Proceedings of the 18th Conference on Computer Science and Intelligence Systems pp. 111–117

DOI: 10.15439/2023F2832
ISSN 2300-5963 ACSIS, Vol. 35

Toward an Optimal Solution to the Network Partitioning Problem

Arman Ferdowsi *
K. N. Toosi University of Technology
Department of Mathematics
Email: armanferdowsi@email.kntu.ac.ir

Maryam Dehghan Chenary
University of Vienna
Institute of Business Decisions and Analytics
Email: maryam.dehghan.chenary@univie.ac.at

***Abstract*—This paper delves into the realm of community detection in network science and graph theory with the overarching objective of unraveling the underlying structures between nodes within a network. In this pursuit, we put forth a novel and comprehensive approach to ascertain the optimal solution to maximizing the renowned community quality metric known as Max-Min Modularity. Through a series of experiments encompassing diverse case studies, we substantiate the efficacy and validity of our proposed approach, further bolstering its credibility.**

***Index Terms*—Graph partitioning, community detection, mathematical programming, exact method**

I. Introduction and Related Work

In COMPLEX networks, nodes usually divide into several subsets sharing common characteristics and relationships, forming *communities*. The discovery and analysis of these structures hold paramount importance in computer science, particularly within the network domain and graph theory. Identifying cohesive and interconnected groups of nodes enables a deeper understanding of complex systems, facilitating insights into structural patterns, functional modules, and underlying relationships. The ability to detect network communities owes immense value and applications in networked systems, such as Social Network Analysis [1], [2], Biological Networks [3], Cosmological Networks [4], WEB analysis [5], Distributed Computing [6], Signal Processing [7], and Data Clustering [8]. It enables us to uncover hierarchical structures, predict missing links, and enhance network resilience.

More concretely, a network can be represented as a graph $G = (V, E)$, where V is the set of vertices and E is the set of edges. A community within the network can be seen as a subset of vertices $C \subseteq V$, characterized by a dense connection of edges among the nodes within the subset and a sparse connection of edges with other subsets; see Fig. 1. In this regard, the community detection problem can be defined as partitioning V into a set of communities $\mathbf{C} = \{C_1, C_2, \ldots, C_k\}$ that often entails optimizing a specific *quality measure* that quantifies the excellence of a community. A wide array of quality measures has been proposed in the literature, encompassing both *connectivity-based* and *topology-based* metrics [9].

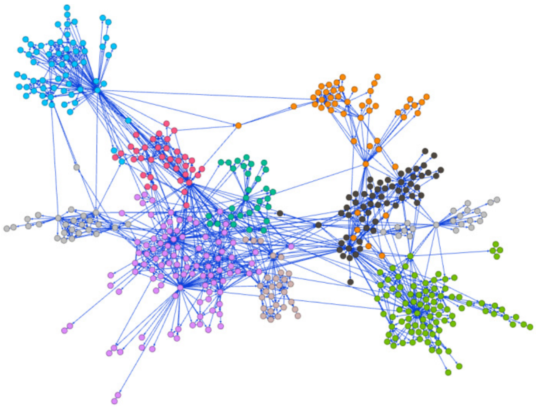


Fig. 1: Illustration of a network and its communities as subsets of vertices with densely connected nodes within each subset and sparser connections to nodes outside the batch.

For example, in [10], a dynamic connectivity-based metric is introduced to assess the quality of a community C by computing the ratio between the sum of the radius of C and the number of edges exiting C, divided by the number of edges with both endpoints belonging to C. The radius, a measure showcasing the size and compactness of the function, plays a crucial role in this evaluation. The authors of [10] also devised a two-stage heuristic algorithm to identify high-quality communities by minimizing the proposed metric. The initial stage of the algorithm, which is particularly crucial for our research, intelligently identifies an initial set of remarkably high-quality communities. This will be followed by a revising phase aimed at refining and enhancing the quality of the communities. The contributions in [10] properly highlight the significance of connectivity-based metrics in assessing community quality.

