of the discovered communities by only measuring how good the partitioning fits the existing edges. This is indeed a drawback because Modularity does not consider the disconnected nodes (absent edges) that lie in the same community. *Max-Min Modularity* [21] is one of the successful extensions of Modularity which improves the accuracy of the measure by penalizing the Modularity quantity when disconnected nodes are in the same community. More precisely, it is assumed in [21] that (in addition to the graph G) a zero-one *relation matrix* $U = (u_{i,j})$ is given that defines whether every pair of disconnected nodes of the network is related or not; where $u_{i,j}$ is one when disconnected nodes i and j are *related*, and zero otherwise. They, in fact, take into account the importance of the *indirect* connections between disconnected nodes by only penalizing the Modularity measure when *unrelated* nodes are in the same community: Consider a complemented graph $G' = (V, E')$, where E' contains an edge between every pair of disconnected nodes of G that is unrelated; i.e., there is an edge between i and j in G' if there is not such an edge in G and also $u_{i,j}$ is zero. Let $A' = (a'_{i,j})$ be the adjacency matrix of G' and d'_i be the degree of node i in G' accordingly. Let m' be the number of the edges in G'. Max-Min Modularity Q_{MM} of a given partition $\mathbf{C}$ of V is

defined as follows:

$$Q_{MM}(\mathbf{C}) = \sum_{i,j \in V} [\frac{1}{2m}(a_{i,j} - \frac{d_i d_j}{2m}) - \frac{1}{2m'}(a'_{i,j} - \frac{d'_i d'_j}{2m'})]\sigma(i,j) \quad (2)$$

We refer to the problem of finding a partition of the network that maximizes Max-Min Modularity as the *Max-Min Modularity Maximization* problem. Chen et al. [21] proposed a hierarchical clustering algorithm (similar to that of Newman [10] for the classical Modularity Maximization Problem) that approximately optimizes Max-Min Modularity in a greedy manner.

A drawback of the approach described in [21] is that it strongly depends on the accuracy of the given relation matrix. So the quantity of the measure might be heavily affected by the node relationships defined by the user in the first place. Therefore, unobserved or misobserved relations between nodes of the network can lead to poor partitioning results. It is worth mentioning that authors of [21] also suggested a systematic (but not necessarily accurate) way for defining the relation matrix U: Two disconnected nodes are related if they connect to the same intermediary node; this is, for every node pairs i and j, $u_{i,j}$ is one only if $\{i,j\} \notin E$ and there is some node k that $\{i,k\} \in E$ and $\{k,j\} \in E$, and zero otherwise.

Main contribution: We develop the first LP-based approach for solving the Max-Min Modularity Maximization problem. First, we provide a more accurate way of defining the relation matrix by exploiting an optimal linear relaxation solution to the standard integer linear programming of the Modularity Maximization problem. After that, we depict the standard integer programming formulation of the Max-Min Modularity Maximization problem. Then, for solving the problem, we employ a row and column generation approach to efficiently solve the linear programming relaxation the problem. This provides an optimal fractional solution to the Max-Min Modularity Maximization problem. Next, we design a new rounding algorithm to obtain integer solutions and, therefore, to determine the community structures. We finally present a computational study of our algorithm on known instances. The computational experiments show that our results highly resemble the optimal solutions and that our algorithm outperforms the previous well-known algorithms, including the algorithm proposed in [21].

The paper is organized as follows: the rest of this section focuses on providing a brief literature review. In Section II, we first introduce the novel relation matrix, and then we model the Max-Min Modularity Maximization problem based on that. Next, in Section III, we depict the row/column generation technique and also the local search-based rounding algorithm. Section IV is then dedicated to the experimental results.

A. Related Works

In the literature, several approaches are proposed to detect communities in the networks: extremal optimization [23], spectral optimization [24], greedy heuristics [25], [26], simulated annealing [27], dynamical clustering [28], deep learning techniques [29], message passing [30], quantum mechanics [31], and more.

Despite a considerable amount of work on the community detection problem, relatively little work solves the problem using linear programming or integer programming techniques. In 2008, Agarwal and Kempe [32] expressed the Modularity Maximization problem as a standard Integer Programming (IP) model and proposed an LP rounding algorithm for the problem. Although the LP relaxation of their model can be solved in polynomial time, as the number of constraints in their model is $O(n^3)$, the rounding algorithm becomes impractical when the number of nodes is large. Consequently, in 2010, a column generation technique is developed in [17] to solve the model more efficiently . Nevertheless, the proposed algorithm could not solve problems with more than a few hundred nodes in a reasonable time. In 2011, Dinh and Thai [33] proposed a sparse LP formulation for the problem with much fewer constraints than that of [32]. Finally, in 2013, Miyamoto [34] proposed a row and column generation approach to solve the sparse LP formulation, resulting in an efficient algorithm for obtaining the optimal value of the sparse LP relaxation (and so an upper bound for the optimal value for the Modularity Maximization problem).

