Heuristic Modularity Maximization Algorithms for Community Detection Rarely Return an Optimal Partition or Anything Similar

Samin Aref[1][0000−0002−5870−9253],
Mahdi Mostajabdaveh[2][0000−0002−2816−909X], and
Hriday Chheda[3][0000−0002−9049−0255]

[1] Department of Mechanical and Industrial Engineering, University of Toronto, M5S3G8, Canada aref@mie.utoronto.ca
[2] Huawei Technologies Canada, V5C 6S7, Canada
[3] Department of Computer Science, University of Toronto, M5S2E4, Canada

Abstract. Community detection is a fundamental problem in computational sciences with extensive applications in various fields. The most commonly used methods are the algorithms designed to maximize modularity over different partitions of the network nodes. Using 80 real and random networks from a wide range of contexts, we investigate the extent to which current heuristic modularity maximization algorithms succeed in returning maximum-modularity (optimal) partitions. We evaluate (1) the ratio of the algorithms' output modularity to the maximum modularity for each input graph, and (2) the maximum similarity between their output partition and any optimal partition of that graph. We compare eight existing heuristic algorithms against an exact integer programming method that globally maximizes modularity. The average modularity-based heuristic algorithm returns optimal partitions for only 16.9% of the 80 graphs considered. Additionally, results on adjusted mutual information reveal substantial dissimilarity between the sub-optimal partitions and any optimal partition of the networks in our experiments. More importantly, our results show that near-optimal partitions are often disproportionately dissimilar to any optimal partition. Taken together, our analysis points to a crucial limitation of commonly used modularity-based heuristics for discovering communities: they rarely produce an optimal partition or a partition resembling an optimal partition. If modularity is to be used for detecting communities, exact or approximate optimization algorithms are recommendable for a more methodologically sound usage of modularity within its applicability limits.

Keywords: Community detection · Network science · Modularity maximization · Integer programming · Graph optimization.

1 Introduction

Community detection (CD), the process of inductively identifying communities within a network, is a core problem in computational sciences, particularly,

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36027-5_48

in physics, computer science, biology, and computational social science [50,19]. Among common approaches for CD are the algorithms which are designed to maximize a utility function, modularity [37], across all possible ways that the nodes of the input network can be partitioned into communities. Modularity measures the fraction of edges within communities minus the expected fraction if the edges were distributed randomly; with the random distribution of the edges being a null model that preserves the node degrees. Despite their name and design philosophy, current modularity maximization algorithms, which are used by no less than tens of thousands of peer-reviewed studies [28], are not guaranteed to maximize modularity [38,24,35]. This has led to uncertainty [20,24] in the extent to which they succeed in returning a maximum-modularity (optimal) partition or something similar.

Modularity is among the first objective functions proposed for optimization-based detection of communities [37,18]. Several limitations [21,16,18,41] of modularity including the resolution limit [17] have led researchers to develop alternative CD methods using stochastic block modeling [23,40,32,45], information theoretic approaches [43,44], and alternative objective functions [2,47,36,34]. Modularity-based algorithms are the most commonly used method for CD [46,19]. Despite the widespread adoption of modularity-based heuristics, there is uncertainty [20,24] in their success in maximizing modularity. This study aims to address this uncertainty by quantifying the extent to which eight commonly used heuristics [13,7,30,46,50,8,48,31] succeed in returning an optimal partition or a partition resembling an optimal partition. After describing the methods and materials, we present the main results followed by a discussion of the methodological ramifications and future directions.

2 Methods and Materials

This study aims to investigate the extent to which eight commonly used heuristic modularity maximization algorithms [13,7,30,46,50,8,48,31] succeed in returning an optimal partition or a partition similar to an optimal partition. To achieve this objective, we quantify the proximity of their results to the globally optimal partition(s), which we obtain using an exact Integer Programming (IP) model for maximizing modularity [9,1,15]. We do not claim that maximum-modularity partitions represent best partitions. Throughout the paper, we use the terms network and graph interchangeably.

2.1 Modularity

Consider the simple graph $G = (V, E)$ with $|V| = n$ nodes, $|E| = m$ edges, adjacency matrix entries a_{ij}, and a partition $X = \{V_1, V_2, \ldots, V_k\}$ of the node set V into k communities. The modularity function $Q_{(G,X)}$ is computed [37,18] according to Eq. (1)

$$Q_{(G,X)} = \frac{1}{2m} \sum_{(i,j)\in V^2, i\leq j} \left(a_{ij} - \gamma \frac{d_i d_j}{2m} \right) \delta(i,j) \tag{1}$$

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36027-5_48

where d_i represents the degree of node i, γ is the resolution parameter[4], and $\delta(i,j)$ is 1 if nodes i and j are in the same community otherwise 0. The term associated with each pair of nodes (i,j) is alternatively represented as $b_{ij} = a_{ij} - \gamma\frac{d_i d_j}{2m}$ and referred to as the modularity matrix entry for (i,j).

2.2 Modularity maximization

The modularity maximization problem for input graph $G = (V,E)$ involves finding a partition X^* whose associated $Q_{(G,X^*)}$ is globally maximum over all possible partitions of the node set V.