On the other hand, topological metrics have also gained considerable attention in the field of community detection. Notably, *Modularity*, introduced by Newman [11], stands out as one of the most widely recognized and extensively utilized measures in this regard. For a network G containing n vertices and m edges, the Modularity (Q) of a given partitioning $\mathbf{C}$ is mathematically defined as follows:

This research was supported by the Austrian Science Fund (FWF) project DMAC (grant no. P32431).

* Also affiliated with TU Wien, Embedded Computing Systems Group. *Email: aferdowsi@ecs.tuwien.ac.at*.

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

where $A = (a_{ij})$ is the adjacency matrix of G with a_{ij} sets to one if an edge exists between node i and node j, and zero otherwise. d_i represents the degree of node i and is defined as the sum of all entries in the i-th row of the adjacency matrix. Moreover, $\sigma(i,j)$ is one if i and j are in the same community and zero otherwise.

Simply put, Modularity quantifies the number of edges within a community minus the expected number of such edges leading to the fact that communities with higher Modularity values have better quality. Therefore, maximizing Modularity results in identifying high-quality communities within a network.

Nevertheless, despite the widespread use of Modularity, it has been known to have certain limitations (see [12], [9] for more details). Notably, Modularity only takes into account the existing edges of the network, meaning it solely evaluates the goodness of a community based on its fit with the observed edges, while it fails to consider disconnected nodes (absent edges) within the same community. This is indeed a drawback since the disconnection of nodes does not inherently imply an absence of underlying relations between them. To overcome this limitation, an extension of Modularity called *Max-Min Modularity* [13] has been developed, which improves the accuracy of the measure by penalizing Modularity when disconnected nodes are present in the same community. In Max-Min Modularity, an additional zero-one relation matrix $U = (u_{ij})$ is introduced, which defines the relationship between pairs of disconnected nodes in the network. The value of u_{ij} is one if disconnected nodes i and j are *related* and zero otherwise. This extension acknowledges the significance of indirect connections between disconnected nodes by penalizing the Modularity measure only when unrelated nodes coexist within a community. In a more abstract sense, consider a complemented graph $G' = (V, E')$, where E' contains an edge between every pair of disconnected nodes in G that are unrelated. In other words, an edge exists between nodes i and j in G' if there is no such edge in G, and u_{ij} is zero. Let $A' = (a'_{ij})$ be the adjacency matrix of G', and d'_i be the degree of node i in G'. Additionally, let m' be the number of edges in G'. The Max-Min Modularity (Q_{MM}) of a given partition $\mathbf{C}$ of V is defined as follows:

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

We refer to the problem of partitioning a network with respect to maximizing the Max-Min Modularity as the *Max-Min Modularity Maximization* problem.

Chen et al. [13] proposed a hierarchical clustering algorithm, similar to what Newman [11] had offered for the classical Modularity Maximization Problem, that approximately greedily optimizes Max-Min Modularity. In addition to the suboptimal precision of the final community detection results obtained through the heuristic approach, the primary drawback of their method lies in its reliance on a user-defined relation matrix rather than a systematic approach. This dependency on subjective input introduces the potential for misinterpretations, biases, and erroneous human decisions, which can lead to significant issues. Relying on a user-defined matrix not only increases the likelihood of errors but also lacks the robustness and objectivity provided by a systematic and automated procedure.

Furthermore, since it is known that solving the Max-Min Modularity Maximization problem is computationally challenging, as it is proven to be NP-hard, most of the approaches for solving it rely on heuristic methods. [13]. There exist, of course, methods focusing more on exact techniques, such as the approach presented in [14], in which the authors successfully formulated the Max-Min Modularity Maximization problem as an *integer programming* model and proposed an equivalent sub-problem that simplified the overall formulation. This streamlined approach facilitated the efficient solving of the model's linear relaxation and provided a systematic means of defining the relation matrix. They further employed a local search strategy to convert the fractional solutions to integer ones that led to obtaining a set of communities. Nevertheless, despite their outstanding breakthrough in the modeling and the solution approach, the error caused by the rounding approach prevents us from obtaining an optimal solution, which is indeed crucial in scenarios where precise analysis is required. This underscores the need for alternative methods that can provide more accurate results.

