On Modularity - NP-Completeness and Beyond*

Ulrik Brandes[1], Daniel Delling[2], Marco Gaertler[2], Robert Görke[2], Martin Hoefer[1], Zoran Nikoloski[3], and Dorothea Wagner[2]

[1] Department of Computer & Information Science, University of Konstanz,
{brandes,hoefer}@inf.uni-konstanz.de
[2] Faculty of Informatics, Universität Karlsruhe (TH),
{delling,gaertler,rgoerke,wagner}@informatik.uni-karlsruhe.de
[3] Department of Applied Mathematics, Faculty of Mathematics and Physics,
Charles University, Prague, nikoloski@kam.mff.cuni.cz

Abstract. Modularity is a recently introduced quality measure for graph clusterings. It has immediately received considerable attention in several disciplines, and in particular in the complex systems literature, although its properties are not well understood. We here present first results on the computational and analytical properties of modularity. The complexity status of modularity maximization is resolved showing that the corresponding decision version is $\mathcal{NP}$-complete in the strong sense. We also give a formulation as an Integer Linear Program (ILP) to facilitate exact optimization, and provide results on the approximation factor of the commonly used greedy algorithm. Completing our investigation, we characterize clusterings with maximum modularity for several graph families.

1 Introduction

Graph clustering is a fundamental problem in the analysis of relational data. Although studied for decades, it applied in many settings, it is now popularly referred to as the problem of partitioning networks into communities. In this line of research, a novel graph clustering index called *modularity* has been proposed recently [1]. The rapidly growing interest in this measure prompted a series of follow-up studies on various applications and possible adjustments (see, e.g., [2,3,4,5,6]). Moreover, an array of heuristic algorithms has been proposed to optimize modularity. These are based on a greedy agglomeration [7,8], on spectral division [9,10], simulated annealing [11,12], or extremal optimization [13] to name but a few prominent examples. While these studies often provide plausibility arguments in favor of the resulting partitions, we know of only one attempt to characterize properties of clusterings with maximum modularity [2]. In particular, none of the proposed algorithms has been shown to produce optimal partitions with respect to modularity.

In this paper we provide first complexity-theoretic results by showing the hardness of modularity maximization, thereby justifying the use of heuristic approaches. Moreover, we cast the problem as an Integer Linear Program to facilitate exact optimization beyond enumeration of all clusterings, give a characterization of clusterings with maximum modularity for several graph families, and investigate the worst-case behavior of the popular greedy agglomerative approach. In fact, we give a graph family for which the greedy approach yields an approximation factor no better than two. In addition, our examples indicate that the quality of the greedy clustering may heavily depend on its tie-breaking strategy. Under worst-case conditions this can yield an infinitely large approximation

* This work was partially supported by the DFG under grants BR 2158/2-3, WA 654/14-3, Research Training Group 1042 ”Explorative Analysis and Visualization of Large Information Spaces” and by EU under grant DELIS (contract no. 001907).

factor. These performance studies are concluded by comparing greedy clusterings from previous publications with the optimum, which yields further insight.

This paper is organized as follows. Section 2 shortly introduces preliminaries, formulations of modularity and basic properties. Our $\mathcal{NP}$-completeness proofs are given in Section 3, followed by an analysis of the greedy approach in Section 4. The theoretical investigation is extended by the ILP formulation, and characterizations of the optimum clusterings for cliques and cycles in Section 5. Our work is concluded by revisiting examples from previous work in Section 6 and a brief discussion in Section 7.

2 Preliminaries

Throughout this paper, we will use the notation of [14]. More precisely, we assume that $G = (V, E)$ is an undirected connected graph with $n := |V|$ vertices, $m := |E|$ edges. Denote by $\mathcal{C} = \{C_1, \ldots, C_k\}$ a partition of V. We call $\mathcal{C}$ a *clustering* of G and the C_i, which are required to be non-empty, *clusters*; $\mathcal{C}$ is called *trivial* if either $k = 1$ or $k = n$. We denote the set of all possible clusterings of a graph G with $\mathcal{A}(G)$. In the following, we often identify a cluster C_i with the induced subgraph of G, i. e., the graph $G[C_i] := (C_i, E(C_i))$, where $E(C_i) := \{\{v, w\} \in E : v, w \in C_i\}$. Then $E(\mathcal{C}) := \bigcup_{i=1}^{k} E(C_i)$ is the set of *intra-cluster edges* and $E \setminus E(\mathcal{C})$ the set of *inter-cluster edges*. The number of intra-cluster edges is denoted by $m(\mathcal{C})$ and the number of inter-cluster edges by $\overline{m}(\mathcal{C})$. The set of edges that have one end-node in C_i and the other end-node in C_j is denoted by $E(C_i, C_j)$.

2.1 Definition of Modularity

Modularity is a quality index for clusterings. Given a simple graph $G = (V, E)$, we follow [1] and define the *modularity* $\mathsf{q}(\mathcal{C})$ of a clustering $\mathcal{C}$ as

$$\mathsf{q}(\mathcal{C}) := \sum_{C \in \mathcal{C}} \left[\frac{|E(C)|}{m} - \left(\frac{|E(C)| + \sum_{C' \in \mathcal{C}} |E(C, C')|}{2m} \right)^2 \right] . \tag{1}$$

Note that C' ranges over all clusters, so that edges in $E(C)$ are counted twice in the squared expression. This is to adjust proportions, since edges in $E(C, C')$, $C \neq C'$, are counted twice as well, once for each ordering of the arguments. Note that we can rewrite Equation (1) into the more convenient form

$$\mathsf{q}(\mathcal{C}) = \sum_{C \in \mathcal{C}} \left[\frac{|E(C)|}{m} - \left(\frac{\sum_{v \in C} \deg(v)}{2m} \right)^2 \right] . \tag{2}$$

This reveals an inherent trade-off: To maximize the first term, many edges should be contained in clusters, whereas the minimization of the second term is achieved by splitting the graph into many clusters with small total degrees each. Note that the first term $|E(\mathcal{C})|/m$ is also known as *coverage* [14].

2.2 Basic Properties

The definition of modularity exhibits several basic properties, which are interesting by themselves, but will also be useful later on. First, we focus on the range of modularity, for which Lemma 1 gives the lower and upper bound.

Lemma 1. *Let G be an undirected and unweighted graph and $\mathcal{C} \in \mathcal{A}(G)$. Then $-1/2 \leq \mathsf{q}(\mathcal{C}) \leq 1$ holds.*

Proof. Let $m_i = |E(C)|$ be the number of edges inside cluster C and $m_e = \sum_{C \neq C' \in \mathcal{C}} |E(C, C')|$ be the number of edges having exactly one end-node in C. Then the contribution of C to $\mathsf{q}(\mathcal{C})$ is:

$$\frac{m_i}{m} - \left(\frac{m_i}{m} + \frac{m_e}{2m}\right)^2 \quad .$$

This expression is strictly decreasing in m_e and, when varying m_i, the only maximum point is at $m_i = (m - m_e)/2$. The contribution of a cluster is minimized when m_i is zero and m_e is as large as possible. Suppose now $m_i = 0$, using the inequality $(a+b)^2 \geq a^2 + b^2$ for all non-negative numbers a and b, modularity has a minimum score for two clusters where all edges are inter-cluster edges. The upper bound is obvious from our reformulation in Eq. (2), and has been observed previously [2,3,15]. It can only be actually attained in the specific case of a graph with no edges, where coverage is defined to be 1.

As a result, any bipartite graph $K_{a,b}$ with the canonic clustering $\mathcal{C} = \{C_a, C_b\}$ yields the minimum modularity of $-1/2$. The following four results characterize the structure of a clustering with maximum modularity.

Corollary 1. *Isolated nodes have no impact on modularity.*

Corollary 1 directly follows from the fact that modularity depends on edges and degrees, thus, an isolated node does not contribute, regardless of its association to a cluster. Therefore, we exclude isolated nodes from further consideration in this work, i. e., all nodes are assumed to be of degree greater than zero.

Lemma 2. *A clustering with maximum modularity has no cluster that consists of a single node with degree 1.*

Proof. Suppose for contradiction that there is a clustering $\mathcal{C}$ with a cluster $C_v = \{v\}$ and $\deg(v) = 1$. Consider a cluster C_u that contains the neighbor node u. Suppose there are a number of m_i intra-cluster edges in C_u and m_e inter-cluster edges connecting C_u to other clusters. Together these clusters add

$$\frac{m_i}{m} - \frac{(2m_i + m_e)^2 + 1}{4m^2}$$

to $\mathsf{q}(\mathcal{C})$. Merging C_v with C_u results in a new contribution of

$$\frac{m_i + 1}{m} - \frac{(2m_i + m_e + 1)^2}{4m^2}$$

The merge yields an increase of

$$\frac{1}{m} - \frac{2m_i + m_e}{2m^2} > 0$$

in modularity, because $m_i + m_e \leq m$ and $m_e \geq 1$. This proves the lemma.

Lemma 3. *There is always a clustering with maximum modularity, in which each cluster consists of a connected subgraph.*

Proof. Consider for contradiction a clustering $\mathcal{C}$ with a cluster C of m_i intra- and m_e inter-cluster edges that consists of a set of more than one connected subgraph. The subgraphs in C do not have to be disconnected in G, they are only disconnected when we consider the edges $E(C)$. Cluster C adds

$$\frac{m_i}{m} - \frac{(2m_i + m_e)^2}{4m^2}$$

to $\mathsf{q}(\mathcal{C})$. Now suppose we create a new clustering $\mathcal{C}'$ by splitting C into two new clusters. Let one cluster C_v consist of the component including node v, i.e. all nodes, which can be reached from a node v with a path running only through nodes of C, i.e. $C_v = \bigcup_{i=1}^{\infty} C_v^i$, where $C_v^i = \{w \mid \exists(w, w_i) \in E(C) \text{ with } w_i \in C_v^{i-1}\}$ and $C_v^0 = \{v\}$. The other nonempty cluster is given by $C - C_v$. Let C_v have m_i^v intra- and m_e^v inter-cluster edges. Together the new clusters add

$$\frac{m_i}{m} - \frac{(2m_i^v + m_e^v)^2 + (2(m - m_i^v) + m - m_e^v)^2}{4m^2}$$

to $\mathsf{q}(\mathcal{C}')$. For $a, b \geq 0$ obviously $a^2 + b^2 \leq (a + b)^2$, and hence $\mathsf{q}(\mathcal{C}') \geq \mathsf{q}(\mathcal{C})$.

Corollary 2. *A clustering of maximummodularity does not include disconnected clusters.*

Corollary 2 directly follows from Lemma 3 and from the exclusion of isolated nodes. Thus, the search for an optimum can be restricted to clusterings, in which clusters are connected subgraphs and there are no clusters consisting of nodes with degree 1.

2.3 Counterintuitive Behavior

In the last section, we confirmed some intuitive properties like connectivity within clusters for clusterings of maximum modularity. Due to the trade-off between coverage and the sums of squared cluster degrees, the index also exhibits some counterintuitive behavior.

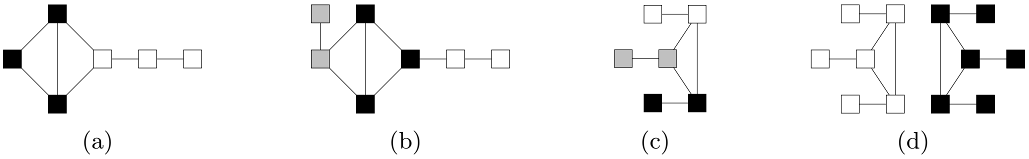


Fig. 1. (a,b) Non-local behavior; (c) a clique K_3 with leaves; (d) scaling behavior. Clusters are represented by colours.

At first glance, modularity seems to be a local quality measure. Recalling Equation (2), each cluster contributes separately. However, the example presented in Figures 1(a) and 1(b) exhibit a typical non-local behavior. In these figures, clusters are represented by color. By adding an additional node connected to the leftmost node, the optimal clustering is altered completely. According to Lemma 2 the additional node has to be clustered together with the leftmost node. This leads to a shift of the leftmost white node from the white cluster to the black cluster, although locally its neighborhood structure has not changed.

A *clique with leaves* is a graph of $2n$ nodes that consists of a clique K_n and n *leaf* nodes of degree one, such that each node of the clique is connected to exactly one leaf node. For a clique we show in Section 5 that the trivial clustering with $k = 1$ has maximum modularity. For a clique with leaves, however, the optimal clustering changes to $k = n$ clusters, in which each cluster consists of a connected pair of leaf and clique nodes. Figure 1(c) shows an example.

Figures 1(c) and 1(d) display the scaling behavior of modularity. By simply doubling the graph presented in Figure 1(c), the optimal clustering is altered completely. While in Figure 1(c) we obtain three clusters each consisting of the minor K_2, the clustering with maximum modularity of the graph in Figure 1(d) consists of two clusters, each being a graph equal to the one in Figure 1(c).

This behavior is in line with the previous observations in [2,4], where it was observed that size and structure of clusters in the optimum clustering depend on the total number of links in the network. Hence, clusters that are identified in smaller graphs might be combined to a larger cluster in a optimum clustering of a larger graph. The formulation of Eq. 2 mathematically explains this observation as modularity optimization strives to optimize the trade-off between coverage and degree sums. This provides a rigorous understanding of the observations made in [2,4].