II. Model Description

Let the binary variable x_{ij} indicate if nodes i and j belong to the same community or not; the value of x_{ij} is zero if nodes i and j belong to the same community, and one otherwise. Let $I_{all} = \{(i,j) \in V^2 \mid i < j\}$; and $q_{ij} = a_{i,j} - \frac{d_i d_j}{2m}$, for each $(i,j) \in I_{all}$. As described in [33], the Modularity Maximization problem can be formulated in terms of the following integer linear program.

$$\max \quad \frac{1}{m} \sum_{(i,j) \in I_{all}} q_{ij}(1 - x_{ij}) \qquad \text{(IP-M)}$$

$$x_{ij} + x_{jk} - x_{ik} \geq 0 \qquad \forall i < j < k \qquad (3)$$

$$x_{ij} - x_{jk} + x_{ik} \geq 0 \qquad \forall i < j < k \qquad (4)$$

$$-x_{ij} + x_{jk} + x_{ik} \geq 0 \qquad \forall i < j < k \qquad (5)$$

$$x_{ij} \in \{0,1\} \qquad \forall (i,j) \in I_{all} \qquad (6)$$

Constraints (3)-(5) guarantee that if i and j are in the same community and j and k are in the same community, then so are i and k. We refer to the relaxation of (IP-M), obtained by replacing the constraints $x_{ij} \in \{0,1\}$ by $x_{ij} \in [0,1]$, as (LP-M).

A. Computing the Relation Matrix via LP

In this section, we provide a systematic and accurate way for defining the relation matrix by exploiting an optimal solution to (LP-M). Let x^* be the optimal solution to (LP-M). This can be obtained efficiently (in polynomial time) using, for example, the row and column generation algorithm of [34]. We note that the optimal fractional solution x^* induces a metric, called *the LP distance*, on the graph G: think of x^*_{ij} as a "*distance*" between nodes i and j. Observe that Constraints

(3)-(5) guarantee the *triangle inequality* for any $i, j, k \in V$ in the induced metric. Clearly, the larger the LP distance of two nodes is, the less related the nodes are. This observation and also the fact that the Modularity Maximization problem can be nicely proposed for weighted graphs [35] motivates us to define the relation matrix and so the complemented (weighted) graph G' using the LP distance (rather than the graph distance). Recall that Chen et al. [21] defined the relation matrix using the distance between nodes in the graph G: Two disconnected nodes are related if they connect to the same intermediary node (if the distance between them in G is two).

We define the relation matrix matrix $A' = (a'_{i,j})$ (and hence G'; $(a'_{i,j})$ represents the weight of the edge between nodes i and j in G') as follows:

$$a'_{i,j} = \begin{cases} x^*_{ij} & \text{if } a_{i,j} = 0 \text{ and } j > i \\ x^*_{ji} & \text{if } a_{i,j} = 0 \text{ and } i > j \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

Before any further discussion, it is worth pointing out that only by replacing our proposed relation matrix with the one used in [21] and then applying their hierarchical algorithm we can attain more accurate results. Fig. 1 proves this claim in the following way. It considers 12 well-known networks whose optimal communities (*ground truth*) are already known and valid. It then provides a comparison between each network's ground truth and the communities discovered by the Max-Min Modularity method with respect to i) the conventional relation matrix U, proposed in [21], (red diagram), and ii) our proposed relation matrix (gray diagram). Section IV describes the networks used and the performance metric *Normalized Mutual Information (NMI)*. However, for now, note that for a given network with known community assignments and a given community detection algorithm, the more the NMI value, which can vary between 0 and 1, the more similarity there is between the discovered communities and the ground truth. The results clearly illustrate that applying the proposed relation matrix leads to more accurate communities, which are more similar to the ground truth.

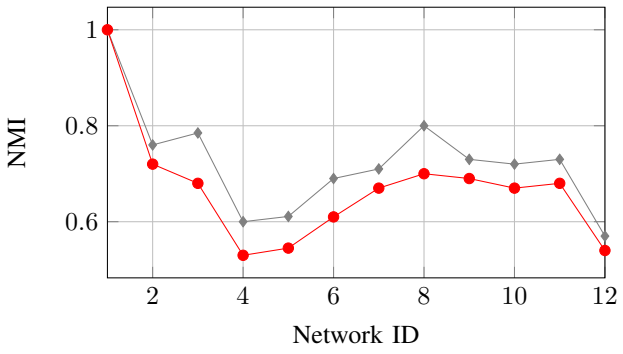

Fig. 1: Comparison between NMI values, for twelve well-known real-world networks, achieved by (i) red curve: Max-Min modularity, proposed in [21] and (ii) gray curve: using our proposed relation matrix but the hierarchical algorithm proposed in [21].

Considerably attractive is that one can still significantly improve the results by solving the standard formulation of the Max-Min Modularity Maximization problem, which will be explained in the following sub-section.

B. Modeling the Max-Min Modularity Maximization problem

First of all, note that it is not difficult to check that the redundant constraints introduced in [36] for the clique partitioning problem are also redundant for the standard formulation of the Modularity Maximization problem and the problem of Max-Min Modularity Maximization. Accordingly, for a given matrix $A' = (a'_{i,j})$, resp. the weighted graph G', defined above, the Max-Min Modularity Maximization problem can be formulated as the following IP. Let $c_{ij} = \frac{q_{ij}}{m} - \frac{q'_{ij}}{m'}$, where $q'_{ij} = a'_{i,j} - \frac{d'_i d'_j}{2m'}$, $d'_i = \sum_{l=1}^{n} a'_{i,l}$, and $m' = \sum_{(i,j)\in I_{all}} a'_{i,j}$; for each $(i,j) \in I_{all}$.