Main contribution:

(1) Building upon the integer programming modeling discussed in [14], we present an alternative integer model for the Max-Min Modularity Maximization problem that offers a significant reduction in the number of variables and constraints. This streamlined model enhances computational efficiency while maintaining the same optimization capabilities. The equivalence of the proposed model and the original formulation proved in Theorem 1, affirms that both models yield the same set of optimal solutions.

(2) Inspired by the prominent algorithm proposed in [10], we devise a methodology to generate an initial feasible solution for the formulated model. By employing a row generation technique in combination with the branch and bound method and leveraging the powerful CPLEX[1] solver, we efficiently and optimally solve the model. This comprehensive approach enables the detection of a set of (near) optimal communities whose efficacy and effectiveness become apparent in the experiments done.

The structure of this paper unfolds as follows: In Section II, we delve into modeling an effective integer programming formulation for the Max-Min Modularity Maximization problem, followed by an insightful theoretical exploration of how

[1] CPLEX is a powerful optimization software package developed by IBM that employs advanced algorithms to solve linear programming, mixed-integer programming, and quadratic programming problems efficiently.

to simplify the model. Subsequently, Section III outlines our devised approach for acquiring an intelligent initial feasible solution and employing a row/column generation technique to solve the model optimally. Finally, Section IV is dedicated to showcasing the experimental results, providing a comprehensive analysis of the outcomes.

II. Mathematical modeling

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 $q_{ij} = a_{ij} - \frac{d_i d_j}{2m}$, for each $(i,j) \in I_{all}$. As described in [15], 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).

Now to turn our attention to the Max-Min Modularity Maximization problem, we first recap the systematic and precise approach provided in [14] for defining the relation matrix governed by an optimal fractional solution x^* to the linear programming problem (LP-M). Considerably crucial is that x^* can be efficiently obtained in polynomial time using various algorithms such as the row and column generation algorithm introduced by [16]. It is also important to note that x^* gives rise to a metric known as the LP distance on the graph G. In this context, x^*_{ij} can be interpreted as the "distance" between nodes i and j, and notably, the constraints (3)-(5) guarantee the fulfillment of the triangle inequality for any nodes $i, j, k \in V$ in the induced metric. Evidently, the larger the LP distance between two nodes, the less related those nodes are. This observation, along with the fact that the Modularity Maximization problem can be effectively formulated for weighted graphs as demonstrated by [17], serves as motivation to define the relation matrix and the corresponding complemented (weighted) graph G' utilizing the LP distance rather than using user knowledge.

In this framework, we define the relation matrix $A' = (a'_{ij})$ (and consequently G', with (a'_{ij}) representing the weight of the edge between nodes i and j in G'), as follows:

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

Consequently, given a matrix $A' = (a'_{ij})$, the Max-Min Modularity Maximization problem can be formulated as the following integer programming problem. Let $c_{ij} = \frac{q_{ij}}{m} - \frac{q'_{ij}}{m'}$, where $q'_{ij} = a'_{ij} - \frac{d'_i d'_j}{2m'}$, $d'_i = \sum_{l=1}^{n} a'_{il}$, and $m' = \sum_{(i,j)\in I_{all}} a'_{ij}$ for each $(i,j) \in I_{all}$.

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

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

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

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

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

The very first thing to note, however, is that solving (IP-MM) falls in the class of NP-hard problems, making it challenging to be optimally solved. Consequently, we came to investigate whether it is possible to simplify (IP-MM) while maintaining the same set of optimal solutions. To achieve this objective, we have obtained the subsequent conceptual insights derived from the notion of row generation: The following lemma demonstrates that the optimal solution remains unaffected when only focusing on variables x_{ij} with $c_{ij} > 0$, and the subsequent theorem establishes that the optimal solution to (IPs-MM) without constraints involving x_{ij} where $c_{ij} \leq 0$ is equivalent to the optimal solution to (IP-MM).

Lemma 1: If a binary variable x_{ij} satisfies the constraints (8), (9), and (10), then it is sufficient to consider only the variables x_{ij} for which $c_{ij} > 0$ in the objective function (IP-MM).