3 $\mathcal{NP}$-Completeness

To formulate our complexity-theoretic result, we consider the following decision problem underlying modularity maximization.

Problem 1 (MODULARITY) *Given a graph G and a number K, is there a clustering $\mathcal{C}$ of G, for which $\mathsf{q}(\mathcal{C}) \geq K$?*

Note that we may ignore the fact that, in principle, K could be a real number in the range $[-1/2, 1]$, because $4m^2 \cdot \mathsf{q}(\mathcal{C})$ is integer for every partition $\mathcal{C}$ of G and polynomially bounded in the size of G. Our hardness result for MODULARITY is based on a transformation from the following decision problem.

Problem 2 (3-PARTITION) *Given $3k$ positive integer numbers $a_1, \ldots, a_{3k}$ such that the sum $\sum_{i=1}^{3k} a_i = kb$ and $b/4 < a_i < b/2$ for an integer b and for all $i = 1, \ldots, 3k$, is there a partition of these numbers into k sets, such that the numbers in each set sum up to b?*

We show that an instance $A = \{a_1, \ldots, a_{3k}\}$ of 3-PARTITION can be transformed into an instance $(G(A), K(A))$ of MODULARITY, such that $G(A)$ has a clustering with modularity at least $K(A)$, if and only if $a_1, \ldots, a_{3k}$ can be partitioned into k sets of sum $b = 1/k \cdot \sum_{i=1}^{k} a_i$ each.

It is crucial that 3-PARTITION is *strongly* $\mathcal{NP}$-complete [16], i.e. the problem remains $\mathcal{NP}$-complete even if the input is represented in unary coding. This implies that no algorithm can decide the problem in time polynomial even in the sum of the input values, unless $\mathcal{P} = \mathcal{NP}$. More importantly, it implies that our transformation need only be pseudo-polynomial.

The reduction is defined as follows. Given an instance A of 3-PARTITION, construct a graph $G(A)$ with k cliques (completely connected subgraphs) $H_1, \ldots, H_k$ of size $a =$

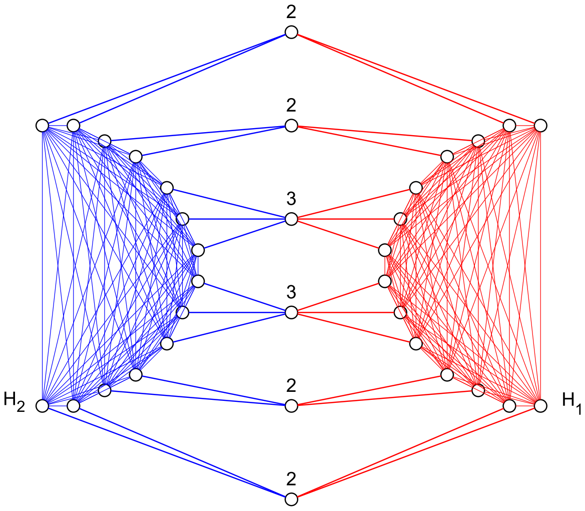


Fig. 2. An example graph $G(A)$ for the instance $A = \{2,2,2,2,3,3\}$ of 3-Partition. Node labels indicate the corresponding numbers $a_i \in A$.

$\sum_{i=1}^{3k} a_i$ each. For each element $a_i \in A$ we introduce a single *element node*, and connect it to a_i nodes in each of the k cliques in such a way that each clique member is connected to exactly one element node. It is easy to see that each clique node then has degree a and the element node corresponding to element $a_i \in A$ has degree ka_i. The number of edges in $G(A)$ is $m = k/2 \cdot a(a+1)$. See Figure 2 for an example. Note that the size of $G(A)$ is polynomial in the unary coding size of A, so that our transformation is indeed pseudo-polynomial.

Before specifying bound $K(A)$ for the instance of Modularity, we will show three properties of maximum modularity clusterings of $G(A)$. Together these properties establish the desired characterization of solutions for 3-Partition by solutions for Modularity.

Lemma 4. *In a maximum modularity clustering of $G(A)$, none of the cliques $H_1, \dots, H_k$ is split.*

We prove the lemma by showing that every clustering that violates the above condition can be modified in order to strictly improve modularity.

Proof. We consider a clustering $\mathcal{C}$ that splits a clique $H \in \{H_1, \dots, H_k\}$ into different clusters and then show how to obtain a clustering with strictly higher modularity. Suppose that $C_1, \dots, C_r \in \mathcal{C}$, $r > 1$, are the clusters that contain nodes of H. For $i = 1, \dots, r$ we denote by n_i the number of nodes of H contained in cluster C_i, $m_i = |E(C_i)|$ the number edges between nodes in C_i, f_i the number of edges between nodes of H in C_i and element nodes in C_i, d_i be the sum of degrees of all nodes in C_i. The contribution of $C_1, \dots, C_r$ to $\mathsf{q}(\mathcal{C})$ is

$$\frac{1}{m}\sum_{i=1}^{r} m_i - \frac{1}{4m^2}\sum_{i=1}^{r} d_i^2 \quad .$$

Now suppose we create a clustering $\mathcal{C}'$ by rearranging the nodes in $C_1, \dots, C_r$ into clusters $C', C'_1, \dots, C'_r$, such that C' contains exactly the nodes of clique H, and each C'_i, $1 \le i \le r$, the remaining elements of C_i (if any). In this new clustering the number of covered edges reduces by $\sum_{i=1}^{r} f_i$, because all nodes from H are removed from the clusters C'_i. This labels the edges connecting the clique nodes to other non-clique nodes of C_i as inter-cluster edges. For H itself there are $\sum_{i=1}^{r}\sum_{j=i+1}^{r} n_i n_j$ edges that are now additionally covered due to the creation of cluster C'. In terms of degrees the new cluster C' contains a nodes of degree a. The sums for the remaining clusters C'_i are reduced by the degrees of the clique nodes, as these nodes are now in C'. So the contribution of these clusters to $\mathsf{q}\left(\mathcal{C}'\right)$ is given by

$$\frac{1}{m}\sum_{i=1}^{r}\left(m_i + \sum_{j=i+1}^{r} n_i n_j - f_i\right) - \frac{1}{4m^2}\left(a^4 + \sum_{i=1}^{r}(d_i - n_i a)^2\right) \quad .$$

Setting $\varDelta := \mathsf{q}\left(\mathcal{C}'\right) - \mathsf{q}\left(\mathcal{C}\right)$, we obtain

$$\begin{aligned}\varDelta &= \frac{1}{m}\left(\sum_{i=1}^{r}\sum_{j=i+1}^{r} n_i n_j - f_i\right) + \frac{1}{4m^2}\left(\left(\sum_{i=1}^{r} 2d_i n_i a - n_i^2 a^2\right) - a^4\right)\\ &= \frac{1}{4m^2}\left(4m\sum_{i=1}^{r}\sum_{j=i+1}^{r} n_i n_j - 4m\sum_{i=1}^{r} f_i + \left(\sum_{i=1}^{r} n_i\left(2d_i a - n_i a^2\right)\right) - a^4\right) \quad .\end{aligned}$$

Using the equation that $2\sum_{i=1}^{r}\sum_{j=i+1}^{r} n_i n_j = \sum_{i=1}^{r}\sum_{j\neq i} n_i n_j$, substituting $m = \frac{k}{2}a(a+1)$ and rearranging terms we get

$$\begin{aligned}\varDelta &= \frac{a}{4m^2}\left(-a^3 - 2k(a+1)\sum_{i=1}^{r} f_i + \sum_{i=1}^{r} n_i\left(2d_i - n_i a + k(a+1)\sum_{j\neq i} n_j\right)\right)\\ &\geq \frac{a}{4m^2}\left(-a^3 - 2k(a+1)\sum_{i=1}^{r} f_i + \sum_{i=1}^{r} n_i\left(n_i a + 2kf_i + k(a+1)\sum_{j\neq i}^{r} n_j\right)\right) .\end{aligned}$$

For the last inequality we use the fact that $d_i \geq n_i a + kf_i$. This inequality holds because C_i contains at least the n_i nodes of degree a from the clique H. In addition, it contains both the clique and element nodes for each edge counted in f_i. For each such edge there are $k-1$ other edges connecting the element node to the $k-1$ other cliques. Hence, we get a contribution of kf_i in the degrees of the element nodes. Combining the terms n_i

and one of the terms $\sum_{j \neq i} n_j$ we obtain

$$\begin{aligned}
\Delta \geq \; & \frac{a}{4m^2}\left(-a^3 - 2k(a+1)\sum_{i=1}^{r} f_i\right) \\
& + \frac{a}{4m^2}\left(\sum_{i=1}^{r} n_i \left(a\sum_{j=1}^{r} n_j + 2kf_i + ((k-1)a+k)\sum_{j\neq i}^{r} n_j\right)\right) \\
= \; & \frac{a}{4m^2}\left(-2k(a+1)\sum_{i=1}^{r} f_i + \sum_{i=1}^{r} n_i \left(2kf_i + ((k-1)a+k)\sum_{j\neq i}^{r} n_j\right)\right) \\
= \; & \frac{a}{4m^2}\left(\sum_{i=1}^{r} 2kf_i(n_i - a - 1)) + ((k-1)a+k)\sum_{i=1}^{r}\sum_{j\neq i}^{r} n_i n_j\right) \\
\geq \; & \frac{a}{4m^2}\left(\sum_{i=1}^{r} 2kn_i(n_i - a - 1) + ((k-1)a+k)\sum_{i=1}^{r}\sum_{j\neq i}^{r} n_i n_j\right) \quad ,
\end{aligned}$$

For the last step we note that $n_i \leq a-1$ and $n_i - a - 1 < 0$ for all $i = 1, \ldots, r$. So increasing f_i decreases the modularity difference. For each node of H there is at most one edge to a node not in H, and thus $f_i \leq n_i$.
By rearranging terms and using the inequality $a \geq 3k$ we get

$$\begin{aligned}
\Delta \geq \; & \frac{a}{4m^2}\sum_{i=1}^{r} n_i \left(2k(n_i - a - 1) + ((k-1)a+k)\sum_{j\neq i}^{r} n_j\right) \\
= \; & \frac{a}{4m^2}\sum_{i=1}^{r} n_i \left(-2k + ((k-1)a-k)\sum_{j\neq i}^{r} n_j\right) \\
\geq \; & \frac{a}{4m^2}((k-1)a - 3k)\sum_{i=1}^{r}\sum_{j\neq i}^{r} n_i n_j \\
\geq \; & \frac{3k^2}{4m^2}(3k-6)\sum_{i=1}^{r}\sum_{j\neq i}^{r} n_i n_j \quad .
\end{aligned}$$

As we can assume $k > 2$ for all relevant instances of 3-Partition, we obtain $\Delta > 0$. This shows that any clustering can be improved by merging each clique completely into a cluster.

Next, we observe that the optimum clustering places at most one clique completely into a single cluster.

Lemma 5. *In a maximum modularity clustering of $G(A)$, every cluster contains at most one of the cliques $H_1, \ldots, H_k$.*

Proof. Consider a maximum modularity clustering. Lemma 4 shows that each of the k cliques $H_1, \ldots, H_k$ is entirely contained in one cluster. Assume that there is a cluster C which contains at least two of the cliques. If C does not contain any element nodes, then the cliques form disconnected components in the cluster. In this case it is easy to see that

the clustering can be improved by splitting C into distinct clusters, one for each clique. In this way we keep the number of edges within clusters the same, however, we reduce the squared degree sums of clusters.
Otherwise, we assume C contains $l > 1$ cliques completely and in addition some element nodes of elements a_j with $j \in J \subseteq \{1, \ldots, k\}$. Note that inside the l cliques $la(a-1)/2$ edges are covered. In addition, for every element node corresponding to an element a_j there are la_j edges included. The degree sum of the cluster is given by the la clique nodes of degree a and some number of element nodes of degree ka_j. The contribution of C to $\mathsf{q}(\mathcal{C})$ is thus given by

$$\frac{1}{m}\left(\frac{l}{2}a(a-1) + l\sum_{j\in J} a_j\right) - \frac{1}{4m^2}\left(la^2 + k\sum_{j\in J} a_j\right)^2 .$$

Now suppose we create $\mathcal{C}'$ by splitting C into C'_1 and C'_2 such that C'_1 completely contains a single clique H. This leaves the number of edges covered within the cliques the same, however, all edges from H to the included element nodes eventually drop out. The degree sum of C'_1 is exactly a^2, and so the contribution of C'_1 and C'_2 to $\mathsf{q}(\mathcal{C}')$ is given by

$$\frac{1}{m}\left(\frac{l}{2}a(a-1) + (l-1)\sum_{j\in J} a_j\right) - \frac{1}{4m^2}\left(\left((l-1)a^2 + k\sum_{j\in J} a_j\right)^2 + a^4\right).$$

Considering the difference we note that

$$\begin{aligned}
\mathsf{q}(\mathcal{C}') - \mathsf{q}(\mathcal{C}) &= -\frac{1}{m}\sum_{j\in J} a_j + \frac{1}{4m^2}\left((2l-1)a^4 + 2ka^2\sum_{j\in J} a_j - a^4\right) \\
&= \frac{2(l-1)a^4 + 2ka^2\sum_{j\in J} a_j - 4m\sum_{j\in J} a_j}{4m^2} \\
&= \frac{2(l-1)a^4 - 2ka\sum_{j\in J} a_j}{4m^2} \\
&\geq \frac{9k^3}{2m^2}(9k-1) \\
&> 0,
\end{aligned}$$

as $k > 0$ for all instances of 3-Partition.