2.3 Sparse IP formulation of modularity maximization

Consider the simple graph $G = (V,E)$ with modularity matrix entries b_{ij}, obtained using the resolution parameter γ. We use the binary decision variable x_{ij} for each pair of distinct nodes $(i,j), i<j$. Their community membership is either the same (represented by $x_{ij}=0$) or different (represented by $x_{ij}=1$). Accordingly, the problem of maximizing the modularity of input graph G can be formulated as an IP model [15] as in Eq. (2).

$$\begin{aligned} \max_{x_{ij}} Q &= \frac{1}{2m}\left(\sum_{(i,j)\in V^2, i<j} b_{ij}(1-x_{ij}) + \sum_{(i,i)\in V^2} b_{ii}\right) \\ \text{s.t.} \quad & x_{ik} + x_{jk} \geq x_{ij} \quad \forall (i,j)\in V^2, i<j, k\in K(i,j) \\ & x_{ij}\in\{0,1\} \quad \forall (i,j)\in V^2, i<j \end{aligned} \tag{2}$$

In Eq. (2), the optimal objective function value equals the maximum modularity for the input graph G. An optimal community assignment is characterized by the optimal values of the x_{ij} variables. $K(i,j)$ indicates a minimum-cardinality separating set [15] for the nodes i,j. Using $K(i,j)$ in the IP model of this problem leads to a more efficient formulation with $\mathcal{O}(n^2)$ constraints [15] instead of $\mathcal{O}(n^3)$ constraints as in earlier IP formulations of the problem [9,1]. Solving this optimization problem is NP-complete [9,35]. We use the *Gurobi* solver (version 10.0) [22] to solve it for the small and mid-sized instances as outlined in Subsection 2.6.

2.4 Reviewing eight heuristic modularity maximization algorithms

We evaluate eight modularity maximization heuristics known as Clauset-Newman-Moore (CNM) [13], Louvain [7], Leicht-Newman (LN) [30], Combo [46], Belief [50], Paris [8], Leiden [48], and EdMot-Louvain [31]. We have used the Python implementations of these eight algorithms which are accessible in the Community Discovery library (*CDlib*) version 0.2.6 [42].

[4] Without loss of generality, we set $\gamma = 1$ for all the analysis in this paper.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36027-5_48

We briefly describe how these eight algorithms use modularity to discover communities. The CNM algorithm initializes each node as a community by itself. It then follows a greedy scheme of merging two communities that contribute the maximum positive value to modularity [13]. The Louvain algorithm involves two sets of iterative steps: (1) locally moving nodes for increasing modularity and (2) aggregating the communities from the first step [7]. Despite Louvain being the most commonly used modularity-based algorithm [28], it may sometimes lead to disconnected components in the same community [48]. The LN algorithm uses spectral optimization to maximize modularity which also supports directed graphs [30]. The Combo algorithm is a general optimization-based CD method which supports modularity maximization among other tasks. It involves two sets of iterative steps: (1) finding the best merger, split, or recombination of communities to maximize modularity and (2) performing a series of Kernighan-Lin bisections [26] on the communities as long as they increase modularity [46]. The Belief algorithm seeks the consensus of different high-modularity partitions through a message-passing algorithm [50] motivated by the premise that maximizing modularity can lead to many poorly correlated competing partitions. The Paris algorithm is suggested to be a modularity-maximization scheme with a sliding resolution [8]; that is, an algorithm capable of capturing the multi-scale community structure of real networks without a resolution parameter. It generates a hierarchical community structure based on a simple distance between communities using a nearest-neighbour chain [8]. The Leiden algorithm attempts to resolve a defect of the Louvain algorithm in returning badly connected communities. It is suggested to guarantee well-connected communities in which all subsets of all communities are locally optimally assigned [48]. The EdMot-Louvain algorithm (EdMot for short) is developed to overcome the hypergraph fragmentation issue observed in previous motif-based CD methods [31]. It first creates the graph of higher-order motifs (small dense subgraph patterns) and then partitions it using the Louvain method to heuristically maximize modularity using higher-order motifs [31].

To evaluate these eight modularity-based algorithms in maximizing modularity, we quantify (1) the ratio of their output modularity to the maximum modularity for each input graph and (2) the maximum similarity between their output partition and any optimal partition of that graph. We obtain optimal partitions by solving the IP model in Eq. (2) using the Gurobi solver (version 10.0) with a termination criterion ensuring global optimality [22].

2.5 Measures for evaluating heuristic algorithms

For a quantitative measure of proximity to global optimality, we define and use the *Global Optimality Percentage* (GOP) as the fraction of the modularity returned by a heuristic method for a network divided by the globally maximum modularity for that network (obtained by solving the IP model in Eq. (2)). In all cases where the modularity returned by a heuristic method equals the maximum modularity for the input graph, we set GOP=1. In cases where a heuristic algorithm returns a partition with a negative modularity value, we set

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36027-5_48

GOP=0 to facilitate easier interpretation of proximity to optimality based on non-negative GOP values.

We also use a quantitative measure for the similarity of a partition to an optimal partition. Normalized Adjusted Mutual Information (AMI) [49] is a measure of similarity between two partitions of the same network. Unlike normalized mutual information [49], AMI adjusts the measurement based on the similarity that two partitions may have by pure chance. AMI for a pair of identical partitions (or permutations of the same partition) equals 1. For two different partitions, however, AMI takes a smaller value (including 0 or negative values close to 0 for two extremely dissimilar partitions).

2.6 Data and resources

For our computational experiments, we include 60 real networks[5] with no more than 2812 edges as well as 10 Erdős-Rényi graphs and 10 Barabási-Albert graphs with 125-153 edges. These instance sizes were chosen to ensure all algorithms terminate within a reasonable time. The computational experiments were implemented in Python 3.9 using a notebook computer with an Intel Core i7-11800H @ 2.30GHz CPU and 64 GB of RAM running Windows 10.