Proof. Suppose we have a binary variable x_{ij} that satisfies the constraints (8), (9), and (10). We will show that if $c_{ij} \leq 0$, then x_{ij} will not affect the optimal solution of the objective function (IP-MM). Let us consider the term $c_{ij}(1-x_{ij})$ in the objective function (IP-MM). If $c_{ij} \leq 0$, then regardless of the value of x_{ij} (0 or 1), the term $c_{ij}(1 - x_{ij})$ will be non-positive. Thus, including x_{ij} in the objective function with $c_{ij} \leq 0$ does not contribute to maximizing the objective. On the other hand, if $c_{ij} > 0$, including x_{ij} in the objective function can potentially increase the objective value by setting x_{ij} to 0 (i.e., nodes i and j belong to the same community) since $c_{ij}(1 - 0) = c_{ij}$. Therefore, it is sufficient to consider only the variables x_{ij} for which $c_{ij} > 0$ in the objective function (IP-MM). □

Theorem 1: For the given integer programming problem, if we exclude the constraints involving variables x_{ij} where $c_{ij} \leq 0$, the optimal solution of the modified problem remains the same as the original problem.

Proof. Let us assume that we have an optimal solution to the original (IP-MM) model, which satisfies all the constraints including those involving variables x_{ij} where $c_{ij} \leq 0$. We will show that by excluding these constraints, we can still obtain the same optimal solution. If x_{ij} satisfies the constraints (8), (9), and (10), its value will not change when we exclude the constraints involving $c_{ij} \leq 0$. The reason is that these constraints do not impose any restrictions on x_{ij}; they only provide additional information. Removing them does not alter the feasible region. Furthermore, since $c_{ij} \leq 0$, excluding these constraints means that the corresponding term $c_{ij}(1 - x_{ij})$ is non-positive and does not contribute to the objective function. Therefore, the objective value remains unchanged. Consequently, the optimal solution for the modified problem without constraints involving x_{ij} where $c_{ij} \leq 0$ is the same as the optimal solution for the original problem. This concludes the proof of the theorem. □

Having these considered, one can simplify (IP-MM) by considering only the variables x_{ij} where $c_{ij} > 0$ in the objective function. We can express this modified model as follows:

$$\max \sum_{(i,j)\in I_{pos}} c_{ij}(1 - x_{ij}), \tag{12}$$

where I_{pos} is the set of all pairs $(i, j) \in I_{all}$ for which $c_{ij} > 0$. We still need to ensure that the constraints (8), (9), and (10) hold for the selected variables. To achieve this, we introduce new binary variables y_{ijk} for all $i < j < k$ such that:

$$y_{ijk} = \begin{cases} 1 & \text{if } x_{ij} + x_{jk} - x_{ik} \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

By introducing these variables, we can replace the constraints (8), (9), and (10) with the following constraint:

$$x_{ij} + x_{jk} - x_{ik} \geq 0 \quad \forall i < j < k \text{ s.t. } (i, j) \in I_{pos} \tag{14}$$

Finally, we include the binary variable definitions:

$$x_{ij} \in \{0, 1\} \qquad \forall (i, j) \in I_{pos} \tag{15}$$

$$y_{ijk} \in \{0, 1\} \qquad \forall i < j < k \text{ s.t. } (i, j) \in I_{pos} \tag{16}$$

Gathering all together, we get an equivalent sparse model for (IP-MM) as follows:

$$\begin{aligned} \max \quad & \sum_{(i,j)\in I_{pos}} c_{ij}(1 - x_{ij}) && \text{(IPs-MM)} \\ & x_{ij} + x_{jk} - x_{ik} \geq 0 && \forall i < j < k \text{ s.t. } (i, j) \in I_{pos} \quad (17) \\ & x_{ij} \in \{0, 1\} && (i, j) \in I_{pos} \quad (18) \\ & y_{ijk} \in \{0, 1\} && \forall i < j < k \text{ s.t. } (i, j) \in I_{pos} \quad (19) \end{aligned}$$

Highly encouraging is that (IPs-MM) has considerably fewer constraints and variables compared to the original (IP-MM) model but preserves the same set of optimal solutions.