Since the clustering is improved in every case, it is not optimal. This is a contradiction.

The previous two lemmas show that any clustering can be strictly improved to a clustering that contains k *clique clusters*, such that each one completely contains one of the cliques $H_1, \ldots, H_k$ (possibly plus some additional element nodes). In particular, this must hold for the optimum clustering as well. Now that we know how the cliques are clustered we turn to the element nodes.
As they are not directly connected, it is never optimal to create a cluster consisting only of element nodes. Splitting such a cluster into singleton clusters, one for each element node, reduces the squared degree sums but keeps the edge coverage at the same value. Hence, such a split yields a clustering with strictly higher modularity. The next lemma shows that we can further strictly improve the modularity of a clustering with a singleton cluster of an element node by joining it with one of the clique clusters.

Lemma 6. *In a maximum modularity clustering of $G(A)$, there is no cluster composed of element nodes only.*

Proof. Consider a clustering $\mathcal{C}$ of maximum modularity and suppose that there is an element node v_i corresponding to the element a_i, which is not part of any clique cluster. As argued above we can improve such a clustering by creating a singleton cluster $C = \{v_i\}$. Suppose C_{min} is the clique cluster, for which the sum of degrees is minimal. We know that C_{min} contains all nodes from a clique H and eventually some other element nodes for elements a_j with $j \in J$ for some index set J. The cluster C_{min} covers all $a(a-1)/2$ edges within H and $\sum_{j\in J} a_j$ edges to element nodes. The degree sum is a^2 for clique nodes and $k\sum_{j\in J} a_j$ for element nodes. As C is a singleton cluster, it covers no edges and the degree sum is ka_i. This yields a contribution of C and C_{min} to $\mathsf{q}(\mathcal{C})$ of

$$\frac{1}{m}\left(\frac{a(a-1)}{2}+\sum_{j\in J} a_j\right)-\frac{1}{4m^2}\left(\left(a^2+k\sum_{j\in J} a_j\right)^2+k^2a_i^2\right) \ .$$

Again, we create a different clustering $\mathcal{C}'$ by joining C and C_{min} to a new cluster C'. This increases the edge coverage by a_i. The new cluster C' has the sum of degrees of both previous clusters. The contribution of C' to $\mathsf{q}(\mathcal{C}')$ is given by

$$\frac{1}{m}\left(\frac{a(a-1)}{2}+a_i+\sum_{j\in J} a_j\right)-\frac{1}{4m^2}\left(a^2+ka_i+k\sum_{j\in J} a_j\right)^2 \ ,$$

so that

$$\begin{aligned}\mathsf{q}(\mathcal{C}')-\mathsf{q}(\mathcal{C}) &= \frac{a_i}{m}-\frac{1}{4m^2}\left(2ka^2a_i+2k^2a_i\sum_{j\in J} a_j\right)\\ &= \frac{1}{4m^2}\left(2ka(a+1)a_i-2ka^2a_i-2k^2a_i\sum_{j\in J} a_j\right)\\ &= \frac{a_i}{4m^2}\left(2ka-2k^2\sum_{j\in J} a_j\right).\end{aligned}$$

At this point recall that C_{min} is the clique cluster with the minimum degree sum. For this cluster the elements corresponding to included element nodes can never sum to more than a/k. In particular, as v_i is not part of any clique cluster, the elements of nodes in C_{min} can never sum to more than $(a-a_i)/k$. Thus,

$$\sum_{j\in J} a_j \leq \frac{1}{k}(a-a_i) < \frac{1}{k}a \ ,$$

and so $\mathsf{q}(\mathcal{C}')-\mathsf{q}(\mathcal{C}) > 0$. This contradicts the assumption that $\mathcal{C}$ is optimal.

We have shown that for the graphs $G(A)$ the clustering of maximum modularity consists of exactly k clique clusters, and each element node belongs to exactly one of the clique clusters. Combining the above results, we now state our main result:

Theorem 3. Modularity *is strongly $\mathcal{NP}$-complete.*

Proof. For a given clustering $\mathcal{C}$ of $G(A)$ we can check in polynomial time whether $\mathsf{q}(\mathcal{C}) \geq K(A)$, so clearly Modularity $\in \mathcal{NP}$.

For $\mathcal{NP}$-completeness we transform an instance $A = \{a_1, \ldots, a_{3k}\}$ of 3-Partition into an instance $(G(A), K(A))$ of Modularity. We have already outlined the construction of the graph $G(A)$ above. For the correct parameter $K(A)$ we consider a clustering in $G(A)$ with the properties derived in the previous lemmas, i. e., a clustering with exactly k clique clusters. Any such clustering yields exactly $(k-1)a$ inter-cluster edges, so the edge coverage is given by

$$\sum_{C \in \mathcal{C}} \frac{|E(C)|}{m} = \frac{m-(k-1)a}{m} = 1 - \frac{2(k-1)a}{ka(a+1)} = 1 - \frac{2k-2}{k(a+1)}.$$

Hence, the clustering $\mathcal{C} = (C_1, \ldots, C_k)$ with maximum modularity must minimize $\deg(C_1)^2 + \deg(C_2)^2 + \ldots + \deg(C_k)^2$. This requires a distribution of the element nodes between the clusters which is as even as possible with respect to the sum of degrees per cluster. In the optimum case we can assign to each cluster element nodes corresponding to elements that sum to $b = 1/k \cdot a$. In this case the sum up of degrees of element nodes in each clique cluster is equal to $k \cdot 1/k \cdot a = a$. This yields $\deg(C_i) = a^2 + a$ for each clique cluster C_i, $i = 1, \ldots, k$, and gives

$$\deg(C_1)^2 + \ldots + \deg(C_k)^2 \geq k(a^2+a)^2 = ka^2(a+1)^2.$$

Equality holds only in the case, in which an assignment of b to each cluster is possible. Hence, if there is a clustering $\mathcal{C}$ with $\mathsf{q}(\mathcal{C})$ of at least

$$K(A) = 1 - \frac{2k-2}{k(a+1)} - \frac{ka^2(a+1)^2}{k^2a^2(a+1)^2} = \frac{(k-1)(a-1)}{k(a+1)}$$

then we know that this clustering must split the element nodes perfectly to the k clique clusters. As each element node is contained in exactly one cluster, this yields a solution for the instance of 3-Partition. With this choice of $K(A)$ the instance $(G(A), K(A))$ of Modularity is satisfiable only if the instance A of 3-Partition is satisfiable.

Otherwise, suppose the instance for 3-Partition is satisfiable. Then there is a partition into k sets such that the sum over each set is $1/k \cdot a$. If we cluster the corresponding graph by joining the element nodes of each set with a different clique, we get a clustering of modularity $K(A)$. This shows that the instance $(G(A), K(A))$ of Modularity is satisfiable if the instance A of 3-Partition is satisfiable. This completes the reduction and proves the theorem.

This result naturally holds also for the straightforward generalization of maximizing modularity in weighted graphs [17]. Instead of using the numbers of edges the definition of modularity employs the sum of edge weights for edges within clusters, between clusters and in the total graph.

3.1 Special Case: Modularity with Bounded Number of Clusters

Next, we consider the two problems of computing the clustering with largest modularity that splits the graph into exactly or at most two clusters. Although these are two different problems, our hardness result will hold for both versions, hence, we define the problem cumulatively.

Problem 4 (k-Modularity) *Given a graph G and a number K, is there a clustering $\mathcal{C}$ of G into exactly/at most k clusters, for which $\mathsf{q}(\mathcal{C}) \geq K$?*

We provide a proof using a reduction that is similar to the one given recently for showing the hardness of the MinDisAgree[2] problem of correlation clustering [18]. We use the problem Minimum Bisection for Cubic Graphs (MB3) for the reduction:

Problem 5 (Minimum Bisection for Cubic Graphs) *Given a 3-regular graph G with n nodes and an integer c, is there a clustering into two clusters of $n/2$ nodes each such that it cuts at most c edges?*

This problem has been shown to be strongly $\mathcal{NP}$-complete in [19]. We construct an instance of 2-Modularity from an instance of MB3 as follows. For each node v from the graph $G = (V, E)$ we attach $n - 1$ new nodes and construct an n-clique. We denote these cliques as *cliq*(v) and refer to them as *node clique* for $v \in V$. Hence, in total we construct n different new cliques, and after this transformation each node from the original graph has degree $n + 2$. Note that a cubic graph with n nodes has exactly $1.5n$ edges. In our adjusted graph there are exactly $m = (n(n-1) + 3)n/2$ edges.

We will show that an optimum clustering which is denoted as $\mathcal{C}^*$ of 2-Modularity in the adjusted graph has exactly two clusters. Furthermore, such a clustering corresponds to a minimum bisection of the underlying MB3 instance. In particular, we give a bound K such that the MB3 instance has a bisection cut of size at most c if and only if the corresponding graph has 2-modularity at least K.

We begin by noting that there is always a clustering $\mathcal{C}$ with $\mathsf{q}(\mathcal{C}) > 0$. Hence, $\mathcal{C}^*$ must have exactly two clusters, as no more than two clusters are allowed. This serves to show that our proof works for both versions of 2-modularity, in which at most or exactly two clusters must be found.

Lemma 7. *For every graph constructed from a MB3 instance, there exists a clustering $\mathcal{C} = \{C_1, C_2\}$ such that $\mathsf{q}(\mathcal{C}) > 0$. In particular, the clustering $\mathcal{C}^*$ has two clusters.*

Proof. Consider the following partition into two clusters. We pick the nodes of *cliq*(v) for some $v \in V$ as C_1 and the remaining graph as C_2. Then

$$\begin{aligned}\mathsf{q}(\mathcal{C}) &= 1 - \frac{3}{m} - \frac{(n(n-1)+3)^2 + ((n-1)(n(n-1)+3))^2}{4m^2} \\ &= \frac{2n-2}{n^2} - \frac{3}{m} = \frac{2}{n} - \frac{2}{n^2} - \frac{3}{m} \\ &> 0 \ ,\end{aligned}$$

as $n \geq 4$ for every cubic graph. Hence $\mathsf{q}(\mathcal{C}) > 0$ and the lemma follows.

Next, we show that in an optimum clustering, all the nodes of one node clique *cliq*(v) are located in one cluster:

Lemma 8. *For every node $v \in V$ there exists a cluster $C \in \mathcal{C}^*$ such that $cliq(v) \subseteq C$.*

Proof. For contradiction we assume a node clique $cliq(v)$ for some $v \in V$ is split in two clusters C_1 and C_2 of the clustering $\mathcal{C} = \{C_1, C_2\}$. Let $k_i := |C_i \cap cliq(v)|$ be the number of nodes located in the corresponding clusters, with $1 \leq k_i \leq n-1$. Note that $k_2 = n - k_1$. In addition, we denote the sum of node degrees in both clusters excluding nodes from $cliq(v)$ by d_1 and d_2:

$$d_i = \sum_{u \in C_i, u \notin cliq(v)} \deg(u).$$

Without loss of generality assume that $d_1 \geq d_2$. Finally, we denote by m' the number of edges covered by the clusters C_1 and C_2.

We define a new clustering $\mathcal{C}'$ as $\{C_1 \setminus cliq(v), C_2 \cup cliq(v)\}$ and denote the difference of the modularity as $\mathit{\Delta} := \mathsf{q}(\mathcal{C}') - \mathsf{q}(\mathcal{C})$. We distinguish two cases depending in which cluster the node v was located with respect to $\mathcal{C}$: In the first case $v \in C_2$ and we obtain:

$$\begin{aligned}
\mathsf{q}(\mathcal{C}) &= \frac{m'}{m} - \frac{(d_1 + k_1(n-1))^2 + (d_2 + (n-k_1)(n-1) + 3)^2}{4m^2} \quad , \\
\mathsf{q}(\mathcal{C}') &= \frac{m' + k_1(n-k_1)}{m} - \frac{d_1^2 + (d_2 + n(n-1) + 3)^2}{4m^2} \text{ and} \\
\mathit{\Delta} &= \frac{k_1(n-k_1)}{m} - \frac{d_1^2 + (d_2 + n(n-1) + 3)^2}{4m^2} \\
&\quad + \frac{(d_1 + k_1(n-1))^2 + (d_2 + (n-k_1)(n-1) + 3)^2}{4m^2} \quad .
\end{aligned}$$

We simplify expression of $\mathit{\Delta}$ as follows:

$$\begin{aligned}
\mathit{\Delta} &= \frac{1}{4m^2}\Bigg(4mk_1(n-k_1) - d_1^2 - (d_2 + n(n-1) + 3)^2 \\
&\qquad + (d_1 + k_1(n-1))^2 + (d_2 + (n-k_1)(n-1) + 3)^2\Bigg) \\
&= \frac{1}{4m^2}\Bigg(4mk_1(n-k_1) + (2k_1^2 - 2nk_1)(n-1)^2 - 6k_1(n-1) \\
&\qquad + 2(d_1 - d_2)k_1(n-1)\Bigg) \\
&\geq \frac{k_1}{4m^2}(4m(n-k_1) - 2(n-k_1)(n-1)^2 - 6(n-1)) \quad .
\end{aligned}$$

We can bound the expression in the bracket in the following way by using the assumption that $d_1 \geq d_2$ and $1 \leq k_1 \leq n-1$:

$$(n-k_1)\Big(4m - 2(n-1)^2\Big) - 6(n-1) \geq (n-k_1)\Big(\underbrace{4m - 2(n-1)^2 - 6(n-1)}_{=:B}\Big) \qquad (3)$$

and, thus, it remains to show that $B > 0$. By filling in the value of m and using the facts that $2n^2(n-1) > 2(n-1)^2$ and $6n > 6(n-1)$ for all $n \geq 4$, we obtain $B > 0$ and thus modularity strictly improves if all nodes are moved from $cliq(v)$ to C_2.