$$\max \sum_{(i,j)\in I_{all}} c_{ij}(1 - x_{ij}) \quad \text{(IP-MM)}$$

$$x_{ij} + x_{jk} - x_{ik} \geq 0 \qquad \forall i < j < k,\ c_{ij} \geq 0 \vee c_{jk} \geq 0 \quad (8)$$

$$x_{ij} - x_{jk} + x_{ik} \geq 0 \qquad \forall i < j < k,\ c_{ij} \geq 0 \vee c_{ik} \geq 0 \quad (9)$$

$$-x_{ij} + x_{jk} + x_{ik} \geq 0 \qquad \forall i < j < k,\ c_{jk} \geq 0 \vee c_{ik} \geq 0 \quad (10)$$

$$x_{ij} \in \{0, 1\} \qquad \forall (i,j) \in I_{all} \quad (11)$$

We refer to the relaxation of (IP-MM), obtained by replacing the constraints $x_{ij} \in \{0, 1\}$ by $x_{ij} \in [0, 1]$, as (LP-MM).

III. Solution Approach

To solve (IP-MM), we first employ a technique to find the optimal solution to (LP-MM) efficiently. Then we propose a local search-based rounding procedure to obtaining the integer solution.

A. Solving (LP-MM)

While (LP-MM) can be solved in polynomial time, it would be deficient for networks exceeding a few hundred vertices since the number of constraints is $3\binom{n}{3} = O(n^3)$, and therefore, rapidly grows with respect to the number of nodes. To tackle this difficulty, we introduce a row/column generation technique heavily inspired by the one proposed in [34] for the Modularity Maximization problem. Let $I = \{(i,j) \in I_{all} \mid c_{ij} > 0\}$ and $I' = \{(i,j) \in I_{all} \mid c_{ij} \leq 0\}$ be two sets of vertex pairs indices in I_{all}. For a given $\mathcal{I} \subseteq I'$, the following formulation presents a sub-problem of (LP-MM) consisting of all pairs in I and some pairs in I'.

$$\max \sum_{(i,j)\in I} c_{ij}(1 - x_{ij}) + \sum_{(i,j)\in \mathcal{I}} c_{ij}(1 - x_{ij}) \quad \text{(LPs-MM}(\mathcal{I})\text{)}$$

$$x_{ij} + x_{jk} - x_{ik} \geq 0 \qquad \forall (i,j),(j,k),(i,k) \in I \cup \mathcal{I}, c_{ij} \geq 0 \vee c_{jk} \geq 0 \quad (12)$$

$$x_{ij} - x_{jk} + x_{ik} \geq 0 \qquad \forall (i,j),(j,k),(i,k) \in I \cup \mathcal{I}, c_{ij} \geq 0 \vee c_{ik} \geq 0 \quad (13)$$

$$-x_{ij} + x_{jk} + x_{ik} \geq 0 \qquad \forall (i,j),(j,k),(i,k) \in I \cup \mathcal{I}, c_{jk} \geq 0 \vee c_{ik} \geq 0 \quad (14)$$

$$x_{ij} \in [0, 1] \qquad \forall\, i < j,\ (i,j),(j,k),(i,k) \in I \cup \mathcal{I} \quad (15)$$

It can be easily turned out that (LPs-MM($\emptyset$)) is the smallest formulation and (LPs-MM(I')) is equivalent to (LP-MM) itself. Please note that, since for all $(i,j) \in \mathcal{I} \subseteq I'$ we have $c_{ij} \leq 0$, (LPs-MM($\mathcal{I}$)) clearly provides an upper bound of the optimal value of (LP-MM), and moreover, adding variables

can never worsen the upper bound. Furthermore, the following theorem brings forward a condition under which the upper bound is equal to the optimal value of (LP-MM).

Theorem 3.1: If an optimal solution $\bar{x^*} = (\bar{x^*_{ij}})_{(i,j)\in I\cup\mathcal{I}}$ to (LPs-MM($\mathcal{I}$)) satisfies the condition $(*)$, then $(x^*_{ij})_{(i,j)\in I_{all}}$ is an optimal solution to (LP-MM), where

$$x^*_{ij} = \begin{cases} \bar{x^*_{ij}} & ; \ (i,j) \in I \cup \mathcal{I} \\ 1 & ; \ \text{otherwise} \end{cases} \tag{16}$$

and

$$(*) \begin{cases} \bar{x^*_{ij}} + \bar{x^*_{jk}} \geq 1; \ (i,j),(j,k) \in I \cup \mathcal{I}, c_{ij} \geq 0 \vee c_{jk} \geq 0, (i,k) \in I^{'} - \mathcal{I} \\ \bar{x^*_{ij}} + \bar{x^*_{ik}} \geq 1; \ (i,j),(i,k) \in I \cup \mathcal{I}, c_{ij} \geq 0 \vee c_{ik} \geq 0, (j,k) \in I^{'} - \mathcal{I} \\ \bar{x^*_{jk}} + \bar{x^*_{ik}} \geq 1; \ (j,k),(i,k) \in I \cup \mathcal{I}, c_{jk} \geq 0 \vee c_{ik} \geq 0, (i,j) \in I^{'} - \mathcal{I} \end{cases}$$