III. Solution Approach

Despite (IPs-MM) providing us with a considerably simpler integer modeling than (IP-MM), it could still remain unlikely to obtain the optimal solution, particularly when dealing with average to large-scale networks. Undoubtedly, one possible way to speed up the branch and bound technique using the CPLEX solver is to feed it with a reasonably good initial solution. Starting with a smart choice among the feasible solution space has turned out to improve performance immensely. In this vein, we came to take advantage of the heuristic two-stage community detection algorithm introduced in [10]. Considerably relevant to our research is the first stage of the algorithm's authority to swiftly find a collection of initial communities with excellent quality w.r.t. optimizing a connectivity-based criterion they established.

In this procedure, the degree of a node and its set of neighbors are defined naturally. Additionally, the inner and outer edges of a community C, denoted as E_C^{in} and E_C^{out}, respectively, represent the edges with both endpoints and one endpoint within C. Let $\alpha \in \{1, \ldots, diameter(G)\}$ be an integer and define $P_\alpha \subseteq V$ as a sequence of nodes arranged in descending order according to their degrees, provided they are all at least α steps away from each other. P_α is referred to as the *influential nodes* of G with respect to α. The distance between a vertex $j \notin C$ and the community C is then determined as the length of the shortest path from j to the influential node of C. The radius of a community C, denoted by $r(C)$, is the maximum distance from its influential node to other vertices within the community.

Subsequently, the authors of [10] proposed a measure to compute the quality of a community C:

$$q(C) = \frac{|E_C^{out}| + r(C)}{|E_C^{in}|}. \tag{20}$$

Furthermore, given a set of communities $\mathbb{C} = \{C_1, \ldots, C_k\}$, the quality of the partition $\mathbb{C}$ is defined as follow:

$$q_G^{\mathbb{C}} = \sum_{C\in\mathbb{C}} q(C). \tag{21}$$

Having everything considered, we can now summarize the procedure of determining an appropriate set of initial communities in the following manner:

- Repeat the procedure below for all α between 2 and $diameter(G)$ and pick $\mathbb{C}_\alpha$ with the minimum $q_G^{\mathbb{C}_\alpha}$ as the best initial set of communities.
  - Establish P_α as defined above.
  - Let each $p \in P_\alpha$ initially designate a sole community, leading to a set of $|P_\alpha|$ communities $\mathbb{C}_\alpha = \{C_1, \ldots, C_{|P_\alpha|}\}$.
  - Assign every $v \notin \mathbb{C}_\alpha$ to its closest community until all nodes belong to a community.

An essential insight of this approach revolves around the idea of ensuring a meaningful spatial distribution of the influential nodes of the network, striking a balance between proximity to facilitate cohesive communities and sufficient distance to avoid interference. To achieve this, the notion of $\alpha-$far nodes is introduced to serve as a criterion to evaluate the distance between potential influential vertices. By leveraging this parameter, one can successfully identify the best set of influential nodes capable of bunching and leading their surrounding vertices.

Now, by putting everything together, we devise the following tractable procedure for optimally solving (IPs-MM):

- Start with the initial communities obtained with the method explained above and employ the following row generation technique:

TABLE I: Networks under-study

ID	Network	n	m
1	Zachary's karate club [18]	34	78
2	Mexican Politicians [19]	35	117
3	Dolphin network [20]	62	159
4	Les Miserables [20]	77	254
5	p53 protein [21]	104	226
6	Books about U.S. politics [22]	105	441
7	American college football [23]	115	613
8	Citation graph drawing [24]	311	640
9	USAir97 [25]	332	2126
10	C. Elegans [26]	453	2025
11	Erdos collaboration [27]	472	1314
12	Electronic circuit [28]	512	819

1) Consider (IPs-MM) without any constraints.
2) Use the CPLEX solver and apply the branch and bound technique to obtain an optimal solution x^* to (IPs-MM).
3) Verify whether all constraints of (IPs-MM) are satisfied by x^*. If not, add the violated ones to the model and go to (2).