In the second case the node $v \in C_1$ and we get the following equations:

$$\begin{aligned}
\mathsf{q}\left(\mathcal{C}\right) &= \frac{m'}{m} - \frac{(d_1 + k_1(n-1)+3)^2 + (d_2 + (n-k_1)(n-1))^2}{4m^2} \ , \\
\mathsf{q}\left(\mathcal{C}'\right) &= \frac{m' + k_1(n-k_1)}{m} - \frac{d_1^2 + (d_2 + n(n-1)+3)^2}{4m^2} \ , \text{ and} \\
\Delta &= \frac{k_1(n-k_1)}{m} - \frac{d_1^2 + (d_2 + n(n-1)+3)^2}{4m^2} \\
&\quad + \frac{(d_1 + k_1(n-1)+3)^2 + (d_2 + (n-k_1)(n-1))^2}{4m^2} \ .
\end{aligned}$$

We simplify expression of Δ as follows:

$$\begin{aligned}
4m^2\Delta &= 4mk_1(n-k_1) + (2k_1^2 - 2nk_1)(n-1)^2 - 6(n-k_1)(n-1) \\
&\quad +2(d_1 - d_2)(k_1(n-1)+3) \\
&\geq 4mk_1(n-k_1) - 2k_1(n-k_1)(n-1)^2 - 6(n-k_1)(n-1))
\end{aligned}$$

Recall $1 \leq k_1 \leq n-1$, and filling in the value of m, we obtain

$$4mk_1 - 2k_1(n-1)^2 - 6(n-1) = 2k_1(n^2(n-1) - (n-1)^2) + 6nk_1 - 6(n-1) > 0 \ ,$$

which holds for all $k_1 \geq 1$ and $n \geq 4$. Also in this case, modularity strictly improves if all nodes are moved from $cliq(v)$ to C_2.

The final lemma before defining the appropriate input parameter K for the 2-Modularity and thus proving the correspondence between the two problem shows that the clusters in the optimum clusterings have the same size.

Lemma 9. *In $\mathcal{C}^*$, each cluster contains exactly $n/2$ complete node cliques.*

Proof. Suppose for contradiction that one cluster C_1 has $l_1 < n/2$ cliques. For completeness of presentation we use m' to denote the unknown (and irrelevant) number of edges covered by the clusters. For the modularity of the clustering is given in Equation (4).

$$\mathsf{q}\left(\mathcal{C}^*\right) = \frac{m'}{m} - \frac{l_1^2(n(n-1)+3)^2 + (n-l_1)^2(n(n-1)+3)^2}{4m^2} \tag{4}$$

We create a new clustering $\mathcal{C}'$ by transferring a complete node clique from cluster C_2 to cluster C_1. As the graph G is 3-regular, we lose at most 3 edges in the coverage part of modularity:

$$\mathsf{q}\left(\mathcal{C}'\right) \geq \frac{m'-3}{m} - \frac{(l_1+1)^2(n(n-1)+3)^2 + (n-l_1-1)^2(n(n-1)+3)^2}{4m^2} \ . \tag{5}$$

We can bound the difference in the following way:

$$\begin{aligned}
\mathsf{q}\left(\mathcal{C}'\right) - \mathsf{q}\left(\mathcal{C}\right) &\geq -\frac{3}{m} + \frac{(l_1^2 + (n-l_1)^2 - (l_1+1)^2 - (n-l_1-1)^2)(n(n-1)+3)^2}{4m^2} \\
&= -\frac{3}{m} + \frac{(2n - 4l_1 - 2)}{n^2} \\
&\geq -\frac{3}{m} + \frac{2}{n^2} = \frac{2}{n^2} - \frac{6}{n^3 - n^2 + 3n} \\
&> 0 \ ,
\end{aligned}$$

for all $n \geq 4$. The analysis uses the fact that we can assume n to be an even number, so $l_1 \leq \frac{n}{2} - 1$ and thus $4l_1 \leq 2n - 4$.

This shows that we can improve every clustering by balancing the number of complete node cliques in the clusters – independent of the loss in edge coverage.

Finally, we can state theorem about the complexity of 2-Modularity:

Theorem 6. *2-*Modularity *is $\mathcal{NP}$-complete (in the strong sense).*

Proof. Let (G, c) be an instance of Minimum Bisection for Cubic Graphs, then we construct a new graph G' as stated above and define $K := 1/2 - c/m$.

As we have shown in Lemma 9 that each cluster of $\mathcal{C}^*$ that is an optimum clustering of G' with respect to 2-Modularity has exactly $n/2$ complete node cliques, the sum of degrees in the clusters is exactly m. Thus, it is easy to see that if the clustering $\mathcal{C}^*$ meets the following inequality

$$\mathsf{q}\left(\mathcal{C}^*\right) \geq 1 - \frac{c}{m} - \frac{2m^2}{4m^2} = \frac{1}{2} - \frac{c}{m} = K \enspace ,$$

then the number of inter-cluster edges can be at most c. Thus the clustering $\mathcal{C}^*$ induces a balanced cut in G with at most c cut edges.

This proof is particularly interesting as it highlights that maximizing modularity in general is hard due to the hardness of minimizing the squared degree sums on the one hand, whereas in the case of two clusters this is due to the hardness of minimizing the edge cut.

4 The Greedy Algorithm

Modularity was originally introduced as a selection criterion for a divisive hierarchical clustering algorithm [1] and later used to define a greedy, agglomerative clustering algorithm [8]. The greedy algorithm starts with the singleton clustering and iteratively merges those two clusters that yield a clustering with the best modularity, i. e., the largest increase or the smallest decrease is chosen. After $n-1$ merges the clustering that achieved the highest modularity is returned. The algorithm maintains a symmetric matrix Δ with entries $\Delta_{i,j} := \mathsf{q}\left(\mathcal{C}_{i,j}\right) - \mathsf{q}\left(\mathcal{C}\right)$, where $\mathcal{C}$ is the current clustering and $\mathcal{C}_{i,j}$ is obtained from $\mathcal{C}$ by merging clusters C_i and C_j. Note that there can be several pairs i and j such that $\Delta_{i,j}$

Algorithm 1: Greedy Algorithm for Maximizing Modularity

```
Input: graph $G = (V, E)$
Output: clustering $\mathcal{C}$ of $G$
$\mathcal{C} \leftarrow$ singletons
initialize matrix $\Delta$
while $|\mathcal{C}| > 1$ do
    find $\{i, j\}$ with $\Delta_{i,j}$ is the maximum entry in the matrix $\Delta$
    merge clusters $i$ and $j$
    update $\Delta$
return clustering with highest modularity
```

is the maximum, in these cases the algorithm selects an arbitrary pair. The pseudo-code for the greedy algorithm is given in Algorithm 1. An efficient implementation using sophisticated data-structures requires $\mathcal{O}\left(n^2 \log n\right)$ runtime. Note that, $n-1$ iterations is an upper bound and one can terminate the algorithms when the matrix Δ contains only non-positive entries. We call this property *single-peakedness*, it is proven in [8]. Since it is $\mathcal{NP}$-hard to maximize modularity in general graphs, it is unlikely that this greedy algorithm is optimal. In fact, we sketch a graph family, where the above greedy algorithm has an approximation factor of 2, asymptotically. In order to prove this statement, we introduce a general construction scheme given in Definition 1. Furthermore, we point out instances where a specific way of breaking ties of merges yield a clustering with modularity of 0, while the optimum clustering has a strictly positive score.

Modularity is defined such that it takes values in the interval $[-1/2, 1]$ for any graph and any clustering. In particular the modularity of a trivial clustering placing all vertices into a single cluster has a value of 0. We use this technical peculiarity to show that the greedy algorithm has an unbounded approximation ratio.

Theorem 7. *There is no finite approximation factor for the greedy algorithm for finding clusterings with maximum modularity.*

Proof. We present a class of graphs, on which the algorithm obtains a clustering of value 0, but for which the optimum clustering has value close to 1/2. A graph G of this class is given by two cliques (V_1, E_1) and (V_2, E_2) of size $|V_1| = |V_2| = n/2$, and $n/2$ matching edges E_m connecting each vertex from V_1 to exactly one vertex in V_2 and vice versa. See Figure 3 for an example with $n = 14$. Note that we can define modularity by associating weights $w(u, v)$ with every existing and non-existing edge in G as follows:

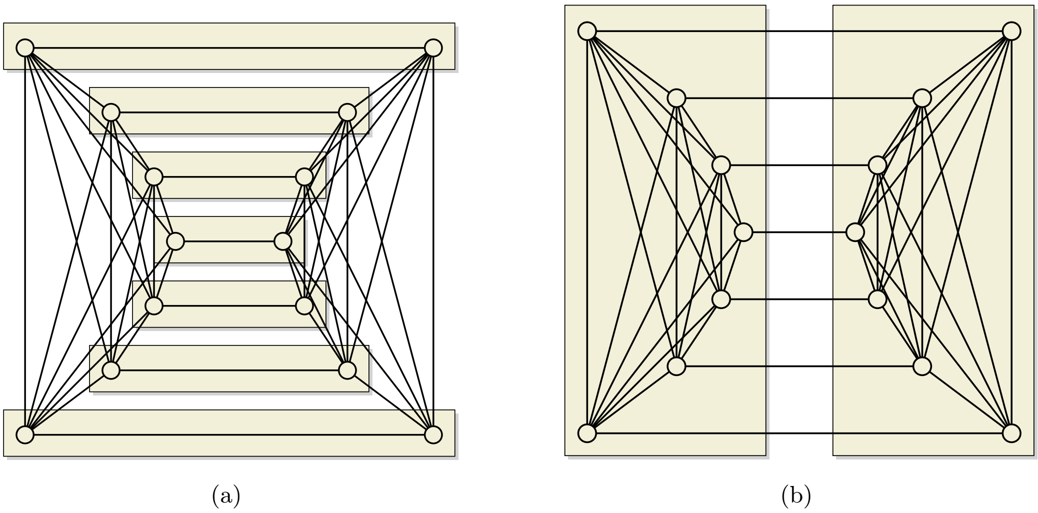
$$w(u,v) = \frac{E_{uv}}{2m} - \frac{\deg(u)\deg(v)}{4m^2} \quad ,$$

where $E_{uv} = 1$ if $(u, v) \in E$ and 0 otherwise. The modularity of a clustering $\mathcal{C}$ is then derived by the summing the weights of the edges covered by $\mathcal{C}$

$$\mathsf{q}\left(\mathcal{C}\right) = \sum_{C \in \mathcal{C}} \sum_{u,v \in C} w(u,v)$$

Note that in this formula we have to count twice the weight for each edge between different vertices u and v (once for every ordering) and once the weight for a non-existing self-loop for every vertex u. Thus, the change of modularity by merging two clusters is given by twice the sum of weights between the clusters.

Now consider a run of the greedy algorithm on the graph of Figure 3. Note that the graph is $n/2$-regular, and thus has $m = n^2/4$ edges. Each existing edge gets a weight of $2/n^2 - 1/n^2 = 1/n^2$, while every non-existing edge receives a weight of $-1/n^2$. As the self-loop is counted by every clustering, the initial trivial singleton clustering has modularity value of $-1/n$. In the first step each cluster merge along any existing edge results in an increase of $2/n^2$. Of all these equivalent possibilities we suppose the algorithm chooses to merge along an edge from E_m to create a cluster C'. In the second step merging a vertex with C' results in change of 0, because one existing and one non-existing edge would be included. Every other merge along an existing edge still has value $2/n^2$. We suppose the algorithm again chooses to merge two singleton clusters along an edge from E_m creating a cluster C''. Afterwards observe that merging clusters C' and C'' yields a change of 0, because two existing and two non-existing edges would be included. Thus, it is again optimal to merge two singleton clusters along an existing edge. If the algorithm continues to merge singleton clusters along the edges from E_m, it will in each iteration make an optimal merge resulting in strictly positive increase in modularity. After $n/2$ steps it has constructed a clustering $\mathcal{C}$ of the type depicted in Figure 3(a). $\mathcal{C}$ consists of one cluster for the vertices of each edge of E_m and has a modularity value of


(a)

(b)

Fig. 3. (a) Clustering with modularity 0; (b) Clustering with modularity close to $\frac{1}{2}$

$$\mathsf{q}(\mathcal{C}) = \frac{2}{n} - \frac{n}{2} \cdot \frac{4n^2}{n^4} = 0.$$

Due to the single-peakedness of the problem [8] all following cluster merges can never increase this value, hence the algorithm will return a clustering of value 0.

