On Finding Graph Clusterings with Maximum Modularity*

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,
{Ulrik.Brandes,Martin.Hoefer}@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 study the problem of finding clusterings with maximum modularity, thus providing theoretical foundations for past and present work based on this measure. More precisely, we prove the conjectured hardness of maximizing modularity both in the general case and with the restriction to cuts, and give an Integer Linear Programming formulation. This is complemented by first insights into the behavior and performance of the commonly applied greedy agglomaration approach.

1 Introduction

Graph clustering is a fundamental graph-theoretic problem in data and, more specifically, network analysis [1]. Studied for decades and applied in many settings, it is currently popular as the problem of partitioning networks into communities. In this line of research, a novel graph clustering index called *modularity* has been proposed recently [2]. The rapidly growing interest in this measure prompted a series of follow-up studies on various applications and possible adjustments (see, e.g., [3,4,5,6,7]). Moreover, an array of heuristic algorithms has been proposed to optimize modularity. These are based on a greedy agglomeration [8,9], on spectral division [10,11], simulated annealing [12,13], or extremal optimization [14] to name but a few prominent examples. While these studies often provide subjective plausibility arguments in favor of the resulting partitions, we know of only one attempt to characterize properties of clusterings with maximum modularity [3]. In particular, none of the proposed algorithms has been shown to produce partitions that are optimal with respect to modularity.

In this paper we study the problem of finding clusterings with maximum modularity, thus providing theoretical foundations for past and present work based on this measure. More precisely, we proof the conjectured hardness of maximizing modularity both in the general case and the restriction to cuts, and give an integer linear programming formulation to facilitate optimization without enumeration of all clusterings. Since the most commonly employed heuristic to optimize modularity is based on greedy agglomeration, we investigate its worst-case behavior. 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 greedy clusterings may heavily depend on the tie-breaking strategy utilized. In fact, in the worst case, no approximation factor can be provided. These performance studies are concluded by partitioning some previously considered networks optimally, which does yield further insight.

This paper is organized as follows. Section 2 contains brief preliminaries, formulations of modularity and an ILP formulation of the problem. Basic and counterintuitive properties of modularity are observed in Sect. 3. Our $\mathcal{NP}$-completeness proofs are given in Sect. 4, followed by an analysis of the greedy approach in Sect. 5. Our work is concluded by characterizations of revisited examples from previous work in Sect. 6 and a brief discussion in Sect. 7.

* 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 the EU under grant DELIS (contract no. 001907).

2 Preliminaries

Throughout this paper, we will use the notation of [15]. 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 [2] and define the *modularity* $\mathsf{q}(\mathcal{C})$ of a clustering $\mathcal{C}$ as

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

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. We prefer to rewrite Equation (1) into the more informative

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

which 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. Note that the first term $|E(\mathcal{C})|/m$ is also known as *coverage* [15].

2.2 Maximizing Modularity via Integer Linear Programming

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

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

The objective function of modularity then becomes

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

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

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

3 Fundamental Observations

In the following, we identify basic structural properties that clusterings with maximum modularity fulfill. We first 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 $-1/2 \leq q(\mathcal{C}) \leq 1$ holds.*

Lemma 1 is proven by minimizing modularity, for details see [16]. 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 upper bound is obvious from our reformulation in Equation (2), and has been observed previously [3,4,17]. It can only be attained in the specific case of a graph with no edges, where coverage is commonly defined to be 1. The following four results strongly characterize the rough 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.*

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

The proofs of Lemmas 2 and 3 can be found in [16] and are straightforward. Both rely on the fact that a strict increase in modularity is possible, if they are violated.

Corollary 2. *A clustering of maximum modularity 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.

3.1 Counterintuitive Behavior

In the last section, we listed some intuitive and desirable properties like connectivity within clusters for clusterings of maximum modularity. However, due to the enforced balance between coverage and the sums of squared cluster degrees, counter-intuitive situations arise. These are non-locality, scaling behavior, and sensitivity to satellites.

Non-Locality. At a first view, 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 colors. 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 rightmost black node from the black cluster to the white cluster, although locally its neighborhood structure has not changed.

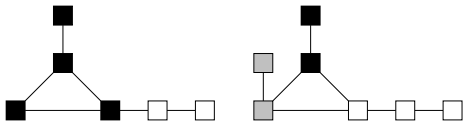
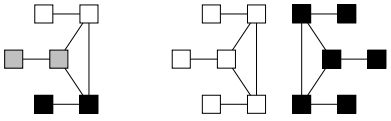


Fig. 1. Non-local behavior. Clusters are represented by colors.

Sensitivity to Satellites. 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, 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 2(a) shows such an example.

Scaling Behavior. Figures 2(a) and 2(b) display the scaling behavior of modularity. By simply doubling the graph presented in Figure 2(a), the optimal clustering is altered completely. While in Figure 2(a) we obtain three clusters each consisting of the minor K_2, the clustering with maximum modularity of the graph in Figure 2(b) consists of two clusters, each being a graph equal to the one in Figure 2(a).