3 Results

We present the main results from our experiments in the following four subsections. In Subsection 3.1, we compare partitions from different algorithms on a single network. In Subsection 3.2, we examine the multiplicity of optimal partitions and investigate the similarity between multiple optimal partitions of the same networks. In Subsection 3.3, we evaluate the effectiveness of the heuristic algorithms on 80 networks by measuring the distance of sub-optimal partitions from an optimal partition. Finally, in Subsection 3.4, we investigate the success rate of the heuristic algorithms in finding an optimal partition.

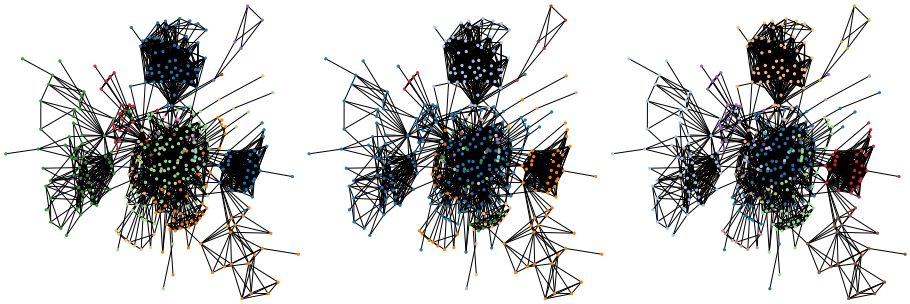
3.1 Comparing partitions from different algorithms on one network

Figure 1 shows one graph and its nine partitions returned by nine CD methods. This graph[6] represents an anonymized Facebook *ego network*[7]. Nodes represent Facebook users, and an edge exists between any pair of users who were friends on Facebook in April 2014 [33]. Communities are shown using node colors.

Panel 1a of Figure 1 shows an optimal partition obtained by solving the IP model in Eq. (2) for the network facebook_friends. It involves $k = 28$ communities, and a maximum modularity value of $Q^* = 0.7157714$. The partitions from the eight heuristic modularity maximization algorithms are all sub-optimal as

[5] All networks are accessible from the Netzschleuder with the details in the Appendix.
[6] facebook_friends network [33] from the Netzschleuder repository
[7] A network of one person's social ties to other persons and their ties to each other

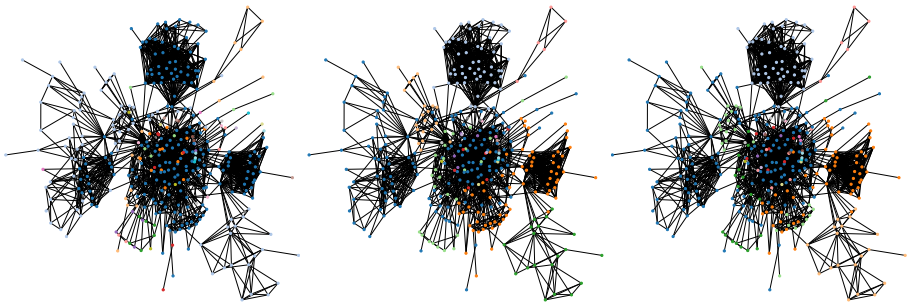
ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36027-5_48



(a) IP, $Q^* = 0.7157714$, $k = 28, \text{AMI} = 1$

(b) CNM, $Q = 0.6971$, $k = 30, \text{AMI} = 0.829$

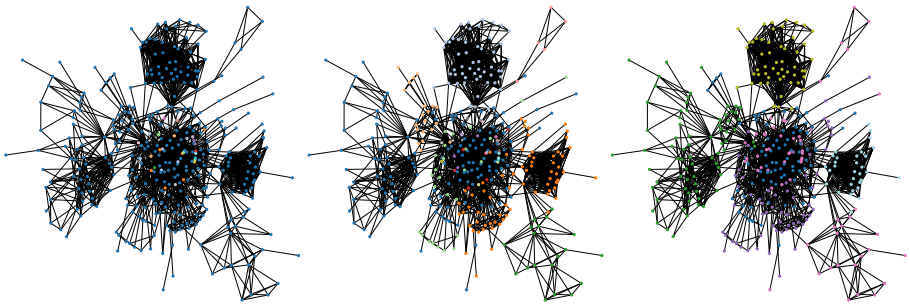
(c) Combo, $Q = 0.7157709$, $k = 13, \text{AMI} = 0.949$



(d) EdMot, $Q = 0.4902$, $k = 53, \text{AMI} = 0.651$

(e) Leiden, $Q = 0.7082$, $k = 32, \text{AMI} = 0.908$

(f) Louvain, $Q = 0.7087$, $k = 29, \text{AMI} = 0.920$



(g) Paris, $Q = 0.0338$, $k = 20, \text{AMI} = 0.363$

(h) LN, $Q = 0.7139$, $k = 28, \text{AMI} = 0.971$

(i) Belief, $Q = 0.4566$, $k = 3, \text{AMI} = 0.786$

Fig. 1: Modularity maximization for one network using nine methods leading to one optimal partition (panel a) and eight sub-optimal partitions (panels b-i) with different Q, k, and AMI values. (Magnify the high-resolution color figure on screen for more details.)

depicted in panels 1b–1i of Figure 1. Compared to other algorithms, the two algorithms Combo and LN have more success in achieving proximity to an optimal partition. LN returns a partition with $k = 28$ communities and a modularity of $Q = 0.7139$ which has the highest AMI among all heuristics (0.971). The rela-

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36027-5_48

tive success of the Combo algorithm is in returning a high-modularity partition with $Q = 0.7157709$, but with $k = 13$ communities and a lower AMI (0.949) compared to LN. The sub-optimal partitions from the other six algorithms have more substantial variations in Q, AMI, and k (number of communities) as shown by the values in the corresponding subcaptions in Figure 1.