Proof. Suppose that $\bar{x^*} = (\bar{x^*_{ij}})_{(i,j)\in I\cup\mathcal{I}}$ is an optimal solution to (LPs-MM($\mathcal{I}$)) that satisfies the condition $(*)$. Let $x^* = (x^*_{ij})_{(i,j)\in I_{all}}$, such that for every $(i,j) \in I_{all}$, (x^*_{ij}) is defined by Equation (16). We indicate that x^* is an optimal solution to (LP-MM). First of all, x^* is feasible for (LP-MM). To prove that, we suffice to confirm that the first set of constraints of (LP-MM) (eq. (8)) is satisfied. The same argument can be expressed for the remaining two sets of constraints. It thereby needs to be determined that for all $i < j < k$ such that $c_{ij} \geq 0$ or $c_{jk} \geq 0$, we have $x^*_{ij} + x^*_{jk} - x^*_{ik} \geq 0$. Note that eight conditions may happen to the pairs (i,j), (j,k), and (i,k). If (i,j), (j,k), $(i,k) \in I \cup \mathcal{I}$, the constraints are satisfied because they are also in (LPs-MM($\mathcal{I}$)). If (i,j), $(j,k) \in I \cup \mathcal{I}$ and $(i,k) \in I' - \mathcal{I}$, the constraints are again satisfied due to the condition $(*)$. Furthermore, in the remaining cases, at least one of x^*_{ij} or x^*_{jk} equals 1, so the constraints are again satisfied. Hence, x^* is feasible for (LP-MM). As a result, it is enough to show that the objective value of x^* in (LP-MM) is equal to that of $\bar{x^*}$ in (LPs-MM($\mathcal{I}$)). Point out that one can rewrite the objective function of (LP-MM) as follows:

$$\underbrace{\sum_{(i,j)\in I} c_{ij}(1 - x_{ij}) + \sum_{(i,j)\in\mathcal{I}} c_{ij}(1 - x_{ij})}_{F} + \underbrace{\sum_{(i,j)\in I_{all}-I-\mathcal{I}} c_{ij}(1 - x_{ij})}_{Z}.$$

F is exactly the objective value of (LPs-MM($\mathcal{I}$)), and Z equals 0 according to Equation (16). Therefore, the objective value of x^* in (LP-MM) is equal to that of $\bar{x^*}$ in (LPs-MM($\mathcal{I}$)). □

Based on the above discussion, we state the following scheme for obtaining the optimal solution to (LP-MM).

- Start solving (LPs-MM($\mathcal{I}$)) with $\mathcal{I} = \emptyset$ and adding those $x_{ij} \in I' - \mathcal{I}$ that violate inequalities in $(*)$ in each iteration, until an optimal solution to (LPs-MM($\mathcal{I}$)) satisfies $(*)$.
- Employing a row generation for solving (LPs-MM($\mathcal{I}$)) in each repeat.

B. Rounding algorithm

Recall from what we discussed in Section II-A that a solution x^* to (LP-MM) expresses the *LP distance* such that the lower the x^*_{ij}, the more tendency the nodes i and j have to be in a same community. Our local search-based procedure rounds the distance between vertices (or, as we will see, move the vertices among communities) based on simultaneously using the LP distance and the value of (IP-MM) [1]. Assume that $x^* = (x^*_{ij})_{(i,j)\in V^2}$ is an optimal fractional solution to (LP-MM). We denote each $C \subset V$ a *community* if we have $x^*_{ij} = 0$ for every $i, j \in C$. Further, by *assigning* node i we mean to round down x^*_{ij} to 0 for every $j \in C$ and round it up to 1; otherwise, where C is the community whose *center node* (explained in the next paragraph) has the minimum distance from i (w.r.t the LP distance).

The main idea of the local search-based rounding procedure is to obtain the best communities leading to the maximum possible value of (IP-MM) by wisely assigning nodes. Intuitively, the algorithm starts from an initial solution, which is the set of communities achieved by the optimal solution x^* to (LP-MM), and then iteratively moves to the neighbor solutions. In brief, it starts with randomly associating a *center node* for each community and then assigning each node j, which is not a member of any community. Next, it computes the value of (IP-MM)[2]. Afterward, it iteratively greedily improves the center vertices based on one of the three functions *Add*, *Delete*, and *Swap* at a time. Then it updates communities by reassigning every node. To be more precise, in each repeat, *Add* and *Delete* functions respectively check whether adding or deleting a center node (and therefore, the corresponding community) can make any progress in the value of (IP-MM) and if that so, the best action leading to this improvement will be recorded. On the other hand, the function *Swap* tries to discover the best switch between a non-center and a center node that leads to the maximum improvement in (IP-MM) value. At last, the best function leading to the best gain in the value of (IP-MM) will be selected, and in this way, communities will be updated. The above procedure will be repeated until the best possible community structures regarding the obtained value of (IP-MM) are found. We note that in the case that solving (LP-MM) does not lead to obtaining any communities at the very beginning (i.e., $x^*_{ij} \neq 0$ for every $i, j \in V$ such that $i \neq j$), the algorithm randomly chooses a number $k \in \{1, 2, \ldots, n\}$ of nodes as center vertices and assigns each of the remaining vertices. Algorithm 1 elaborates the pseudo-code of this technique.

IV. Computational results

In this section, we present a performance evaluation for our proposed method by using 12 commonly-used and well-known real-world networks that are listed in Table I. Ground truth (i.e., the optimal community structures) is available and known for each of these networks, and therefore, one can facilely measure the quality of a community detection algorithm by estimating the similarities between the communities obtained by the algorithm and the ground truth. For doing this, we use the well-known performance metric NMI.