(a) (b)

Fig. 2. Scaling behavior. Clusters are represented by colors.

This behavior is in line with the previous observations in [3,5], 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 Equation 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 [3,5].

4 $\mathcal{NP}$-Completeness

It has been conjectured that maximizing modularity is hard [9], but no formal proof was provided to date. We next show that decision version of modularity maximization is indeed $\mathcal{NP}$-complete.

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

Note that we may ignore the fact that, in principle, K could be a real number in the range $[0,1]$, because $4m^2 \cdot \mathrm{q}(\mathcal{C})$ 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 [18], 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.

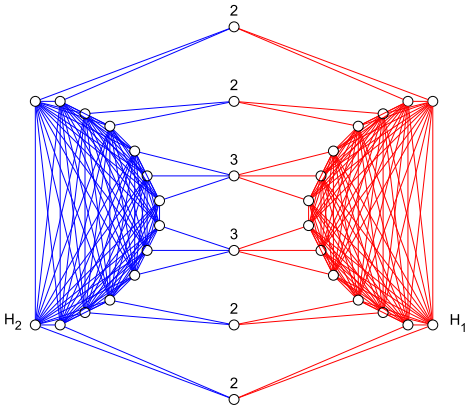


Fig. 3. An example graph $G(A)$ for the instance $A = \{2,2,2,2,3,3\}$ of 3-PARTITION. Node labels indicate the corresponding numbers $a_i \in A$.

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 $a = \sum_{i=1}^{3k} a_i$ 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 3 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 $H_1,\ldots,H_k$ is split.*

The proof of Lemma 4 can be found in [16]. It is based on the fact that modularity can be increased by a modification of the clustering, if Lemma 4 is violated. 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 $H_1,\ldots,H_k$.*

The proof of Lemma 5 follows the same intuition as that of Lemma 4. It can also be found in [16].

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.*

Closely following the proofs of the previous two lemmas, we obtain the proof of Lemma 6 in [16].

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 each cluster element nodes corresponding to elements that sum to $b = 1/k\cdot a$. In this case the sum 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 [19]. 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.

4.1 Special Case: Modularity with a Bounded Number of Clusters

A common clustering approach is based on iteratively identifying cuts with respect to some quality measures, see for example [20,21,22]. The general problem being $\mathcal{NP}$-complete, we now complete our hardness results by proving that the restricted optimization problem is hard as well. More precisely, we consider the two problems of computing the clustering with maximum 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 (2-MODULARITY) *Given a graph G and a number K, is there a clustering $\mathcal{C}$ of G into exactly/at most 2 clusters, for which $\mathsf{q}(\mathcal{C}) \geq K$?*

Our proof uses a reduction similar to the one for showing the hardness of the "MinDisAgree[2]" problem of correlation clustering [23]. The reduction is from MINIMUM BISECTION FOR CUBIC GRAPHS (MB3).

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 [24]. 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 $\mathcal{C}^*$ 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.*

The proof of Lemma 7 can be found in [16]. Next, we show that in an optimum clustering, all the nodes of one node clique $cliq(v)$ are located in one cluster. The proof is also published in [16]

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

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. The proof can be found in [16].

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

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

Theorem 6. *2-MODULARITY is strongly $\mathcal{NP}$-complete.*

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

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

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.

5 The Greedy Algorithm

In contrast to the abovementioned iterative cutting strategy, another commonly used approach to find clusterings with good quality scores is based on greedy agglomeration [15,25]. In the case of modularity, this approach is particularly widespread [8,9]. It 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 $\Delta_{i,j} := \mathsf{q}(\mathcal{C}_{i,j}) - \mathsf{q}(\mathcal{C})$, 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 $\Delta_{i,j}$ is the maximum, in these cases the algorithm selects an arbitrary pair. An efficient implementation using appropriate data structures requires $O\left(n^2 \log n\right)$ runtime. Note that $n-1$ is an upper bound on the number of iterations and that one can terminate the algorithm as soon as the matrix Δ contains only non-positive entries. This is due to a property called *single-peakedness*, proven in [9].

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 (Theorem 9). In order to prove this statement, we introduce a general construction scheme given in Definition 1. While the former result relies on a deterministic procedure of the algorithm, in the following we even 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 (Theorem 7).

Modularity is defined such that it takes values in the interval $[-1/2, 1]$ for any graph and any clustering (Lemma 1). In particular the modularity of a trivial clustering placing all vertices into a single cluster has a value of 0. We exploit 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 potentially 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 4 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:

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

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 $\mathcal{C}$

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

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 4. 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 4(a). $\mathcal{C}$ consists of one cluster for the vertices of each edge of E_m and has a modularity value of

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

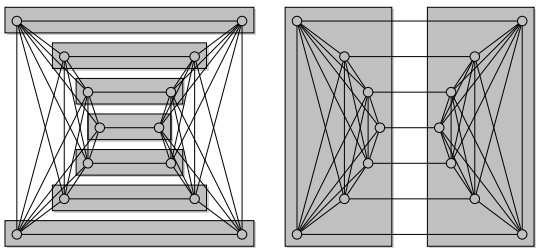
Due to the single-peakedness of the problem [9] all following cluster merges can never increase this value, hence the algorithm will return a clustering of value 0.