On the other hand consider a clustering $\mathcal{C}^* = \{C_1, C_2\}$ with two clusters, one for each clique $C_1 = V_1$ and $C_2 = V_2$ (see Figure 3(b)). This clustering has a modularity of

$$\mathsf{q}(\mathcal{C}^*) = \frac{n(n-2)}{n^2} - 2\frac{4n^2}{16n^2} = \frac{1}{2} - \frac{2}{n}.$$

This shows that the approximation ratio of the greedy algorithm can be infinitely large, because no finite approximation factor can outweigh a value of 0 with one strictly greater than 0.

The key observation is, that the proof considers a worst-case scenario in the sense that greedy is in each iteration supposed to pick exactly the ”worst” merge choice of several equivalently attractive alternatives. If greedy chooses in an early iteration to merge along an edge from E_1 or E_2, the resulting clustering will be significantly better. As mentioned earlier, this negative result is due to formulation of modularity, which yields values from the interval $[-1/2, 1]$. For instance, a linear remapping of the range of modularity to the interval $[0, 1]$, the greedy algorithm yields a value of $1/3$ compared to the new optimum score of $2/3$. In this case the approximation factor would be 2.

Next, we provide a decreased lower bound for a different class of graphs and no assumptions on the random choices of the algorithm.

Definition 1. *Let $G = (V, E)$ and $H = (V', E')$ be two non-empty, simple, undirected, and unweighted graphs and let $u \in V'$ be a node. The* product $G \star_u H$ *is defined as the graph (V'', E'') with the nodeset $V'' := V \cup V \times V'$ and the edgeset $E'' := E \cup E''_c \cup E''_H$ where*

$$\begin{aligned} E''_c &:= \{\ \{v, (v, u)\}\ \mid v \in V\} \qquad \textit{and} \\ E''_H &:= \{\ \{(v, v'), (v, w')\}\ \mid v \in V, v', w' \in V'', \{v', w'\} \in E\}\ \ . \end{aligned}$$

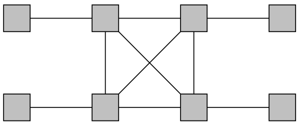


Fig. 4. The graph $K_4 \star_u P_1$.

An example is given in Figure 4. The product $G \star_u H$ is a graph that contains G and for each node v of G a copy H_v of H. For each copy the node in H_v corresponding to $u \in H$ is connected to v. We use the notation (v, w') to refer to the copy of node w' of H, which is located in H_v. In the following we consider only a special case: Let $n \geq 2$ be an integer, $H = (V', E')$ be an undirected and connected graph with at least two nodes, and $u \in V'$ an arbitrary but fixed node. We denote by $\mathcal{C}^g_k$ the clustering obtained with the greedy algorithm applied to $K_n \star_u H$ starting from singletons and performing at most k steps that all have a positive increase in modularity. Furthermore, let m be the number of edges in $K_n \star_u H$. Based on the merging policy of the greedy algorithm we can characterize the final clustering $\mathcal{C}^g_n$. It has n clusters, each of which includes a vertex v of G and his copy of H. More precisely, Lemma 10 and 11 describe the first occurring merges. In contrast, Lemma 12 states that after a certain number of specific merge-operations, no coarsening of the clustering yields a higher score of modularity.

Lemma 10. *If $2 \cdot |E'| < n$ and $\mathcal{C}^g_k$ has two clusters C_i and C_j such that both belong to the same copy of H and $E(C_i, C_j) \neq \emptyset$, then merging C_i and C_j increases the modularity.*

Proof. The difference of modularity after and before the merge is

$$\varDelta\mathsf{q}\,(C_i, C_j) := \frac{|E(C_i, C_j)|}{m} - \frac{\sum_{u \in C_i} \deg u \cdot \sum_{u' \in C_j} \deg u'}{2m^2}\ \ .$$

Since $|E(C_i, C_j)| \geq 1$ and $\sum_{u \in C_i \cup C_j} \deg u \leq 2|E'|+1$, we obtain the following inequalities:

$$\begin{aligned} \varDelta\mathsf{q}\,(C_i, C_j) &\geq \frac{1}{m} - \frac{\sum_{u \in C_i} \deg u \cdot \sum_{u' \in C_j} \deg u'}{2m^2} \\ &\geq \frac{1}{m} - \frac{(2|E'| + 1)^2}{2m^2}\ \ . \end{aligned}$$

Due to the fact that $2|E'| < n$, $m = \binom{n}{2} + n + n \cdot |E'|$, and $|E'| > 1$, we can conclude

$$m \cdot \varDelta\mathsf{q}\,(C_i, C_j) \geq 1 - \frac{n^2 + 2n + 1}{2n + 2\binom{n}{2} + 2n} \geq 0\ \ .$$

Lemma 11. *If $2 \cdot |E'| + 1 < n$ and $\mathcal{C}_k^g$ has the cluster $C := \{v\}$ and a cluster C_i containing only nodes of the v-copy of H and $u \in C_i$, then merging C_i and C increases the modularity. Furthermore such a merge increases the modularity by more than a merge of two clusters $\{w\}, \{w'\}$ for $w, w' \in V$ would.*

Proof. First note that $\sum_{w \in C_i} \deg w \leq 2|E'| + 1 \leq n - 1$. Thus the increase in modularity is

$$m \cdot \Delta\mathsf{q}(C_i, C) \geq 1 - \frac{n \cdot (n-1)}{2m} \geq 1 - \frac{n \cdot (n-1)}{n(n-1) + 4n} = 1 - \frac{n \cdot (n-1)}{n^2 + 3n} \geq 0$$

Analogous we obtain

$$m \cdot \Delta\mathsf{q}(\{v\}, \{w\}) = 1 - \frac{n^2}{2m} \geq 0$$

Since $n(n-1)/(2m) < n^2/(2m)$, we conclude $\Delta\mathsf{q}(C_i, C) \geq \Delta\mathsf{q}(\{v\}, \{w\}) \geq 0$.

Lemma 12. *If $2 \cdot |E'| + 1 < n$ and $\mathcal{C}_k^g$ has two clusters C_i and C_j each containing at least two nodes where (exactly) one belongs to V, then the merge of C_i and C_j is never executed.*

Proof. This proof consists of two parts. The first part shows that the merge of C_i and C_j yields a lower increase in modularity than merging H completely into one cluster, including the connecting node $v \in V$. The second part shows that two such clusters cannot be merged.

First, let $v \in V$ be the node of C_i. If $\{v\} \times V'$ is not completely contained in C_i, then let C be a cluster of $\mathcal{C}_k^g$ such that $C \cap \{v\} \times V' \neq \emptyset$ and $C \neq C_i$. Since the greedy algorithm only merges connected clusters, we get that $C \subset \{v\} \times V'$. Then merging C_i and C increases the modularity more than the merge of C_i and C_j:

$$\begin{aligned} m \cdot \Delta\mathsf{q}(C_i, C) &\geq 1 - \frac{(n+d) \cdot d'}{2m} \\ m \cdot \Delta\mathsf{q}(C_i, C_j) &= 1 - \frac{(n+d) \cdot (n+\widetilde{d})}{2m} \quad , \end{aligned}$$

where d is the sum of degrees of nodes in C_i without v, d' is the sum of degrees of nodes in C, and $\widetilde{d}$ is the sum of degrees of nodes in C_j without those belonging to V. Since $d' \leq 2|E'| \leq n$ and $\widetilde{d} \geq 1$ the merge of C_i and C is performed before the merge of C_i and C_j.

Second, if $C_i = \{v\} \cup \{v\} \times V'$ and $C_i = \{w\} \cup \{w\} \times V'$, then the merge of the two clusters decreases the modularity:

$$\begin{aligned} m \cdot \Delta\mathsf{q}(C_i, C_j) &= 1 - \frac{(n + 2|E'| + 1)^2}{n^2 + (1 + 2|E'|)n} \\ &= 1 - \frac{n^2 + 4n|E'| + 2n + (2|E'| + 1)^2}{n^2 + 2|E'|n + n} \\ &= 1 - 1 - \frac{2n|E'| + n + (2|E'| + 1)^2}{n^2 + 2|E'|n + n} \quad . \end{aligned}$$

Since $|E'| \geq 1$, the change in modularity Δq is negative. Thus the merge will not be executed.

Theorem 8. *Let $n \geq 2$ be an integer and $H = (V', E')$ be a undirected and connected graph with at least two nodes. If $2|E'| + 1 < n$ then the greedy algorithm returns the clustering $\mathcal{C}^g := \{\{v\} \cup \{v\} \times V' \mid v \in V\}$ for $K_n \star_u H$ (for any fixed $u \in H$). This clustering has a modularity score of*

$$4m^2 \cdot \mathsf{q}(\mathcal{C}^g) = 4m\left((|E'| + 1) \cdot n\right) - n\left(2|E'| + 1 + n\right)^2 \ .$$

Proof. Since the greedy algorithm only merges two clusters if they are connected, Lemmas 10, 11 and 12 ensure that $\mathcal{C}^g = \mathcal{C}^g_k$ for some sufficient large k. According to Lemma 12, no further merge can occur. Thus, Theorem 8 is proven.

The next corollary reveals that the clustering, in which G and each copy of H form individual clusters, has a greater modularity score. We first observe an explicit expression for modularity.

Corollary 3. *The clustering $\mathcal{C}^s$ is defined as $\mathcal{C}^s := \{V\} \cup \{\{v\} \times V' \mid v \in V\}$ and, according to Equation (2), its modularity is*

$$4m^2 \cdot \mathsf{q}(\mathcal{C}^s) = 4m\left(|E'|n + \binom{n}{2}\right) - n\left(2|E'| + 1\right)^2 - \left(n \cdot (n - 1 + 1)\right)^2 \ .$$

If $n \geq 2$ and $2|E'| + 1 < n$, then clustering $\mathcal{C}^s$ has higher modularity than $\mathcal{C}^g$.

Theorem 9. *The approximation factor of the greedy algorithm for finding clusterings with maximum modularity is at least 2.*

Proof. We prove this statement by showing that the quotient $\mathsf{q}(\mathcal{C}^s) / \mathsf{q}(\mathcal{C}^g)$ is asymptotically 2 for a certain graph family. Therefore, we simplify the modularity scores with respect to their dominant terms. Note that $4m = 2n^2 + 4n|E'| + o(n^2)$. By Theorem 8 clustering $\mathcal{C}^g$ yields

$$4m\left((|E'| + 1) \cdot n\right) - n\left(2|E'| + 1 + n\right)^2 = 4m \cdot n \cdot |E'| + o\left(n^4\right) \tag{6}$$

and by Corollary 3 $\mathcal{C}^s$ yields

$$\begin{aligned} 4m\left(|E'|n + \binom{n}{2}\right) & - n\left(2|E'| + 1\right)^2 - \left(n \cdot (n - 1 + 1)\right)^2 \\ & = 4m \cdot n \cdot |E'| + 2mn^2 - n^4 + o\left(n^4\right) \\ & = 4m \cdot n \cdot |E'| + 2n^3|E'| + o\left(n^4\right) \end{aligned} \tag{7}$$

Thus, we obtain the following equation:

$$\mathcal{R}^g := \frac{\mathsf{q}(\mathcal{C}^s)}{\mathsf{q}(\mathcal{C}^g)} = 1 + \frac{2n^3|E'| + o(n^4)}{4mn|E'| + o(n^4)}$$

and for sufficiently large n we can omit the additional terms which are contained in $o(n^4)$:

$$\frac{2n^2}{2n^2 + 4n|E'| + o(n^2)} = \frac{1}{1 + \frac{2|E'|}{n} + o(1)}$$

By selecting paths with $1/2\sqrt{n}$ edges as graphs H, we obtain that $\mathcal{R}^g \geq 2 - \varepsilon$ for every positive ε.

The quotient $\mathsf{q}(\mathcal{C}^s) / \mathsf{q}(\mathcal{C}^g)$ asymptotically approaches 2 for n going to infinity on $K_n \star_u H$ with H a path of length $1/2\sqrt{n}$.

5 Optimality Results

5.1 Formulation as Integer Linear Program

The problem of maximizing modularity can be cast into a very simple and intuitive integer linear program (ILP). Given a graph $G = (V, E)$ with $n := |V|$ nodes, we define n^2 decision variables $X_{uv} \in \{0, 1\}$, one for every pair of nodes $u, v \in V$. The key idea is that these variables can be interpreted as an equivalence relation (over V) and thus form a clustering. In order to ensure consistency, we need the following constraints, which guarantee

$$\begin{aligned}
&\text{reflexivity } \forall\, u\colon X_{uu} = 1 \enspace , \\
&\text{symmetry } \forall\, u, v\colon X_{uv} = X_{vu} \enspace , \text{ and} \\
&\text{transitivity } \forall\, u, v, w\colon \begin{cases} X_{uv} + X_{vw} - 2 \cdot X_{uw} \leq 1 \\ X_{uw} + X_{uv} - 2 \cdot X_{vw} \leq 1 \\ X_{vw} + X_{uw} - 2 \cdot X_{uv} \leq 1 \end{cases} \quad .
\end{aligned}$$

The objective function of modularity then becomes

$$\frac{1}{2m} \sum_{(u,v) \in V^2} \left(E_{uv} - \frac{\deg(u)\deg(v)}{2m} \right) X_{uv} \enspace ,$$

$$\text{with } E_{uv} = \begin{cases} 1 & \text{, if } (u,v) \in E \\ 0 & \text{, otherwise} \end{cases} \quad .$$

Note that this ILP can be simplified by pruning redundant variables and constraints, leaving only $\binom{n}{2}$ variables and $\binom{n}{3}$ constraints.