A. Normalized Mutual Information (NMI)

NMI [50] is indeed a well-known clustering comparison metric. Nevertheless, it can perfectly evaluate the similarity

[1]By a *value of* (*IP-MM*), we mean the value of the objective function of (IP-MM) with respect to an integer solution.

[2]Note that, after assigning all vertices, we have integer solution.

Algorithm 1: Local search-based rounding procedure.

Input: $x^* = (x^*_{ij})_{(i,j)\in V^2}$ *// an optimal solution to (LP-MM).*
Output: set of communities $\mathcal{C}$ of the network G.
let $T = \{T_1, T_2, \ldots, T_k\}$ be the set of k initial communities obtained by x^*;
if $|T| \neq \emptyset$ **then**
  let $S = \{\mu_1, \mu_2, \ldots, \mu_k\}$ such that μ_i is a randomly selected member of T_i, for all $i \in \{1, 2, \ldots, k\}$;
else
  let $S = \{\mu_1, \mu_2, \ldots, \mu_k\}$ be a set of k randomly chosen vertices, for a random $k \in \{2, \ldots, n\}$;
$(\mathcal{C}, Q) \leftarrow$*CalculateGain(S)*;
$Q_{temp} \leftarrow 0$;
while $Q > (1+\epsilon)Q_{temp}$ **do**
  // small constant ϵ guarantees that running time remains polynomial. See [37], [38].
  $Q_{temp} \leftarrow Q$;
  $(\mathcal{C}, Q, S) \leftarrow BestMove(S)$;
Return $(\mathcal{C})$;
– –
// Functions declaration:
CalculateGain (S)
  let $C_i = \{\mu_i\}$, for every $\mu_i \in S$ and $1 \leq i \leq |S|$;
  for *every* $i \in V - S$ **do**
    assign i; (i.e., $C_j \leftarrow C_j \cup \{i\}$ where $j = argmin\{x^*_{i\mu_j} : 1 \leq j \leq |S|\}$)
  $\mathcal{C} \leftarrow \{C_1, C_2, \ldots, C_k\}$;
  $Q \leftarrow$ the value of (IP-MM) w.r.t the set of communities $\mathcal{C}$;
  Return $(\mathcal{C}, Q)$;
BestMove (S)
  $(S^{add}, \mathcal{C}^{add}, Q^{add}) \leftarrow$ *Add(S)*;
  $(S^{delete}, \mathcal{C}^{delete}, Q^{delete}) \leftarrow$ *Delete(S)*;
  $(S^{swap}, \mathcal{C}^{swap}, Q^{swap}) \leftarrow$ *Swap(S)*;
  retrieve the highest (IP-MM) value Q, the best set of communities $\mathcal{C}$, and the best set of center nodes S;
  Return $(\mathcal{C}, Q, S)$;
Add (S)
  for *every* $i \in V - S$ **do**
    $S^{add} \leftarrow S \cup \{i\}$;
    $(\mathcal{C}^{add}, Q^{add}) \leftarrow$ *CalculateGain*(S^{add});
    remember current S^{add}, $\mathcal{C}^{add}$, and Q^{add};
  Return $(S^{add}, \mathcal{C}^{add}, Q^{add})$ corresponding to the highest obtained Q^{add};
Delete (S)
  for *every* $i \in S$ **do**
    $S^{delete} \leftarrow S - \{i\}$;
    $(\mathcal{C}^{delete}, Q^{delete}) \leftarrow$ *CalculateGain*(S^{delete});
    remember current S^{delete}, $\mathcal{C}^{delete}$, and Q^{delete};
  Return $(S^{delete}, \mathcal{C}^{delete}, Q^{delete})$ corresponding to the highest obtained Q^{delete};
Swap (S)
  for *every* $i \in S$ **do**
    for *every* $j \in V - S$ **do**
      $S^{swap} \leftarrow (S - \{i\}) \cup \{j\}$;
      $(\mathcal{C}^{swap}, Q^{swap}) \leftarrow$ *CalculateGain*(S^{swap});
      remember current S^{swap}, $\mathcal{C}^{swap}$, and Q^{swap};
  Return $(S^{swap}, \mathcal{C}^{swap}, Q^{swap})$ corresponding to the highest Q^{swap};

between the optimal communities and those discovered by an algorithm. Suppose that for a given network G, $\mathcal{C}(\mathcal{A}) = \{C_1, \ldots, C_k\}$ and $\mathcal{C}' = \{C'_1, \ldots, C'_{k'}\}$ be respectively a set of communities obtained by an algorithm $\mathcal{A}$ and the ground truth. The NMI value corresponding to the algorithm $\mathcal{A}$ can be written as

$$NMI = \frac{-2\sum_{x=1}^{|\mathcal{C}|}\sum_{y=1}^{|\mathcal{C}'|}\frac{|C_x \cap C'_y|}{n} log(\frac{n|C_x \cap C'_y|)}{|C_x||C'_y|}}{\sum_{x=1}^{|\mathcal{C}|}\frac{C_x}{n} log(\frac{C_x}{n}) + \sum_{y=1}^{|\mathcal{C}'|}\frac{C'_y}{n} log(\frac{C'_y}{n})} \quad (17)$$

TABLE I: Networks under-study

ID	Network	n	m
1	Zachary's karate club [39]	34	78
2	Mexican Politicians [40]	35	117
3	Dolphin network [41]	62	159
4	Les Miserables [41]	77	254
5	p53 protein [42]	104	226
6	Books about U.S. politics [43]	105	441
7	American college football [44]	115	613
8	Citation graph drawing [45]	311	640
9	USAir97 [46]	332	2126
10	C. Elegans [47]	453	2025
11	Erdos collaboration [48]	472	1314
12	Electronic circuit [49]	512	819

In the case where the detected communities are identical to the ground truth, the NMI takes its maximum value one, while in the case where the two sets totally disagree, the NMI score is zero. Generally, the more the NMI, the better community structures have been found.