IV. Computational results

Within this section, we conduct a comprehensive performance evaluation of our proposed methodology. To maintain fairness, we take into account exactly the set of 12 networks used in [14]. These networks, outlined in Table I, are among the recognized and commonly utilized real-world networks utilized in this context, and each of them possesses a corresponding ground truth, representing the optimal community structures. Hence, it becomes convenient to evaluate the effectiveness of a community detection algorithm by quantifying the similarities between the algorithmically derived communities and the ground truth. To facilitate this evaluation, we employ the widely acknowledged performance metric NMI (*Normalized Mutual Information*).

A. Normalized Mutual Information (NMI)

NMI, as described in [29], is a widely recognized and established metric for evaluating the similarity between clusters. However, it can effectively measure the agreement between the optimal communities and those discovered by an algorithm. Consider a network G with n nodes, where $\mathcal{C}(\mathcal{A}) = C_1, \ldots, C_k$ represents the communities obtained by algorithm $\mathcal{A}$, and $\mathcal{C}' = C'_1, \ldots, C'_{k'}$ denotes the ground truth communities. The NMI value corresponding to the algorithm $\mathcal{A}$ can be computed as follows

$$NMI(\mathcal{A}) = \frac{-2\sum_{x=1}^{|\mathcal{C}|}\sum_{y=1}^{|\mathcal{C}'|}\frac{|C_x \cap C'_y|}{n} log(\frac{n|C_x \cap C'_y|)}{|C_x||C'_y|})}{\sum_{x=1}^{|\mathcal{C}|}\frac{C_x}{n} log(\frac{C_x}{n}) + \sum_{y=1}^{|\mathcal{C}'|}\frac{C'_y}{n} log(\frac{C'_y}{n})} \tag{22}$$

When the detected communities perfectly align with the ground truth, NMI attains its maximum value of one. Conversely, if the two sets exhibit no similarity, the NMI score is zero. In general, a higher NMI value indicates a more accurate and effective discovery of community structures.

B. Experiments

In what follows, we present a thorough evaluation demonstrating our proposed method's superiority in achieving high-quality communities over several competing algorithms. All the experiments are conducted on a computer system with a processor *Intel(R) Core i9-12900KF @ 3.2GHz, 16.6 Core(s)* and *128 GB* of *RAM*. Algorithms are implemented with C++, and CPLEX optimizer 12.9 is used for solving linear programming.

Fig. 2 delivers the comparisons by evaluating communities that are discovered based on the following cases:

- Our method (the blue curve): Obtaining the communities by optimally solving (IPs-MM) with the proposed method consisting of a row and column generation procedure.
- Method proposed in [14] (the red diagram): Solving the linear relaxation model of (IP-MM) via a row and column generation technique and applying a local search manner rounding procedure for obtaining the communities.
- Max-Min Modularity, proposed in [13] (green diagram): Using a user-defined relation matrix and applying a hierarchical heuristic algorithm for maximizing the Max-Min Modularity.

It is evident to conclude the promising outperformance of our proposed community detection method. In particular, the considerable gap between the blue and red curves clearly shows the advancement of using an exact method rather than relying on just heuristic approaches. While the technique in [14] took into account optimally solving the liner relaxation version of the (IP-MM), their proposed local search-based rounding procedure for obtaining the solution to (IP-MM) causes a significant error. The lever provided by the simpler model (IPs-MM), which was proven to be equivalent to (IP-MM), enabled us to seek an optimal solution to the model and, therefore, discover high-quality communities considerably better than those in [14]. This dominance could be more pronounced when noticing that the communities obtained in [14] were superior to a wide range of other algorithms.

Herein, for the sake of more visualization, Fig. 3 displays the schematic representations of the *Erdos collaboration* and *C. elegans* networks, along with the communities identified by the proposed algorithm.

Furthermore, to strengthen the assessment of our algorithm, we also decided to examine its execution time, for which we came to follow twofold perspectives: first, to determine how solving (IPs-MM) instead of (IP-MM) could enhance time complexity, and second, to compare the execution time of our model with that of the integer programming model proposed in [14] for the Max-Min Modularity Maximization problem. Fig. 4 illustrates the results of these comparisons.