5.2 Characterization of Cliques and Cycles

In this section, we provide several results on the structure of clusterings with maximum modularity for cliques and cycles. This extends previous work, in particular [2], in which in which cycles and cycles of cliques were used to reason about global properties of modularity.

A first observation is that modularity can be simplified for general d-regular graphs as follows.

Corollary 4. *Let $G = (V, E)$ be an unweighted d-regular graph and $\mathcal{C} = \{C_1, \dots, C_k\} \in \mathcal{A}(G)$. Then the following equality holds:*

$$q(\mathcal{C}) = \frac{|E(C)|}{dn/2} - \frac{1}{n^2} \sum_{i=1}^{k} |C_i|^2 \enspace . \tag{8}$$

The correctness of the corollary can be read off the definition given in Equation (2) and the fact that $|E| = d|V|/2$. Thus, for regular graphs modularity only depends on cluster sizes and coverage.

Cliques We first deal with the case of complete graphs. Corollary 5 provides a simplified formulation for modularity. From this rewriting, the clustering with maximum modularity can directly be obtained.

Corollary 5. *Let K_n be a complete graph on n nodes and $\mathcal{C} := \{C_1, \dots, C_k\} \in \mathcal{A}(K_n)$. Then the following equality holds:*

$$\mathsf{q}(\mathcal{C}) = -\frac{1}{n-1} + \frac{1}{n^2(n-1)} \sum_{i=1}^{k} |C_i|^2 \quad . \tag{9}$$

Proof. Coverage of $\mathcal{C}$ can be expressed in terms of cluster sizes as follows:

$$\begin{aligned} |E(\mathcal{C})| &= \binom{n}{2} - \sum_{i=1}^{k} \prod_{j>i} |C_i| \cdot |C_j| = \binom{n}{2} - \frac{1}{2} \sum_{i=1}^{k} \prod_{j \neq i} |C_i| \cdot |C_j| \\ &= \binom{n}{2} - \frac{1}{2} \sum_{i=1}^{k} |C_i| \cdot \sum_{j \neq i} |C_j| = \binom{n}{2} - \frac{1}{2} \sum_{i=1}^{k} |C_i| \cdot (n - |C_i|) \\ &= \binom{n}{2} - \frac{1}{2} \left(n^2 - \sum_{i=1}^{k} |C_i|^2 \right) = -\frac{n}{2} + \frac{1}{2} \sum_{i=1}^{k} |C_i|^2 \quad . \end{aligned}$$

Thus, we obtain

$$\begin{aligned} \mathsf{q}(\mathcal{C}) &= -\frac{1}{n-1} + \frac{1}{n(n-1)} \sum_{i=1}^{k} |C_i|^2 - \frac{1}{n^2} \sum_{i=1}^{k} |C_i|^2 \\ &= -\frac{1}{n-1} + \frac{1}{n^2 \cdot (n-1)} \sum_{i=1}^{k} |C_i|^2 \quad , \end{aligned}$$

which proves the equation.

Thus, maximizing modularity is equivalent to maximizing the squares of cluster sizes. Using the general inequality $(a+b)^2 \geq a^2 + b^2$ for non-negative real numbers, the clustering with maximum modularity is the 1–clustering. More precisely:

Theorem 10. *Let k and n be integers, K_{kn} be the complete graph on $k \cdot n$ nodes and $\mathcal{C}$ a clustering such that each cluster contains exactly n elements. Then the following equality holds:*

$$\mathsf{q}(\mathcal{C}) = \left(-1 + \frac{1}{k} \right) \cdot \frac{1}{kn - 1} \quad .$$

For fixed $k > 1$ and as n tends to infinity, modularity is always strictly negative, but tends to zero. Only for $k = 1$ modularity is zero and thus is the global maximum.

As Theorem 10 deals with one clique, the following corollary provides the optimal result for k disjoint cliques.

Corollary 6. *The maximum modularity of a graph consisting of k disjoint cliques of size n is $1 - 1/k$.*

The corollary follows from the definition of modularity in Equation (2). Corollary 6 gives a glimpse on how previous approaches have succeeded to upper bound modularity as it was pointed out in the context of Lemma 1.

Cycles Next, we focus on simple cycles, i. e., connected 2-regular graphs. According to Equation (8), modularity can be expressed as given in Equation (10), if each cluster is connected which may safely be assumed (see Corollary 2).

$$\mathsf{q}\left(\mathcal{C}\right) = \frac{n-k}{n} - \frac{1}{n^2}\sum_{i=1}^{k} |C_i|^2 \quad . \tag{10}$$

In the following, we prove that clusterings with maximum modularity are balanced with respect to the number and the sizes of clusters. First we characterize the distribution of cluster sizes for clusterings with maximum modularity, fixing the number k of clusters. For convenience, we minimize $F := 1 - \mathsf{q}\left(\mathcal{C}\right)$, where the argument of F is the distribution of the cluster sizes.

Proposition 1. *Let k and n be integers, the set $D^{(k)} := \left\{x \in \mathbb{N}^k \,\middle|\, \sum_{i=1}^{k} x_i = n\right\}$, and the function $F\colon D^{(k)} \to \mathbb{R}$ defined as*

$$F(x) := \frac{k}{n} + \frac{1}{n^2}\sum_{i=1}^{k} x_i^2 \qquad \text{for } x \in D^{(k)} \quad .$$

Then, F has a global minimum at x^* with $x_i^* = \left\lfloor \frac{n}{k} \right\rfloor$ for $i = 1, \ldots, k-r$ and $x_i^* = \left\lceil \frac{n}{k} \right\rceil$ for $i = k-r+1, \ldots, k$, where $0 \le r < k$ and $r \equiv n \mod k$.

Proof. Since k and n are given, minimizing F is equivalent to minimizing $\sum_i x_i^2$. Thus let us rewrite this term:

$$\begin{aligned} \sum_{i=1}^{k}\left(x_i - \frac{n}{k}\right)^2 &= \sum_{i=1}^{k} x_i^2 - 2\frac{n}{k}\sum_{i=1}^{k} x_i + k \cdot \left(\frac{n}{k}\right)^2 \\ &= \sum_{i=1}^{k} x_i^2 - 2\frac{n^2}{k} + \frac{n^2}{k} \\ \iff \qquad \sum_{i=1}^{k} x_i^2 &= \underbrace{\sum_{i=1}^{k}\left(x_i - \frac{n}{k}\right)^2}_{=:h(x)} + \frac{n^2}{k} \end{aligned}$$

Thus minimizing F is equivalent to minimizing h. If r is 0, then $h(x^*) = 0$. For every other vector y the function h is strictly positive, since at least one summand is positive. Thus x^* is a global optimum.

Let $r > 0$. First, we show that every vector $x \in D^{(k)}$ that is close to $(\frac{n}{k}, \ldots, \frac{n}{k})$ has (in principle) the form of x^*. Let $x \in D \cap [\left\lfloor \frac{n}{k} \right\rfloor, \left\lceil \frac{n}{k} \right\rceil]^k$, then it is easy to verify that there are $k-r$ entries that have value $\left\lfloor \frac{n}{k} \right\rfloor$ and the remaining r entries have value $\left\lceil \frac{n}{k} \right\rceil$. Any 'shift of one unit' between two variables having the same value, increases the corresponding cost: Let $\varepsilon := \left\lceil \frac{n}{k} \right\rceil - \frac{n}{k}$ and $x_i = x_j = \left\lceil \frac{n}{k} \right\rceil$. Replacing x_i with $\left\lfloor \frac{n}{k} \right\rfloor$ and x_j with $\left\lceil \frac{n}{k} \right\rceil + 1$, causes an increase of h by $5 + 2\varepsilon > 0$. Similarly, in the case of $x_i = x_j = \left\lfloor \frac{n}{k} \right\rfloor$ and the reassignment $x_i = \left\lceil \frac{n}{k} \right\rceil$ and $x_j = \left\lfloor \frac{n}{k} \right\rfloor - 1$, causes an increase of h by $2 > 0$.

Finally, we show that any vector of $D^{(k)}$ can be reach from x^* by 'shifting one unit' between variables. Let $x \in D^{(k)}$ and with loss of generality, we assume that $x_i \le x_{i+1}$ for all i. We define a sequence of elements in $D^{(k)}$ as follows:

1. $x^{(0)} := x^*$
2. if $x^{(i)} \neq x$, define $x^{(i+1)}$ as follows

$$x_j^{(i+1)} := \begin{cases} x_j^{(i)} - 1 & \text{, if } j = \min\left\{\ell \mid x_\ell^{(i)} > x_\ell\right\} =: L \\ x_j^{(i)} + 1 & \text{, if } j = \max\left\{\ell \mid x_\ell^{(i)} < x_\ell\right\} =: L' \\ x_j^{(i)} & \text{, otherwise} \end{cases}$$

Note that all obtained vectors $x^{(i)}$ are elements of $D^{(k)}$ and meet the condition of $x_j^{(i)} \leq x_{j+1}^{(i)}$. Furthermore, we gain the following formula for the cost:

$$\sum_j \left(x_j^{(i+1)}\right)^2 = \sum_j \left(x_j^{(i)}\right)^2 + 2\left(x_{L'}^{(i)} - x_L^{(i)} + 1\right) \quad .$$

Since $L < L'$, one obtains $x_{L'}^{(i)} \geq x_L^{(i)}$. Thus x^* is a global optimum in $D^{(k)}$.

Due to the special structure of simple cycles, we can swap neighboring clusters without changing the modularity. Thus, we can safely assume that clusters are sorted according to their sizes, starting with the smallest element. Then x^* is the only optimum. Evaluating F at x^* leads to a term that only depends on k and n. Hence, we can characterize the clusterings with maximum modularity only with respect to the number of clusters. The function to be minimized is given in Lemma 13:

Lemma 13. *Let C_n be a simple cycle with n nodes, $h\colon [1,\ldots,n] \to \mathbb{R}$ a function defined as*

$$h(x) := x \cdot n + n + \left\lfloor \frac{n}{x} \right\rfloor \left(2n - x \cdot \left(1 + \left\lfloor \frac{n}{x} \right\rfloor\right)\right) \quad ,$$

and k^* be the argument of the global minimum of h. Then every clustering of C_n with maximum modularity has k^* clusters.

Proof. Note, that $h(k) = F(x^*)$, where F is the function of Proposition 1 with the given k. Consider first the following equations:

$$\begin{aligned} \sum_{i=1}^k (x_i^*)^2 &= (k-r) \cdot \left\lfloor \frac{n}{k} \right\rfloor^2 + r \cdot \left\lceil \frac{n}{k} \right\rceil^2 \\ &= (k-r)\frac{(n-r)^2}{k^2} + r\left(\frac{(n-r)}{k} + 1\right)^2 \\ &= \frac{n-r}{k}\left((n-r) + 2r\right) + r = \frac{n^2 - r^2}{k} + r \\ &= \frac{1}{k}\left(n^2 - \left(n - \left\lfloor \frac{n}{k} \right\rfloor k\right)^2\right) + n - \left\lfloor \frac{n}{k} \right\rfloor k \\ &= 2n\left\lfloor \frac{n}{k} \right\rfloor - k\left\lfloor \frac{n}{k} \right\rfloor^2 + n - \left\lfloor \frac{n}{k} \right\rfloor k \\ &= n + \left\lfloor \frac{n}{k} \right\rfloor \left(2n - k\left(\left\lfloor \frac{n}{k} \right\rfloor + 1\right)\right) \end{aligned}$$

Since maximizing modularity is equivalent to minimize the expression $k/n + 1/n^2 \sum_i x_i^2$ for $(x_i) \in \bigcup_{j=1}^n D^{(j)}$. Note that every vector (x_i) can be realized as clustering with connected clusters. Since we have characterized the global minima for fixed k, it is sufficient to find the global minima by varying k.

Finally we obtain the characterization for clusterings with maximum modularity for simple cycles.

Theorem 11. *Let n be an integer and C_n a simple cycle with n nodes. Then every clustering $\mathcal{C}$ with maximum modularityhas k cluster of almost equal size, where*

$$k \in \left[\frac{n}{\sqrt{n+\sqrt{n}}} - 1, \frac{1}{2} + \sqrt{\frac{1}{4} + n}\right] \ .$$

Furthermore, there are only 3 *possible values for k for sufficiently large n.*

Proof. First, we show that the function h can be bounded by the inequalities given in (11) and is monotonically increasing (decreasing) for certain choices of k.

$$kn + \frac{n^2}{k} \leq h(k) \leq kn + \frac{n^2}{k} + \frac{k}{4} \ . \tag{11}$$

In order to verify the Inequalities (11), let ε_k be defined as $n/k - \lfloor n/k \rfloor$ (≥ 0). Then the definition of h can be rewritten as follows:

$$\begin{aligned} h(k) &= kn + n + \left\lfloor\frac{n}{k}\right\rfloor \left(2n - \left(1 + \left\lfloor\frac{n}{k}\right\rfloor\right) k\right) \\ &= kn + n + \left(\frac{n}{k} - \varepsilon_k\right)\left(2n - \left(1 + \frac{n}{k} - \varepsilon_k\right) k\right) \\ &= kn + n + \frac{2n^2}{k} - (1-\varepsilon_k)n - \frac{n^2}{k} - 2n\varepsilon_k + (1-\varepsilon_k)k\varepsilon_k + n\varepsilon_k \\ &= kn + \frac{n^2}{k} + (1-\varepsilon_k)\varepsilon_k k \ . \end{aligned}$$

Replacing the term $(1-\varepsilon_k)\varepsilon_k k$ by a lower (upper) bound of 0 ($k/4$) proves the given statements.