3.2 Multiplicity of optimal partitions

While the partition which maximizes modularity is often unique, some graphs have multiple optimal partitions. For all networks considered in our analysis, we obtain all optimal partitions using the Gurobi solver by running it with a special configuration for finding all optimal partitions [22]. Figure 2 shows a protein network[8] and its four optimal partitions. In this network, nodes represent proteins and an edge represents a binding interaction between two proteins (PDZ-domain-mediated protein–protein binding interaction) [6]. All four optimal partitions have $Q^* = 0.80267$ and $k = 29$.

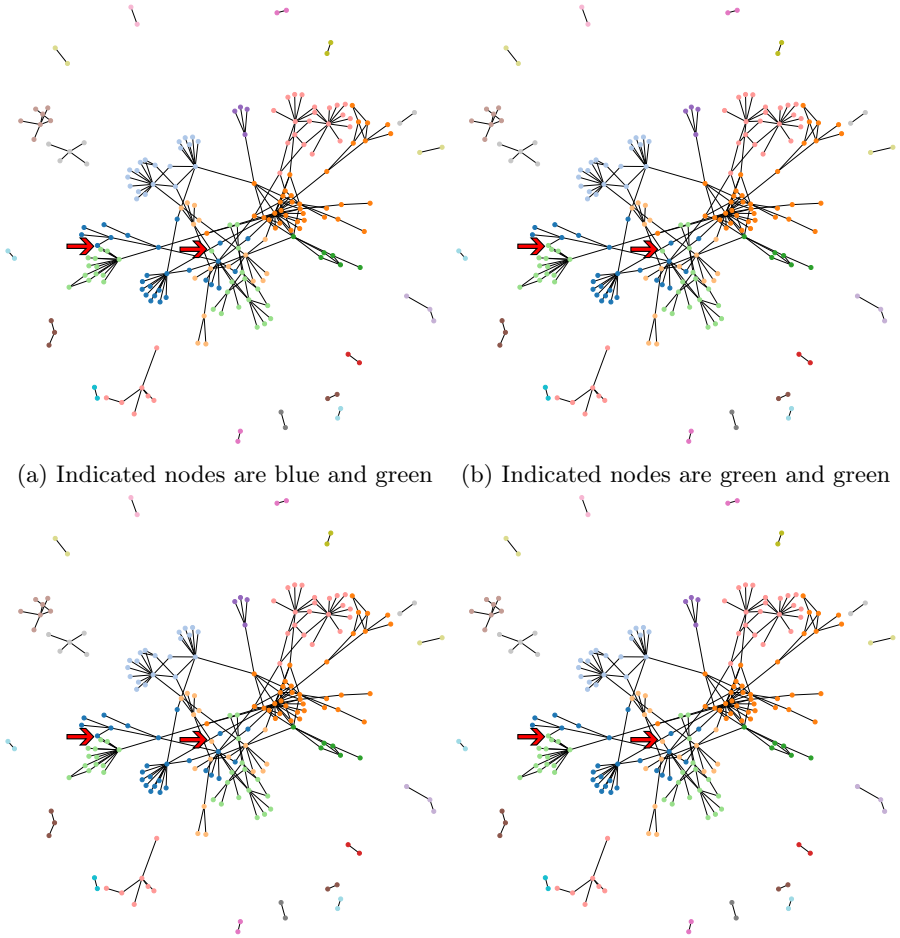
The differences between optimal partitions of this network are in the community assignments for two nodes indicated by red arrows in Figure 2. The six pairwise AMI values for the optimal partitions are all > 0.98 confirming the high level of similarity between the four optimal partitions in Figure 2.

Obtaining all optimal partitions for all 80 networks, we observed that 89% of the graphs have unique optimal partitions and the multiplicity of optimal partitions is a relatively rare event. Given the possibility of multiple optimal partitions in some graphs, we calculated the AMI for the partition of each heuristic algorithm and each of the multiple optimal partitions of that graph. We then conservatively reported the maximum AMI of each heuristic for each graph to quantify the similarity between that partition and its closest optimal partition. Consequently, a low value of AMI for a partition obtained by a heuristic algorithm indicates its dissimilarity to any optimal partition.

Our results suggest that the rarely observed multiple optimal partitions of a graph often have a high degree of similarity (AMI values > 0.9) because their differences are often only in the community assignments of a very few nodes (as in Figure 2). Dissimilarity between multiple optimal partitions of a network seems to be exceptional, but it has been observed in one of our 80 networks: *contiguous USA*[9], where nodes are US states and each edge indicates a land-based border between two states. The AMI of the two optimal partitions for this network is exceptionally low (0.34). Upon further investigation, we observed that one optimal partition combines five communities of the other optimal partition together. This makes the two partitions related in terms of belonging to a clustering hierarchy, while they are not similar according to an AMI definition of partition similarity. These exceptional cases are possible due to the mathematical symmetries resulted from the value of γ used in Eq. (1) for defining modularity. Our results suggest that there is usually a distinct uniqueness to an

[8] interactome_pdz network [6] from the Netzschleuder repository
[9] contiguous_usa network [27] from the Netzschleuder repository