Our proposed row and column generation technique, coupled with the intelligently determined initial solution, resulted in a significant speed improvement when optimally solving (IPs-MM), surpassing the performance of solving (IP-MM).

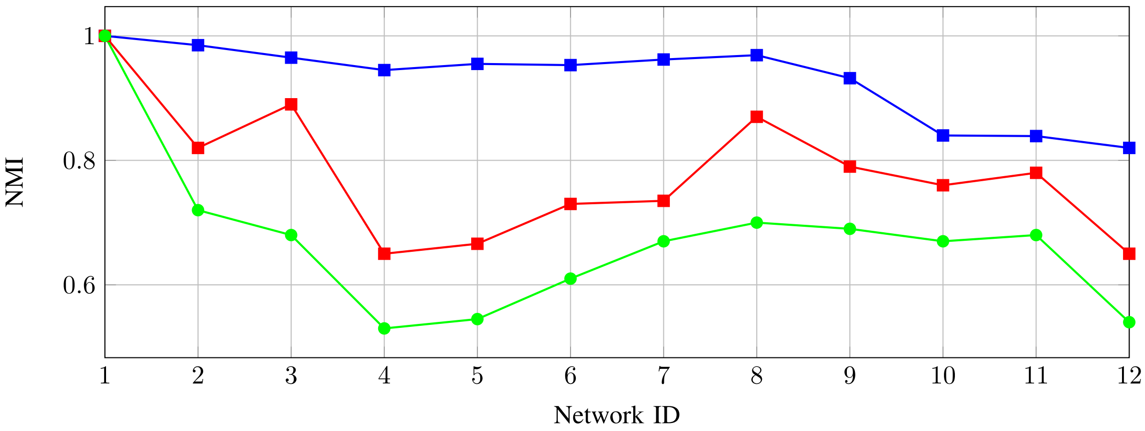
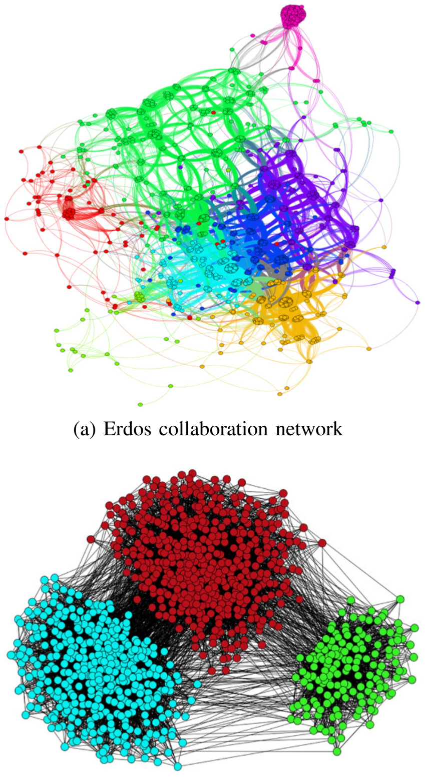


Fig. 2: Comparison between NMI values achieved by (i) blue curve: our method, (ii) red curve: method in [14], and (iii) green curve: the conventional Max-Min modularity.



(a) Erdos collaboration network

(b) C. Elegans network

Fig. 3: Two networks from Table I and their detected communities using the proposed algorithm.

This highlights the substantial impact of our simplified model and solution approach. Furthermore, our method demonstrated faster execution compared to solving the equivalent integer formulation of the sub-problem for the Max-Min Modularity Maximization problem proposed in [14], which further validates the efficiency of our simplification. As could be naturally expected due to the NP-hardness nature of the problem, our model performs slower than when solving the LP relaxation version of the model in [14] plus using the rounding algorithm. Nevertheless, the inaccurate results obtained in [14] reveal its untrustworthy against this work's proposed model.