B. Experiments

In what follows, we provide a complete evaluation that shows how our relation matrix or/and rounding technique can individually resp. together improve the old-fashion relation matrix or/and other rounding procedures and the conventional Max-Min Modularity algorithm.

All tests are conducted on a computer system with a processor Intel(R) Core(TM) i5-7300U CPU @ 2.60GHz, 2712 Mhz, 2 Core(s), 4 Logical Processor(s), 8 GB of Rams, and Win10 OS. Algorithms are implemented with C++, and CPLEX optimizer 12.9 is used for solving linear programming.

Fig. 2 provides a comprehensive comparison by evaluating communities that are discovered based on the following cases:

- Our proposed method (the blue diagram): Using the relation matrix, proposed in Section II-A, to model the Max-Min Modularity Maximization problem, solving (LP-MM) via the row/column generation method introduced in Section III-A, and detecting communities (obtaining integer solutions) by the devised rounding technique (Section III-B).
- Replacing the relation matrix suggested in [21] with our relation matrix but using the hierarchical algorithm proposed in [21] (the gray diagram).
- Applying the relation matrix introduced in [21] to model the Max-Min Modularity Maximization problem and using our proposed rounding procedure to obtain communities (the yellow diagram).
- Using our relation matrix and row/column generation technique to solve (LP-MM), but employing the rounding algorithm proposed by Agarwal and Kempe [32] [3] instead of our rounding procedure; (The black diagram).

[3]The authors of [32] introduced a rounding procedure to obtain the integer solution to the Modularity Maximization problem. Their method is actually derived from a rounding procedure that is originally proposed for the correlation clustering problem. However, it led to raising unwarranted singleton and a number of low-quality communities that made them apply a series of Kernighan-Lin shifts [51] to improve community structures. Here we used their technique to rounds (LP-MM) solution.

- Applying our rounding method to the optimal solution to the linear programming relaxation of the Modularity Maximization problem obtained in [34] (green diagram).
- Max-Min Modularity, proposed in [21] (red diagram).

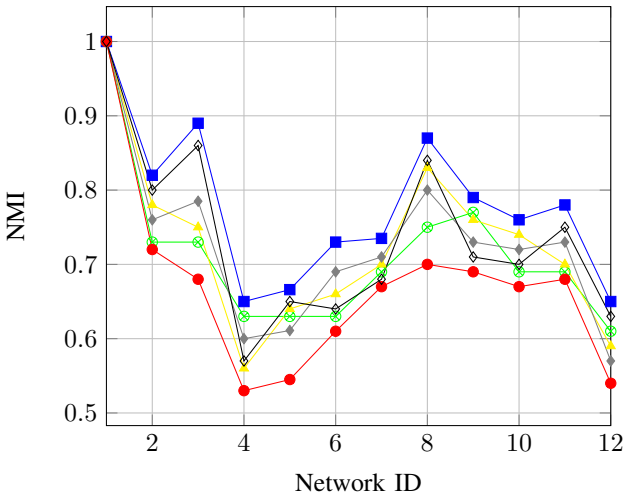

Fig. 2: Comparison between NMI values achieved by (i) blue curve: our method, (ii) gray curve: using our proposed relation matrix but the old-fashion algorithm, (iii) yellow curve: using the old-fashion relation matrix, but applying our proposed rounding procedure, (iv) green curve: applying our rounding method to the optimal solution to the linear programming relaxation of the Modularity Maximization problem, (v) black curve: applying our relation matrix but using another rounding algorithm to find communities. and (vi) red curve: conventional Max-Min modularity.

One can obviously conclude that the best results are achieved when the proposed method, including the new relations matrix and also the devised rounding algorithm, is used. In particular, comparing the blue and green diagrams shows the advantage of solving (linear relaxation) of the Max-Min Modularity Maximization problem rather than solving the (linear relaxation) of the Modularity Maximization problem. On the other hand, the worst result occurs when we just use the Max-Min Modularity proposed in [21]. So, an immediate consequence might be that while the idea behind the Max-Min Modularity is so clever and interesting, the relation matrix and also the hierarchical algorithm introduced in [21] do not lead to a very high-quality result.

As we already mentioned in Section II-A, comparing the gray and red diagrams can show us the superiority of using our proposed relation matrix instead of the one introduce in [21]. On the other hand, by considering the blue and black diagrams, one can recognize the preponderance of the proposed rounding algorithm over the famous rounding procedure suggested in [32]. A final remark might be that although the proposed relation matrix and the developed rounding technique alone improves the results, the high efficiency of the method considerably relies on their simultaneous application. It means that, for example, applying our proposed rounding algorithm but using the traditional relation matrix cannot always lead the promising results; See the yellow diagram.