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36027-5_48



(a) Indicated nodes are blue and green (b) Indicated nodes are green and green

(c) Indicated nodes are blue and orange (d) Indicated nodes are green and orange

Fig. 2: A protein network and its four optimal partitions (panels a-d). The red arrows show the differences between optimal partitions. (Magnify the high-resolution color figure on screen for more details.)

optimal partition (or a group of similar optimal partitions) for a given network in comparison to sub-optimal partitions. This new perspective is contrary to the premise that maximizing modularity leads to many competing partitions with almost the same modularity [50] and no clear way of selecting between them [41]. It is the failure to actually maximize modularity that may lead to many poorly correlated competing partitions with unknown distances from the desired objective (both in modularity and in partition similarity). What remains to be analyzed is how different sub-optimal partitions are from an optimal partition

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36027-5_48

and how often heuristic modularity maximization algorithms return sub-optimal partitions. We investigate these two questions in the next two subsections.

3.3 Evaluating heuristic algorithms on 80 networks

For summarizing the results of eight heuristics on 80 networks, we present four scatter-plots of GOP and AMI. Figure 3 shows GOP on the y-axes and AMI on the x-axes for the combination of each network and algorithm. For each algorithm (color-coded), there are 60 data points for the 60 real networks and 2 data points representing the average of 10 Erdős-Rényi and the average of 10 Barabási-Albert graphs. The first three letters of the network names are indicated on each data point (magnify the figure on screen for the details). Four 45-degree lines are drawn to indicate the areas where the GOP and AMI are equal. In other words, the 45-degree lines show areas where the extent of sub-optimality ($1 - \text{GOP}$) is associated with a dissimilarity ($1 - \text{AMI}$) of the same size between the sub-optimal partition and any optimal partition.

Looking at the y-axes values in Figure 3, we observe that there is a substantial variation in the values of GOP (i.e. the extent of sub-optimality) for the eight heuristic algorithms. The Belief algorithm returns partitions associated with negative modularity values for 45 of the 80 instances (leading to most of its data points having GOP$= 0$ and being concentrated at the bottom of the scatter-plot). The Paris algorithm returns partitions with modularity values substantially smaller than the maximum modularity values. Aside from a few exceptions, all data points for Leiden and LN have the same position indicating their identical performance on most of these instances. The two algorithms CNM and EdMot seem to have higher variation in GOP (compared to the other algorithms) for these instances. Overall, the four algorithms with highest and increasing performance in returning close-to-maximum modularity values are LN, Leiden, Louvain, and Combo respectively. Despite that these instance are graphs with no more than 2812 edges, they are, according to Figure 3, challenging instances for these heuristic algorithms to optimize. Given that modularity maximization is an NP-complete problem [9,35], one can argue that the performance of these heuristic methods in term of proximity to an optimal partition does not improve for larger networks.

The x-axes values in Figure 3 show considerable dissimilarity between the sub-optimal partitions and an optimal partition for these 80 instances. Except for the Combo algorithm, a large number of the sub-optimal partitions obtained by these heuristic algorithms have AMI values smaller than 0.6. This indicates that their sub-optimal partitions are substantially different from any optimal partition. Even for data points concentrated at the top of the scatter-plots which have $0.95 < \text{GOP} < 1$, we see AMI values substantially smaller than 1. Compared to the other seven heuristics, Combo appears to consistently return partitions with large AMIs on a larger number of these 80 instances.

Focusing on the position of data points, we observe that they are mostly located above their corresponding 45-degree line. This indicates that sub-optimal partitions tend to be disproportionately dissimilar to any optimal partition (as

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36027-5_48

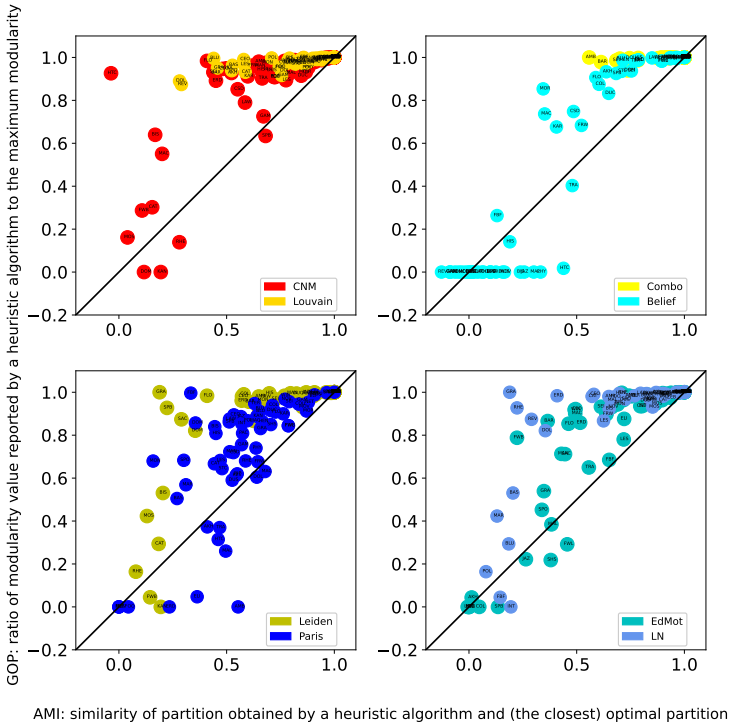


Fig. 3: Global optimality percentage and normalized adjusted mutual information measured for eight modularity maximization heuristics in comparison with (all) globally optimal partitions. (Magnify the high-resolution figure on screen for more details.)

foreshadowed in [14]). This result goes against the naive viewpoint that close-to-maximum modularity partitions are also close to an optimal partition. Our results are aligned with previous concerns that these heuristics may result in degenerate solutions far from the underlying community structure [20] and they have a high risk of algorithmic failure [24].