Second, the function h is monotonically increasing for $k \geq 1/2 + \sqrt{1/4+n}$ and monotonically decreasing for $k \leq n/\sqrt{n+\sqrt{n}} - 1$. In order to prove the first part, it is sufficient to show that $h(k) \leq h(k+1)$ for every suitable k.

$$\begin{aligned} h(k+1) - h(k) = &(k+1)n + n + \left\lfloor\frac{n}{k+1}\right\rfloor \left(2n - \left(1 + \left\lfloor\frac{n}{k+1}\right\rfloor\right)(k+1)\right) \\ &-kn - n - \left\lfloor\frac{n}{k}\right\rfloor \left(2n - \left(1 + \left\lfloor\frac{n}{k}\right\rfloor\right) k\right) \\ =& n + 2n\left(\left\lfloor\frac{n}{k+1}\right\rfloor - \left\lfloor\frac{n}{k}\right\rfloor\right) - \left(1 + \left\lfloor\frac{n}{k+1}\right\rfloor\right)\left\lfloor\frac{n}{k+1}\right\rfloor \\ &+k\left(\left(1 + \left\lfloor\frac{n}{k}\right\rfloor\right)\left\lfloor\frac{n}{k}\right\rfloor - \left(1 + \left\lfloor\frac{n}{k+1}\right\rfloor\right)\left\lfloor\frac{n}{k+1}\right\rfloor\right) \end{aligned}$$

Since $\lfloor\cdot\rfloor$ is discrete and $|\lfloor x \rfloor - \lfloor x-1 \rfloor| \leq 1$, one obtains:

$$h(k+1) - h(k) = \begin{cases} n - \left\lfloor\frac{n}{k}\right\rfloor^2 - \left\lfloor\frac{n}{k}\right\rfloor & \text{, if } \left\lfloor\frac{n}{k}\right\rfloor = \left\lfloor\frac{n}{k-1}\right\rfloor \\ 3n - \left\lfloor\frac{n}{k}\right\rfloor^2 - \left\lfloor\frac{n}{k}\right\rfloor + 2k\left\lfloor\frac{n}{k}\right\rfloor & \text{, otherwise} \end{cases} \tag{12}$$

Since $3n - \lfloor n/k \rfloor^2 - \lfloor n/k \rfloor + 2k \lfloor n/k \rfloor > n - \lfloor n/k \rfloor^2 - \lfloor n/k \rfloor$, it is sufficient to show that $n - \lfloor n/k \rfloor^2 - \lfloor n/k \rfloor \geq 0$. This inequality is fulfilled if $n - (n/k)^2 - n/k \geq 0$. Solving the quadratic equations leads to $k \geq 1/2 + \sqrt{1/4 + n}$.

Using the above bound, for the second part, it is sufficient to show that

$$kn + \frac{n^2}{k} - (k+1)n - \frac{n^2}{k+1} - \frac{k+1}{4} \geq 0 \ , \tag{13}$$

since this implies that the upper bound of $h(k+1)$ is smaller than (the lower bound of) $h(k)$. One can rewrite the left side of Inequality (13) as:

$$kn + \frac{n^2}{k} - (k+1)n - \frac{n^2}{k+1} - \frac{k+1}{4} = -n + \frac{n^2}{k(k+1)} - \frac{k+1}{4} \quad .$$

Since $h(k) - h(k+1)$ is monotonically decreasing for $0 \leq k \leq \sqrt{n}$, it is sufficient to show that $h(k) - h(k+1)$ is non-negative for the maximum value of k. We show that the lower bound $h_{-}(k) := -n + n^2/(k+1)^2 - (k+1)/4$ is non-negative.

$$\begin{aligned} h_{-}\left(\frac{n}{\sqrt{n+\sqrt{n}}} - 1\right) &= -n - \frac{n}{4\sqrt{n+\sqrt{n}}} + \frac{n^2(n+\sqrt{n})}{n^2} \\ &= \sqrt{n} - \underbrace{\frac{n}{4\sqrt{n+\sqrt{n}}}}_{\leq \frac{1}{4}\sqrt{n}} \geq 0 \end{aligned}$$

Summarizing, the number of clusters k (of an optimum clustering) can only be contained in the given interval, since outside the function h is either monotonically increasing or decreasing.The length of the interval is less than

$$\frac{1}{2} + \underbrace{\sqrt{\frac{1}{4} + n} - \frac{n}{\sqrt{n+\sqrt{n}}}}_{=:\ell(n)} + 1 \ \ .$$

The function $\ell(n)$ can be rewritten as follows:

$$\begin{aligned} \ell(n) &= \frac{\sqrt{\left(\frac{1}{4}+n\right)\left(\sqrt{n+\sqrt{n}}\right)} - n}{\sqrt{n+\sqrt{n}}} \\ &\leq \frac{\left(n + \frac{1+\varepsilon}{2}\sqrt{n}\right) - n}{\sqrt{n+\sqrt{n}}} \qquad (14) \\ &\leq \frac{1+\varepsilon}{2}\sqrt{\frac{n}{n+\sqrt{n}}} \ , \end{aligned}$$

for every positive ε. Inequality (14) is due to the fact that

$$\begin{aligned} \left(\frac{1}{4}+n\right)\left(\sqrt{n+\sqrt{n}}\right) &\leq n^2 + n\sqrt{n} + \frac{1}{4}\left(n+\sqrt{n}\right) \\ &\leq n^2 + 2\frac{1+\varepsilon}{2}n\sqrt{n} + \frac{(1+\varepsilon)^2}{4}n \\ &= \left(n + \frac{1+\varepsilon}{2}\sqrt{n}\right)^2 \ , \end{aligned}$$

for sufficiently large n.

5.3 Characterization of Special Trees

We show that computing the clustering with maximum modularity is possible in polynomial time for two special families of trees: trees with $O(\log n)$ internal nodes and caterpillar trees.

Trees with $\boldsymbol{O(\log n)}$ internal nodes. We safely assume that the clusters are connected subgraphs, and that there are no clusters consisting only of leaf nodes (see Section 2). This significantly reduces the search space for a clustering with maximum modularity. For each edge we specify, whether it is an inter-cluster edge or not. The clustering then results directly from the given properties. Adapting Equation (2) we obtain:

Corollary 7. Modularity on trees *is given by the function* $q_T : 2^E \to \mathbb{R}$ *as*

$$q_T(S) = \frac{n - |S|}{n-1} - \frac{1}{4(n-1)^2} \sum_{C \in G_S} \left(\sum_{v \in C} \deg(v) \right)^2 ,$$

where $C \in G_S$ is a component in the tree G after removing the edge set S.

If there are n_i internal nodes in G, we have at most $n_i - 1$ edges, for which we must make the decision of being an inter-cluster edge. This leaves $2^{n_i - 1}$ candidate clusterings for maximum modularity. For tree structures with $n_i \in O(\log n)$ the number of candidate clusterings reduces to polynomial in n. The argument also applies for forests.

Corollary 8. *For forests with $O(\log n)$ internal nodes there is a polynomial time algorithm to find the clustering with maximum modularity.*

As an interesting special case we consider the star.

Lemma 14. *For a star there is no clustering with positive modularity. The star is the only tree network with this property.*

Proof. In any clustering the center node can be located in only one cluster. If there is more than one cluster, the others must either consist of single leaf nodes or of more than one connected component. Hence, by Lemmas 2 and 3 we see that the clustering with maximum modularity consists of one cluster encompassing the complete star. This yields an optimum modularity of 0. This proves the property for the star.
Consider a tree T with more than one internal node. We consider a clustering $\mathcal{C}$ of two clusters. As T is connected, there must be at least two adjacent internal nodes u and v. The two clusters C_u and C_v consist of the two components in $T - (u, v)$ with $u \in C_u$ and $v \in C_v$. Suppose there are k edges internal edges between nodes of C_u. Then

$$\mathrm{q}(\mathcal{C}) = \frac{n-2}{n-1} - \frac{(2k+1)^2 + (2(n-k)-3)^2}{4(n-1)^2},$$

and we see that

$$\begin{aligned}\frac{k}{2}(n-1)^2 \cdot \mathrm{q}(\mathcal{C}) &= n-k-2-\frac{1}{4k}\\ &> n-k-3\\ &\geq 0\end{aligned}$$

For the last two inequalities we note that $k \geq 1$, because u is an internal node. Furthermore, $k \leq n-3$, because (u,v) is an edge connecting two clusters and there must be at least one edge between nodes of C_v.

Caterpillar Trees. A *caterpillar* consists of a path of $P = (V_p, E_p)$ of n_p nodes. For each node $v \in V_p$ there are t_v additional nodes of degree 1, which are adjacent only to v. Hence, each path node v has degree $\deg(v) = t_v + 1$ if it is one end of the path and $\deg(v) = t_v + 2$ if it is inside the path. In total the caterpillar has $n = n_p + \sum_{v \in V_p} t_v$ nodes and $n-1$ edges.

Theorem 12. *There is an algorithm to find the clustering of maximum modularity on caterpillars in time $O(n_p^4)$.*

Proof. Note that due to Lemma 2, only edges between path nodes must be considered as cluster borders. For each node $v \in V_p$ we construct a weight

$$w(v) = \begin{cases} 2t_v + 1 & \text{, if } v \text{ is an end node of the path} \\ 2(t_v+1) & \text{otherwise} \end{cases},$$

and extend this function to node sets as $w(C) = \sum_{v \in C \cap V_p} w(v)$. The modularity of a clustering represented by $S \subseteq E$ is then given as

$$\mathrm{q}_T(S) = 1 - \frac{k}{n-1} + \frac{1}{(n-1)^2} \sum_{C \in G_S} w(C)^2.$$

Hence, for the rest of this proof we will disregard outer star nodes and consider only the weighted star centres on the path P. We present an algorithm to find the optimum clustering for a given number of k clusters that runs in $O(n_p^2 k)$ time. This directly translates into a algorithm to find the optimum clustering in $O(n_p^4)$ time.
To minimise the modularity for a clustering with k clusters one needs to minimise the function $h(\mathcal{C}) = \sum_{C \in G_S} w(C)^2$. In the optimum case it is possible to divide the weight equally, and assign each cluster a weight of $\mu = w(V_p)/k$. Thus, $h(C) \geq k\mu^2$. Minimizing h is equivalent to minimizing the extension over the lower bound captured by the following function f:

$$\begin{aligned} f(S) &:= \left(\sum_{C \in G_S} w(C)^2 \right) - k\mu^2 = \sum_{C \in G_S} (w(C)^2 - \mu^2) \\ &= \sum_{C \in G_S} (\mu - (\mu - w(C))^2 - \mu^2 = \sum_{C \in G_S} (\mu - w(C))^2 - 2(\mu - w(C)) \\ &= \sum_{C \in G_S} (\mu - w(C))^2 \end{aligned}$$

The last equality follows, because $\sum_{C \in G_S} w(C) = w(V_p) = k\mu$. The function f measures the deviation between cluster weights and the optimum cluster weight in l_2-norm. Thus, our task reduces to find an equilibrated partition of a path that optimizes f. This problem has been considered before in the area of graph partitioning [20]. For any $k = 1, \dots, n_p - 1$ Algorithm 2 uses a dynamic programming approach and runs in $O(n_p^2 k)$ time. It outputs the set S of inter-cluster edges of the best clustering under all clusterings with exactly k clusters. The problem is reduced to solving a shortest path problem in an adjusted network G_c. Let the nodes of P be labeled from left to right as $v_1, \dots v_{n_p}$. Furthermore, number the clusters from left to right increasingly. For each edge $e_i = (v_i, v_{i+1})$, there is a node u_{ij} in G_c, if e_i can feasibly be the border between clusters C_j and C_{j+1}. For example, for e_1 there is only node u_{11}, for e_2 there are u_{21} and u_{22}, and for e_{n_p-1} there is only $u_{n_p-1,k-1}$. In addition there is a starting node u_{00} and an end node $u_{n_p,k}$. The algorithm creates directed edges between nodes $(u_{l,j-1}, u_{ij})$ if $l < i$. This edge indicates that there is a cluster $C_j = \{v_{l+1}, \dots, v_i\}$. The weight of this edge is the difference under l_2-norm between the cluster weight and μ. It is easy to observe that any path between u_{00} and $u_{n_p,k}$ corresponds to a set of edges specifying $k-1$ cluster borders. The value for f of this clustering, i.e. the L_2-norm distance between cluster weights and average weight μ, is correctly captured by the edge weights on the path. Hence, the shortest path represents a clustering, which yields the minimal value for f and thus maximum modularity. The most time consuming part is the construction of the network G_c. The three loops yield a complexity of $O(n_p^2 k)$. In the end we can use the algorithm to compute the best clustering with k clusters for any $k = 1, \dots, n_p$. The best of these clusterings is the desired clustering with maximum modularity. As $k \leq n_p$, the running time of $O(n_p^4)$ follows. This proves the theorem.