V. Conclusion

In this work, we first introduced a systematic way to generate a more accurate relation matrix for the Max-Min Modularity Maximization problem based on the optimal solution to the linear relaxation programming of the Modularity Maximization problem. After that, according to this new relation matrix, we modeled the standard integer formulation for the Max-Min Modularity Maximization problem and employed a row/column generation technique to solve its linear relaxation version. We also devised a local search-based rounding method that facilitates us to round fractional solutions to integer ones and detect communities of a network in a very accurate way. The proposed computational experiments showed that our results highly resemble the optimal solutions and that our algorithm outperforms the previous well-known algorithms.

References

[1] L. Jiang, L. Shi, L. Liu, J. Yao, and M. A. Yousuf, "User interest community detection on social media using collaborative filtering," *Wireless Networks*, pp. 1–7, 2019.

[2] M. Wang, C. Wang, J. X. Yu, and J. Zhang, "Community detection in social networks: an in-depth benchmarking study with a procedure-oriented framework," *Proceedings of the VLDB Endowment*, vol. 8, no. 10, pp. 998–1009, 2015.

[3] Y. Atay, I. Koc, I. Babaoglu, and H. Kodaz, "Community detection from biological and social networks: A comparative analysis of metaheuristic algorithms," *Applied Soft Computing*, vol. 50, pp. 194–211, 2017.

[4] D. Krioukov, M. Kitsak, R. S. Sinkovits, D. Rideout, D. Meyer, and M. Boguñá, "Network cosmology," *Scientific reports*, vol. 2, p. 793, 2012.

[5] S. Aparicio, J. Villazón-Terrazas, and G. Álvarez, "A model for scale-free networks: application to twitter," *Entropy*, vol. 17, no. 8, pp. 5848–5867, 2015.

[6] N. Tremblay and P. Borgnat, "Graph wavelets for multiscale community mining," *IEEE Transactions on Signal Processing*, vol. 62, no. 20, pp. 5227–5239, 2014.

[7] O. A. Linares, G. M. Botelho, F. A. Rodrigues, and J. B. Neto, "Segmentation of large images based on super-pixels and community detection in graphs," *IET Image Processing*, vol. 11, no. 12, pp. 1219–1228, 2017.

[8] L. M. Freitas and M. G. Carneiro, "Community detection to invariant pattern clustering in images," in *2019 8th Brazilian Conference on Intelligent Systems (BRACIS)*. IEEE, 2019, pp. 610–615.

[9] F. D. Malliaros and M. Vazirgiannis, "Clustering and community detection in directed networks: A survey," *Physics Reports*, vol. 533, no. 4, pp. 95–142, 2013.

[10] M. E. Newman, "Modularity and community structure in networks," *Proceedings of the national academy of sciences*, vol. 103, no. 23, pp. 8577–8582, 2006.

[11] U. Brandes, D. Delling, M. Gaertler, R. Gorke, M. Hoefer, Z. Nikoloski, and D. Wagner, "On modularity clustering," *IEEE transactions on knowledge and data engineering*, vol. 20, no. 2, pp. 172–188, 2007.

[12] P. Schuetz and A. Caflisch, "Efficient modularity optimization by multistep greedy algorithm and vertex mover refinement," *Physical Review E*, vol. 77, no. 4, p. 046112, 2008.

[13] S. Cafieri, A. Costa, and P. Hansen, "Reformulation of a model for hierarchical divisive graph modularity maximization," *Annals of Operations Research*, vol. 222, no. 1, pp. 213–226, 2014.

[14] B. Rajita and S. Panda, "Community detection techniques for evolving social networks," in *2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*. IEEE, 2019, pp. 681–686.