3.4 Success rate of heuristic algorithms in maximizing modularity

Our GOP results for the eight heuristic algorithms allow us to answer a fundamental question about the heuristic modularity maximization algorithms: how often each algorithm returns an optimal (a maximum-modularity) partition? We report the fraction of networks (out of 80) for which a given algorithm returns an optimal partition. Combo [46] has the highest success rate, returning an optimal partition for 55% of the networks. LN [30] and Leiden [48] maximize modularity for 26.2% of the networks considered. Louvain [7] has a success rate of 18.7%. The algorithms CNM [13], EdMot [31], Paris [8], and Belief [50] have success

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36027-5_48

rates of 5%, 2.5%, 1.2%, and 0% respectively. These are arguably low success rates for what the name *modularity maximization algorithm* implies or the idea of discovering network communities through maximizing a function.

Earlier in Figure 3, we observed that near-optimal partitions tend to be disproportionately dissimilar to any optimal partition. In other words, close-to-maximum modularity partitions are rarely close to any optimal partition. Taken together with the low success rates of heuristic algorithms in maximizing modularity, our results indicate a crucial mismatch between the design philosophy of modularity maximization algorithms for CD and their capabilities: heuristic modularity maximization algorithms rarely return an optimal partition or a partition resembling an optimal partition.

4 Discussions and Future Directions

Understanding modularity capabilities and limitations has been complicated by the under-studied sub-optimality of modularity-based heuristics and their methodological consequences. Previous methodological studies [29,12,11,36,41], which have shed light on other aspects, had rarely disentangled the heuristic aspect of these algorithms from the fundamental concept of modularity. Our study is a continuation of previous efforts [20] in separating the effects of sub-optimality (or the choice of using greedy algorithms [24]) from the effects of using modularity on the fundamental task of detecting communities.

We analyzed the effectiveness of eight heuristics in maximizing modularity. While our findings are limited to a few algorithms, their combined usage by tens of thousands of peer-reviewed studies [28] motivates the importance of this assessment. Most heuristic algorithms for modularity maximization tend to scale well for large networks [51]. They are widely used not only because of their scalability or ease of implementation [24], but also because their high risk of algorithmic failure is not well understood [24]. The scalability of these heuristics comes at a cost: their partitions have no guarantee of proximity to an optimal partition [20] and, as our results showed, they rarely return an optimal partition. Moreover, we showed that their sub-optimal partitions tend to be disproportionately dissimilar to any optimal partition.

Neither using modularity nor succeeding in maximizing it is required for CD at the big-picture level. A recent study suggests modularity maximization is the most problematic CD method and considers it harmful [41]. Another study shows that, given computational feasibility, exact maximization of multiresolution modularity outperforms other CD methods in accurate and stable retrieval of planted communities [4] suggesting the relevance of modularity for CD. For some applications and contexts, *general* CD algorithms [39] which scale to large instance sizes are needed. However, for a "narrow set of tasks" [39, pp.7], involving small and mid-sized networks, *specialized* algorithms which outperform general algorithms are useful.

Our findings suggest that if modularity is to be used for detecting communities, developing approximation [10,14,25] and exact [3,4] algorithms are rec-

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36027-5_48

ommendable for a more methodologically sound usage of modularity within its applicability limits. Exact algorithms can also reveal the formal guarantees of performance [19] for accurate modularity-based algorithms.

A promising path forward could be using the advances in integer programming to develop a specialized accurate algorithm for solving the modularity maximization IP models [9,1,15] for networks of practical relevance within the limits of computational feasibility. New heuristic and approximation algorithms that strike a balance between accurate computations and scalability may also be useful particularly for large-scale networks

Author contributions Conceptualization (SA); data curation (SA, HC); formal analysis (SA, MM); funding acquisition (SA); investigation (SA); methodology (SA,MM); project administration (SA); resources (SA, MM); software (SA, HC, MM); supervision (SA, MM); validation (SA, MM); visualization (SA, MM); writing - original draft preparation (SA); writing - review & editing (SA, MM).

Acknowledgements We are thankful to the three anonymous referees and Ly Dinh for their helpful comments. We acknowledge Zachary P. Neal for pointing us to this problem and Santo Fortunato for the encouraging correspondence. Accessing CDlib, Netzschleuder, and ICON has been particularly helpful in this study for which we thank Giulio Rossetti, Tiago P. Peixoto, Aaron Clauset, and everyone contributing to these open science initiatives. This study has been supported by the Data Sciences Institute at the University of Toronto.