Algorithm 2: Finding the clustering of k clusters with maximum modularity

```
Input: A caterpillar tree $G$ and an integer $k$
Output: Set $S$ of inter-cluster edges
Initialize $G_c = (\{u_{00}\}, E_c)$
Set $\mu = \frac{w(V_p)}{k}$ for $i = 1, \dots, n_p$ do
    for $j = \max(1, i+1+k-n_p)$ to $\min(i, k-1)$ do
        add $u_{ij}$ to $V_c$
        for each $u_{l,j-1}$ with $1 \leq l < i$ do
            Add $e' = (u_{l,j-1}, u_{ij})$ to $E$
            Let $w(e') = (w(\{v_{l+1}, \dots, v_i\}) - \mu)^2$
Solve shortest path problem on $G_c$ between $u_{00}$ and $u_{n_p,k}$
return $S = \{e_i \in E_p \mid \exists u_{ij}$ on the shortest path$\}$
```

Our analysis reveals that by dropping leaf nodes and introducing suitable node weights based on degrees, the optimal clustering can be found with a dynamic programming algorithm [20]. In general, optimizing modularity on trees for a fixed number of clusters is a special case of the tree equipartition problem, in which partitions are measured with the l_2-norm. While this problem is NP-complete in general [21], the modularity case is special as node weights depend on degrees. Finally, note that for the special case of a simple path it is possible to adapt ideas of the proof of Theorem 11 to derive a similar characterization of the optimal clustering.

6 Examples Revisited

In the following, we discuss two selected networks that were, among others, frequently considered in related work.

The first instance is the karate club network of Zachary originally introduced in [22] and used for demonstration in [23]. The network models social interactions between members of a karate club. More precisely, friendship between the members is presented before the club split up due to an internal dispute. A representation of the network is given in Figure 5. The partition that has resulted from the split is given by the shape of the nodes, while the colors indicate the clustering calculated by the greedy algorithm and blocks refer to a optimum clustering maximizing modularity, that has been obtained by solving the above ILP. The corresponding scores of modularity are 0.431 for the optimum clustering, 0.397 for the greedy clustering, and 0.383 for the clustering given by the split. Even though this is another example in which the greedy algorithm does not perform optimally, its score is comparatively good. Furthermore, the example shows one of the potential pitfalls the greedy algorithm can encounter: Due to the attempt to balance the squared sum of degrees (over the clusters), a node with large degree (white square) and one with small degree (white circle) are merged at an early stage. However, using the same argument, such a cluster will unlikely be merged with another one. Thus, small clusters with skewed degree distributions occur.

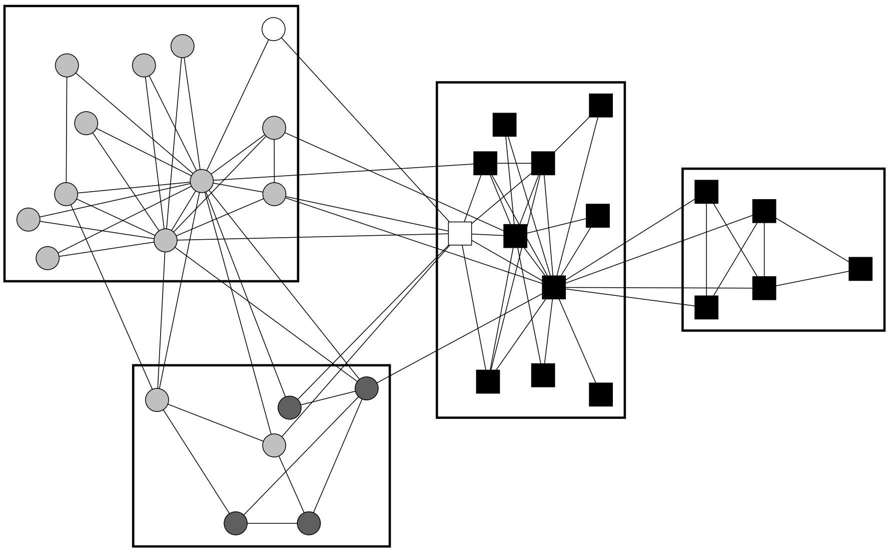


Fig. 5. Karate club network of Zachary [22]. The different clusterings are coded as follows: blocks represent the optimum clustering (with respect to modularity), colors correspond to the greedy clustering, and shapes code the split that occurred in reality.

The second instance is a network of books on politics, compiled by V. Krebs and used for demonstration in [9]. The nodes represent books on American politics bought from `Amazon.com` and edges join pairs of books that are frequently purchased together. A representation of the network is given in Figure 6. The optimum clustering maximizing modularity is give by the shapes of nodes, the colors of nodes indicate a clustering calculated by the greedy algorithm and the blocks show a clustering calculated by Geometric MST Clustering (GMC) which is introduced in [24] using the geometric mean of coverage and performance, both of which are quality indices discussed in the same paper. The corresponding scores of modularity are 0.527 for the optimum clustering, 0.502 for the greedy clustering, and 0.510 for the GMC clustering. Similar to the first example, the greedy algorithm is suboptimal, but relatively close to the optimum. Interestingly, GMC outperforms the greedy algorithm although it does not consider modularity in its calculations. This illustrates the fact that there probably are many *intuitive* clusterings close to the optimum clustering that all have relatively similar values of modularity. In analogy to the first example, we observe the same merge-artifact, namely the two nodes represented as dark-grey triangles.

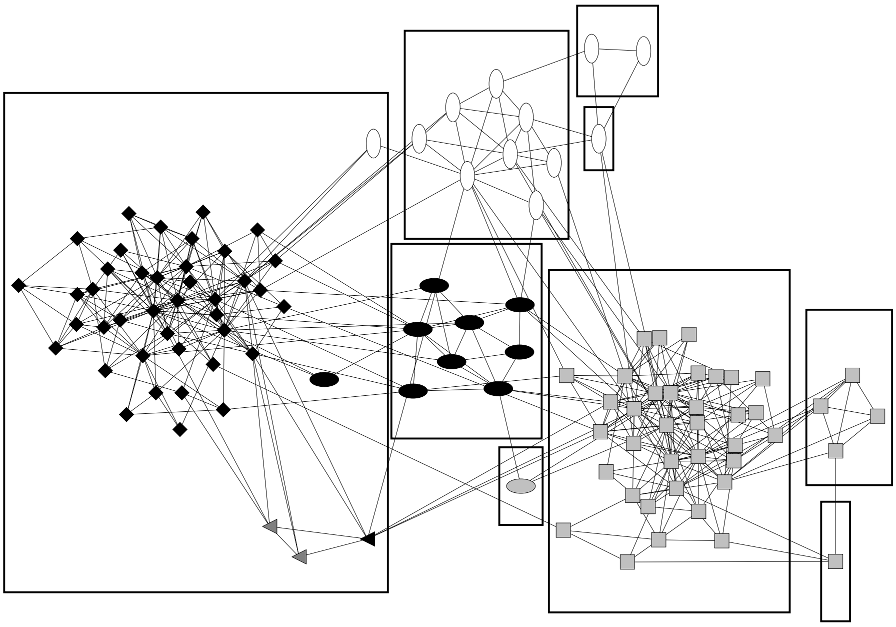


Fig. 6. The networks of books on politics compiled by V. Krebs. The different clusterings are coded as follows: blocks represent the clustering calculated with GMC, colors correspond to the greedy clustering, and shapes code the optimum clustering (with respect to modularity).

Summarizing, the two examples illustrated several interesting facts. First of all, an artifical pattern in the optimization process of the greedy algorithm is revealed: The early merge of two nodes, one with a high and one with a low degree, results in a cluster which will not be merged with another one later on. In general, this can prevent finding the optimum clustering. Nevertheless, it performs relatively well on the given instances and is at most 10% off the optimum. However, applying other algorithms that do not optimize modularity, we observe that the obtained clusterings have similar scores. Thus, achieving good scores of modularity does not seem to be too hard on these instances. On the one hand, these clusterings roughly agree in terms of the overall structure, on the other hand, they differ in numbers of clusters and even feature artifacts such as small clusters of size one or two. Considering that both examples exhibit significant community structure, we thus predict that there are many intuitive clusterings being structurally close (with respect to lattice structure) and that most suitable clustering algorithms probably identify one of them.

7 Conclusion

This paper represents the first approach to characterize the popular clustering index modularity with respect to optimality results and computational hardness. We have settled the open question about the complexity status of modularity maximization by proving its $\mathcal{NP}$-completeness in the strong sense. On the one hand, this justifies the use of approximation algorithms and heuristics, such as the widespread greedy approach. For the latter we prove a first lower bound on the approximation factor. Currently we are investigating the impact of scaling in order to improve this bound. On the other hand, by characterizing the structure of a clustering with maximum modularity, we established optimality results for certain graph families. Our analysis of the greedy algorithm also includes a brief comparison with the optimum clustering which is calculated via ILP on several real-world instances. For the future we plan an extended analysis and the development of a clustering algorithm with provable performance guarantees. The special properties of the measure, its popularity in application domains and the absence of fundamental theoretical insights hitherto, render further mathematically rigorous treatment of modularity necessary.

References

1. Newman, M.E.J., Girvan, M.: Finding and evaluating community structure in networks. Physical Review E **69** (2004)
2. Fortunato, S., Barthelemy, M.: Resolution Limit in Community Detection. In: Proceedings of the National Academy of Sciences. (2007) 36–41
3. Ziv, E., Middendorf, M., Wiggins, C.: Information-Theoretic Approach to Network Modularity. Physical Review E **71** (2005)
4. Muff, S., Rao, F., Caflisch, A.: Local Modularity Measure for Network Clusterizations. Physical Review E **72** (2005)
5. Fine, P., Paolo, E.D., Philippides, A.: Spatially Constrained Networks and the Evolution of Modular Control Systems. In: 9th Intl. Conference on the Simulation of Adaptive Behavior (SAB). (2006)
6. Gaertler, M., Görke, R., Wagner, D.: Significance-Driven Graph Clustering. In: Proceedings of the 3rd International Conference on Algorithmic Aspects in Information and Management (AAIM'07). Lecture Notes in Computer Science, Springer-Verlag (2007) to appear; accepted for publication.
7. Newman, M.E.J.: Fast Algorithm for Detecting Community Structure in Networks. Physical Review E **69** (2004)
8. Clauset, A., Newman, M.E.J., Moore, C.: Finding community structure in very large networks. Physical Review E **70** (2004)
9. Newman, M.: Modularity and Community Structure in Networks. In: Proceedings of the National Academy of Sciences. (2005) 8577–8582
10. White, S., Smyth, P.: A Spectral Clustering Approach to Finding Communities in Graph. In: SIAM Data Mining Conference. (2005)
11. Guimerà, R., Sales-Pardo, M., Amaral, L.A.N.: Modularity from Fluctuations in Random Graphs and Complex Networks. Physical Review E **70** (2004)
12. Reichardt, J., Bornholdt, S.: Statistical Mechanics of Community Detection. Physical Review E **74** (2006)
13. Duch, J., Arenas, A.: Community Detection in Complex Networks using Extremal Optimization. Physical Review E **72** (2005)
14. Gaertler, M.: Clustering. In Brandes, U., Erlebach, T., eds.: Network Analysis: Methodological Foundations. Volume 3418 of Lecture Notes in Computer Science. Springer-Verlag (2005) 178–215
15. Danon, L., Díaz-Guilera, A., Duch, J., Arenas, A.: Comparing community structure identification. Journal of Statistical Mechanics (2005)

16. Garey, M.R., Johnson, D.S.: Computers and Intractability. A Guide to the Theory of $\mathcal{NP}$-Completeness. W. H. Freeman and Company (1979)
17. Newman, M.: Analysis of Weighted Networks. Technical report, Cornell University, Santa Fe Institute, University of Michigan (2004)
18. Giotis, I., Guruswami, V.: Correlation Clustering with a Fixed Number of Clusters. In: Proceedings of the 17th Annual ACM–SIAM Symposium on Discrete Algorithms (SODA'06), New York, NY, USA (2006) 1167–1176
19. Bui, T., Chaudhuri, S., Leighton, F., Sipser, M.: Graph bisection algorithms with good average case behavior. Combinatorica **7** (1987) 171–191
20. Simeone, B.: Optimal connected partitions of graphs. DIMACS Tutorial (1999) http://rutcor.rutgers.edu/∼boros/LSDO/BrunoSimeone.html.
21. Schröder, M.: Gebiete optimal aufteilen. PhD thesis, School of Economics and Business Engineering, Universität Karlsruhe (2001)
22. Zachary, W.W.: An Information Flow Model for Conflict and Fission in Small Groups. Journal of Anthropological Research **33** (1977) 452–473
23. Newman, M.E.J., Girvan, M.: Mixing Patterns and Community Structure in Networks. In Pastor-Satorras, R., Rubi, M., Diaz-Guilera, A., eds.: Statistical Mechanics of Complex Networks. Volume 625 of Lecture Notes in Physics. Springer-Verlag (2003) 66–87
24. Brandes, U., Gaertler, M., Wagner, D.: Experiments on Graph Clustering Algorithms. In: Proceedings of the 11th Annual European Symposium on Algorithms (ESA'03). Volume 2832 of Lecture Notes in Computer Science. (2003) 568–579