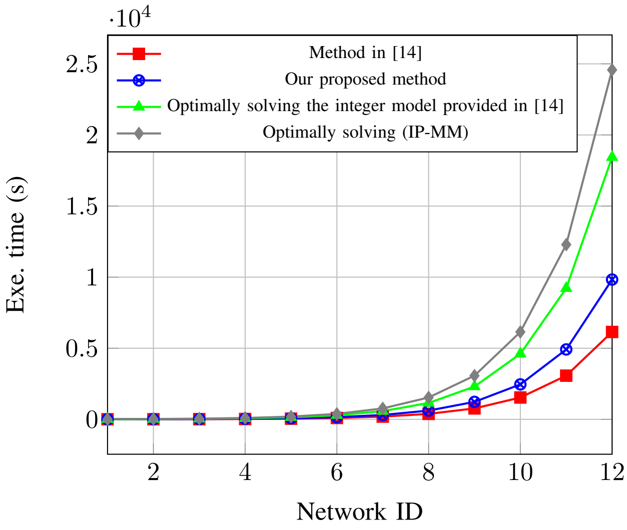


Fig. 4: The time elapsed (in terms of seconds) for solving different methods.

We complete this section by highlighting that even though the method presented in [14] can yield promising community structures in large-scale networks, for situations where accuracy is paramount, exact methods become significantly crucial. In such cases, the proposed method in this work can provide substantial assistance.

V. Conclusion

In this study, we addressed the Max-Min Modularity Maximization problem, a widely recognized metric for community evaluation. To enhance the problem's solution efficiency, we proposed an integer programming model that exhibits a reduced number of variables and constraints while preserving the same set of optimal solutions as the original model. By incorporating a row and column generation technique guided by an intelligently determined initial feasible solution, we were able to achieve optimal solutions in a remarkably efficient manner. The resulting solution provided us with a set of communities that exhibit notable similarities with the optimal community structures, indicating the effectiveness of our approach. This not only improved the overall quality of the obtained communities but also demonstrated the advantages of our model in terms of computational time.

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] P. Hui, E. Yoneki, S. Y. Chan, and J. Crowcroft, "Distributed community detection in delay tolerant networks," in *Proceedings of 2nd ACM/IEEE international workshop on Mobility in the evolving internet architecture*, 2007, pp. 1–8.

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

[8] 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.

[9] 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.

[10] A. Ferdowsi, M. Dehghan Chenary, and A. Khanteymoori, "Tscda: a dynamic two-stage community discovery approach," *Social Network Analysis and Mining*, vol. 12, no. 1, p. 46, 2022.

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

[12] A. Ferdowsi, "An integer programming approach reinforced by a message-passing procedure for detecting dense attributed subgraphs," in *2022 17th Conference on Computer Science and Intelligence Systems (FedCSIS)*. IEEE, 2022, pp. 569–576.

[13] 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.

[14] A. Ferdowsi and A. Khanteymoori, "Discovering communities in networks: A linear programming approach using max-min modularity," in *2021 16th Conference on Computer Science and Intelligence Systems (FedCSIS)*. IEEE, 2021, pp. 329–335.

[15] T. N. Dinh and M. T. Thai, "Finding community structure with performance guarantees in complex networks," *arXiv preprint arXiv:1108.4034*, 2011.

[16] A. Miyauchi and Y. Miyamoto, "Computing an upper bound of modularity," *The European Physical Journal B*, vol. 86, no. 7, p. 302, 2013.

[17] M. E. Newman, "Analysis of weighted networks," *Physical review E*, vol. 70, no. 5, p. 056131, 2004.

[18] 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.

[19] J. Gil-Mendieta and S. Schmidt, "The political network in mexico," *Social Networks*, vol. 18, no. 4, pp. 355–381, 1996.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] N. Meghanathan, "A greedy algorithm for neighborhood overlap-based community detection," *Algorithms*, vol. 9, no. 1, p. 8, 2016.

[25] V. Batagelj and A. Mrvar, "Pajek datasets (2006)," 2009.

[26] 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.

[27] A. Mrvar and V. Batagelj, *Pajek: Programs for Analysis and Visualization of Very Large Networks: Reference Manual: List of Commands with Short Explanation Version 5.10*. A. Mrvar, 2020.

[28] S. Chand and S. Mehta, "Community detection using nature inspired algorithm," in *Hybrid Intelligence for Social Networks*. Springer, 2017, pp. 47–76.

[29] 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.