References

1. Agarwal, G., Kempe, D.: Modularity-maximizing graph communities via mathematical programming. The European Physical Journal B **66**(3), 409–418 (2008). https://doi.org/10.1140/epjb/e2008-00425-1
2. Aldecoa, R., Marín, I.: Deciphering network community structure by surprise. PLOS ONE **6**(9), 1–8 (2011). https://doi.org/10.1371/journal.pone.0024195
3. Aloise, D., Cafieri, S., Caporossi, G., Hansen, P., Perron, S., Liberti, L.: Column generation algorithms for exact modularity maximization in networks. Physical Review E **82**(4), 046112 (2010). https://doi.org/10.1103/PhysRevE.82.046112
4. Aref, S., Chheda, H., Mostajabdaveh, M.: The Bayan algorithm: Detecting communities in networks through exact and approximate optimization of modularity. arXiv preprint arXiv:2209.04562 (2022)
5. Aref, S., Chheda, H., Mostajabdaveh, M.: Dataset of networks used in accessing the Bayan algorithm for community detection (2023). https://doi.org/10.6084/m9.figshare.22442785
6. Beuming, T., Skrabanek, L., Niv, M.Y., Mukherjee, P., Weinstein, H.: PDZBase: a protein–protein interaction database for PDZ-domains. Bioinformatics **21**(6), 827–828 (2005)
7. Blondel, V.D., Guillaume, J.L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. Journal of statistical mechanics: theory and experiment **2008**(10), P10008 (2008). https://doi.org/10.1088/1742-5468/2008/10/P10008

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36027-5_48

8. Bonald, T., Charpentier, B., Galland, A., Hollocou, A.: Hierarchical graph clustering using node pair sampling. In: MLG 2018 - 14th International Workshop on Mining and Learning with Graphs. London, UK (2018)
9. Brandes, U., Delling, D., Gaertler, M., Gorke, R., Hoefer, M., Nikoloski, Z., Wagner, D.: On modularity clustering. IEEE transactions on knowledge and data engineering **20**(2), 172–188 (2007)
10. Cafieri, S., Costa, A., Hansen, P.: Reformulation of a model for hierarchical divisive graph modularity maximization. Annals of Operations Research **222**, 213–226 (2014)
11. Chen, S., Wang, Z.Z., Tang, L., Tang, Y.N., Gao, Y.Y., Li, H.J., Xiang, J., Zhang, Y.: Global vs local modularity for network community detection. PLOS ONE **13**(10), 1–21 (2018). https://doi.org/10.1371/journal.pone.0205284
12. Chen, T., Singh, P., Bassler, K.E.: Network community detection using modularity density measures. Journal of Statistical Mechanics: Theory and Experiment **2018**(5), 053406 (2018). https://doi.org/10.1088/1742-5468/aabfc8
13. Clauset, A., Newman, M.E., Moore, C.: Finding community structure in very large networks. Physical review E **70**(6), 066111 (2004)
14. Dinh, T.N., Li, X., Thai, M.T.: Network clustering via maximizing modularity: Approximation algorithms and theoretical limits. In: 2015 IEEE International Conference on Data Mining. pp. 101–110 (2015). https://doi.org/10.1109/ICDM.2015.139
15. Dinh, T.N., Thai, M.T.: Toward optimal community detection: From trees to general weighted networks. Internet Mathematics **11**(3), 181–200 (2015)
16. Fortunato, S.: Community detection in graphs. Physics reports **486**(3-5), 75–174 (2010)
17. Fortunato, S., Barthélemy, M.: Resolution limit in community detection. Proceedings of the National Academy of Sciences **104**(1), 36–41 (2007)
18. Fortunato, S., Hric, D.: Community detection in networks: A user guide. Physics Reports **659**, 1–44 (2016). https://doi.org/10.1016/j.physrep.2016.09.002
19. Fortunato, S., Newman, M.E.: 20 years of network community detection. Nature Physics **18**, 848–850 (2022)
20. Good, B.H., De Montjoye, Y.A., Clauset, A.: Performance of modularity maximization in practical contexts. Physical Review E **81**(4), 046106 (2010)
21. Guimerà, R., Sales-Pardo, M., Amaral, L.A.N.: Modularity from fluctuations in random graphs and complex networks. Phys. Rev. E **70**, 025101 (Aug 2004)
22. Gurobi Optimization Inc.: Gurobi optimizer reference manual (2023), url: https://gurobi.com/documentation/10.0/refman/index.html date accessed 16 Feb 2023
23. Karrer, B., Newman, M.E.J.: Stochastic blockmodels and community structure in networks. Phys. Rev. E **83**, 016107 (2011)
24. Kawamoto, T., Kabashima, Y.: Counting the number of metastable states in the modularity landscape: Algorithmic detectability limit of greedy algorithms in community detection. Physical Review E **99**(1), 010301 (2019)
25. Kawase, Y., Matsui, T., Miyauchi, A.: Additive approximation algorithms for modularity maximization. Journal of Computer and System Sciences **117**, 182–201 (2021). https://doi.org/10.1016/j.jcss.2020.11.005
26. Kernighan, B.W., Lin, S.: An efficient heuristic procedure for partitioning graphs. The Bell system technical journal **49**(2), 291–307 (1970)
27. Knuth, D.E.: The Stanford GraphBase: a platform for combinatorial computing, vol. 1. ACM Press New York (1993)