[15] A. Ferdowsi and A. Abhari, "Generating high-quality synthetic graphs for community detection in social networks," in *2020 Spring Simulation Conference (SpringSim)*. IEEE, 2020, pp. 1–10.
[16] R. Guimera and L. A. N. Amaral, "Functional cartography of complex metabolic networks," *nature*, vol. 433, no. 7028, pp. 895–900, 2005.
[17] D. Aloise, S. Cafieri, G. Caporossi, P. Hansen, S. Perron, and L. Liberti, "Column generation algorithms for exact modularity maximization in networks," *Physical Review E*, vol. 82, no. 4, p. 046112, 2010.
[18] G. Xu, S. Tsoka, and L. G. Papageorgiou, "Finding community structures in complex networks using mixed integer optimisation," *The European Physical Journal B*, vol. 60, no. 2, pp. 231–239, 2007.
[19] S. Fortunato and M. Barthelemy, "Resolution limit in community detection," *Proceedings of the national academy of sciences*, vol. 104, no. 1, pp. 36–41, 2007.
[20] T. Chakraborty, A. Dalmia, A. Mukherjee, and N. Ganguly, "Metrics for community analysis: A survey," *ACM Computing Surveys (CSUR)*, vol. 50, no. 4, pp. 1–37, 2017.
[21] J. Chen, O. R. Zaïane, and R. Goebel, "Detecting communities in social networks using max-min modularity," in *Proceedings of the 2009 SIAM international conference on data mining*. SIAM, 2009, pp. 978–989.
[22] J. Scripps, P.-N. Tan, and A.-H. Esfahanian, "Exploration of link structure and community-based node roles in network analysis," in *Seventh IEEE international conference on data mining (ICDM 2007)*. IEEE, 2007, pp. 649–654.
[23] J. Duch and A. Arenas, "Community detection in complex networks using extremal optimization," *Physical review E*, vol. 72, no. 2, p. 027104, 2005.
[24] T. Richardson, P. J. Mucha, and M. A. Porter, "Spectral tripartitioning of networks," *Physical Review E*, vol. 80, no. 3, p. 036111, 2009.
[25] M. E. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical review E*, vol. 69, no. 2, p. 026113, 2004.
[26] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of statistical mechanics: theory and experiment*, vol. 2008, no. 10, p. P10008, 2008.
[27] R. Guimera and L. A. N. Amaral, "Cartography of complex networks: modules and universal roles," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2005, no. 02, p. P02001, 2005.
[28] S. Boccaletti, M. Ivanchenko, V. Latora, A. Pluchino, and A. Rapisarda, "Detecting complex network modularity by dynamical clustering," *Physical Review E*, vol. 75, no. 4, p. 045102, 2007.
[29] A. Tsitsulin, J. Palowitch, B. Perozzi, and E. Müller, "Graph clustering with graph neural networks," *arXiv preprint arXiv:2006.16904*, 2020.
[30] C. Shi, Y. Liu, and P. Zhang, "Weighted community detection and data clustering using message passing," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2018, no. 3, p. 033405, 2018.
[31] Y. Q. Niu, B. Q. Hu, W. Zhang, and M. Wang, "Detecting the community structure in complex networks based on quantum mechanics," *Physica A: Statistical Mechanics and Its Applications*, vol. 387, no. 24, pp. 6215–6224, 2008.
[32] G. Agarwal and D. Kempe, "Modularity-maximizing graph communities via mathematical programming," *The European Physical Journal B*, vol. 66, no. 3, pp. 409–418, 2008.
[33] T. N. Dinh and M. T. Thai, "Finding community structure with performance guarantees in complex networks," *arXiv preprint arXiv:1108.4034*, 2011.
[34] A. Miyauchi and Y. Miyamoto, "Computing an upper bound of modularity," *The European Physical Journal B*, vol. 86, no. 7, p. 302, 2013.
[35] M. E. Newman, "Analysis of weighted networks," *Physical review E*, vol. 70, no. 5, p. 056131, 2004.
[36] A. Miyauchi and N. Sukegawa, "Redundant constraints in the standard formulation for the clique partitioning problem," *Optimization Letters*, vol. 9, no. 1, pp. 199–207, 2015.
[37] V. Arya, N. Garg, R. Khandekar, A. Meyerson, K. Munagala, and V. Pandit, "Local search heuristics for k-median and facility location problems," *SIAM Journal on computing*, vol. 33, no. 3, pp. 544–562, 2004.
[38] A. Gupta and K. Tangwongsan, "Simpler analyses of local search algorithms for facility location," *arXiv preprint arXiv:0809.2554*, 2008.
[39] W. W. Zachary, "An information flow model for conflict and fission in small groups," *Journal of anthropological research*, vol. 33, no. 4, pp. 452–473, 1977.
[40] J. Gil-Mendieta and S. Schmidt, "The political network in mexico," *Social Networks*, vol. 18, no. 4, pp. 355–381, 1996.
[41] D. Lusseau, K. Schneider, O. J. Boisseau, P. Haase, E. Slooten, and S. M. Dawson, "The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations," *Behavioral Ecology and Sociobiology*, vol. 54, no. 4, pp. 396–405, 2003.
[42] R. Guimera, L. Danon, A. Diaz-Guilera, F. Giralt, and A. Arenas, "Self-similar community structure in a network of human interactions," *Physical review E*, vol. 68, no. 6, p. 065103, 2003.
[43] A. Mahajan and M. Kaur, "Various approaches of community detection in complex networks: a glance," *International Journal of Information Technology and Computer Science (IJITCS)*, vol. 8, no. 35, 2016.
[44] M. Girvan and M. E. Newman, "Community structure in social and biological networks," *Proceedings of the national academy of sciences*, vol. 99, no. 12, pp. 7821–7826, 2002.
[45] N. Meghanathan, "A greedy algorithm for neighborhood overlap-based community detection," *Algorithms*, vol. 9, no. 1, p. 8, 2016.
[46] V. Batagelj and A. Mrvar, "Pajek datasets (2006)," 2009.
[47] A. Cangelosi and D. Parisi, "A neural network model of caenorhabditis elegans: the circuit of touch sensitivity," *Neural processing letters*, vol. 6, no. 3, pp. 91–98, 1997.
[48] V. Batagelj and A. Mrvar, "Pajek." 2014.
[49] S. Chand and S. Mehta, "Community detection using nature inspired algorithm," in *Hybrid Intelligence for Social Networks*. Springer, 2017, pp. 47–76.
[50] L. Danon, A. Diaz-Guilera, J. Duch, and A. Arenas, "Comparing community structure identification," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2005, no. 09, p. P09008, 2005.
[51] B. W. Kernighan and S. Lin, "An efficient heuristic procedure for partitioning graphs," *The Bell system technical journal*, vol. 49, no. 2, pp. 291–307, 1970.