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36027-5_48

28. Kosowski, A., Saulpic, D., Mallmann-Trenn, F., Cohen-addad, V.P.: On the power of Louvain for graph clustering. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (eds.) Advances in Neural Information Processing Systems 33 (NeurIPS'20) (2020)
29. Lancichinetti, A., Fortunato, S.: Limits of modularity maximization in community detection. Physical Review E **84**(6), 066122 (2011). https://doi.org/10.1103/PhysRevE.84.066122
30. Leicht, E.A., Newman, M.E.J.: Community structure in directed networks. Physical Review Letters **100**(11), 118703 (2008). https://doi.org/10.1103/PhysRevLett.100.118703
31. Li, P.Z., Huang, L., Wang, C.D., Lai, J.H.: EdMot: An edge enhancement approach for motif-aware community detection. In: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining. pp. 479–487 (2019)
32. Liu, X., Yang, B., Chen, H., Musial, K., Chen, H., Li, Y., Zuo, W.: A scalable redefined stochastic blockmodel. ACM Transactions on Knowledge Discovery from Data (TKDD) **15**(3), 1–28 (2021)
33. Maier, B.F., Brockmann, D.: Cover time for random walks on arbitrary complex networks. Physical Review E **96**(4), 042307 (2017)
34. Marchese, E., Caldarelli, G., Squartini, T.: Detecting mesoscale structures by surprise. Communications Physics **5**(1), 1–16 (2022)
35. Meeks, K., Skerman, F.: The parameterised complexity of computing the maximum modularity of a graph. Algorithmica **82**(8), 2174–2199 (2020)
36. Miasnikof, P., Shestopaloff, A.Y., Bonner, A.J., Lawryshyn, Y., Pardalos, P.M.: A density-based statistical analysis of graph clustering algorithm performance. Journal of Complex Networks **8**(3) (2020)
37. Newman, M.E.J.: Modularity and community structure in networks. Proceedings of the National Academy of Sciences **103**(23), 8577–8582 (2006). https://doi.org/10.1073/pnas.0601602103
38. Newman, M.E.J.: Equivalence between modularity optimization and maximum likelihood methods for community detection. Physical Review E **94**(5), 052315 (2016). https://doi.org/10.1103/PhysRevE.94.052315
39. Peel, L., Larremore, D.B., Clauset, A.: The ground truth about metadata and community detection in networks. Science advances **3**(5), e1602548 (2017)
40. Peixoto, T.P.: Efficient monte carlo and greedy heuristic for the inference of stochastic block models. Physical Review E **89**(1), 012804 (2014)
41. Peixoto, T.P.: Descriptive vs. Inferential Community Detection in Networks: Pitfalls, Myths and Half-Truths. Elements in the Structure and Dynamics of Complex Networks, Cambridge University Press (2023)
42. Rossetti, G., Milli, L., Cazabet, R.: CDlib: a Python library to extract, compare and evaluate communities from complex networks. Applied Network Science **4**(1), 1–26 (2019)
43. Rosvall, M., Bergstrom, C.T.: An information-theoretic framework for resolving community structure in complex networks. Proceedings of the National Academy of Sciences **104**(18), 7327–7331 (2007). https://doi.org/10.1073/pnas.0611034104
44. Rosvall, M., Bergstrom, C.T.: Maps of random walks on complex networks reveal community structure. Proceedings of the National Academy of Sciences **105**(4), 1118–1123 (2008). https://doi.org/10.1073/pnas.0706851105
45. Serrano, B., Vidal, T.: Community detection in the stochastic block model by mixed integer programming (2021)

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36027-5_48

46. Sobolevsky, S., Campari, R., Belyi, A., Ratti, C.: General optimization technique for high-quality community detection in complex networks. Physical Review E **90**(1), 012811 (2014)
47. Traag, V.A., Aldecoa, R., Delvenne, J.C.: Detecting communities using asymptotical surprise. Phys. Rev. E **92**, 022816 (2015). https://doi.org/10.1103/PhysRevE.92.022816
48. Traag, V.A., Waltman, L., van Eck, N.J.: From Louvain to Leiden: guaranteeing well-connected communities. Scientific Reports **9**(1) (2019). https://doi.org/10.1038/s41598-019-41695-z
49. Vinh, N.X., Epps, J., Bailey, J.: Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. Journal of Machine Learning Research **11**(95), 2837–2854 (2010), http://jmlr.org/papers/v11/vinh10a.html
50. Zhang, P., Moore, C.: Scalable detection of statistically significant communities and hierarchies, using message passing for modularity. Proceedings of the National Academy of Sciences **111**(51), 18144–18149 (2014)
51. Zhao, X., Liang, J., Wang, J.: A community detection algorithm based on graph compression for large-scale social networks. Information Sciences **551**, 358–372 (2021)

Appendix

The data on 20 random graphs used in this study are available in a *FigShare* data repository [5]. The 60 real networks are loaded as simple undirected graphs. They are available in the publicly accessible network repository Netzschleuder with the 60 names below:

dom, packet_delays, sa_companies, ambassador, florentine_families, rhesus_monkey, kangaroo, internet_top_pop, high_tech_company, moviegalaxies, november17, moreno_taro, sp_baboons, bison, dutch_school, zebras, cattle, moreno_sheep, 7th_graders, college_freshmen, hens, freshmen, karate, dutch_criticism, montreal, ceo_club, windsurfers, elite, macaque_neural, sp_kenyan_households, contiguous_usa, cs_department, dolphins, macaques, terrorists_911, train_terrorists, highschool, law_firm, baseball, blumenau_drug, lesmis, fresh_webs, sp_office, swingers, polbooks, game_thrones, football, football_tsevans, sp_high_school_new, foodweb_baywet, revolution, foodweb_little_rock, student_cooperation, jazz_collab, interactome_pdz, physician_trust, malaria_genes, marvel_partnerships, facebook_friends, netscience

For more information on each network and its original source, one may check the Netzschleuder website by adding the network name at the end of the url: https://networks.skewed.de/net/. For example, https://networks.skewed.de/net/malaria_genes provides additional information for the *malaria_genes* network. In cases of multiple networks existing with the same name in Netzschleuder, we have used the first network (e.g. we have used the *HVR_1* network from https://networks.skewed.de/net/malaria_genes).

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36027-5_48