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 4(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.



(a) Clustering with modularity 0 (b) Clustering with modularity close to $1/2$

Fig. 4. Worst-case agglomeration vs. optimum.

Next, we provide a weaker lower bound for a different class of graphs, but making no assumptions on 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*

$$E''_c := \{\{v,(v,u)\} \mid v \in V\} \qquad \text{and}$$
$$E''_H := \left\{\left\{(v,v'),(v,w')\right\} \mid v \in V, v',w' \in V'', \left\{v',w'\right\} \in E\right\} \quad .$$

An example is given in Figure 5. 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.

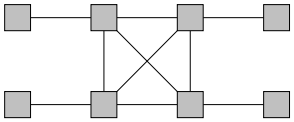


Fig. 5. The graph $K_4 \star_u P_1$.

Theorem 8. *Let $n \geq 2$ be an integer and $H=(V',E')$ be an 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*

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

The proof of Theorem 8, which relies on the graph construction described above, builds upon three lemmas which can be found in [16]. 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*

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

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

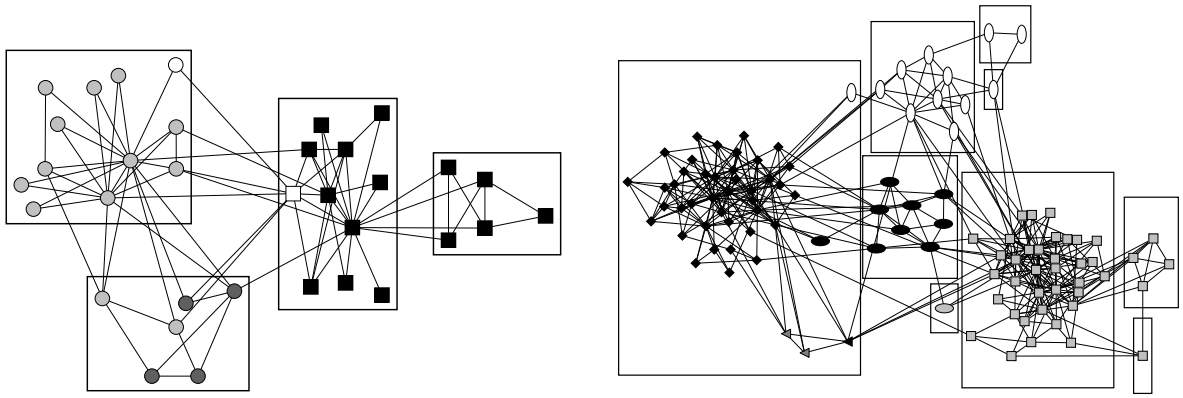
Theorem 9. *The approximation factor of the greedy algorithm for finding clusterings with maximum modularity is no better than 2.*

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 $1/2\sqrt{n}$. The full proof of Theorem 9 can also be found in [16].

6 Examples Revisited

Applying our results about maximizing modularity gained so far, we revisit two example networks that were used in related work [26,27,10]. More precisely, we compare published greedy solutions with respective optima, thus revealing two peculiarities of modularity. First, we illustrate a behavioral pattern of the greedy merge strategy and, second, we relativize the quality of the greedy approach.

The first instance is the karate club network of Zachary originally introduced in [26] and used for demonstration in [27]. 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 6(a). 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



(a) Karate club network of Zachary [26]. 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.

(b) 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).

Fig. 6. Examples

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. Observe the following peculiarity: 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 relatively soon. Using the same argument, such a cluster will unlikely be merged with another one. As a result, a cluster rarely has only one node, but relative small clusters still occur, featuring skew distribution of node degrees.

The second instance is a network of books on politics, compiled by V. Krebs and used for demonstration in [10]. 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(b). The optimum clustering maximizing modularity is given 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 [28] 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. A key observation is that GMC outperforms the greedy algorithm although it does not consider modularity in its calculations. Moreover, the comparison of the structure of the calculated clusterings reveals that several clusterings close to the optimum one still have relative large modularity score. Thus, the good performance of the greedy approach comes as no surprise.

7 Conclusion

We provide the first formal assessments of a popular clustering index known as modularity. We have settled the open question about the complexity status of modularity maximization by proving its $\mathcal{NP}$-completeness in the strong sense. We show that this even holds for the restricted version with a bound of two on the number of clusters. This justifies the further investigation of approximation algorithms and heuristics, such as the widespread greedy approach. For the latter we prove a first lower bound on the approximation factor. Our analysis of the greedy algorithm also includes a brief comparison with the optimum clustering which is calculated via ILP on some real-world instances, thus encouraging a reconsideration of previous results. 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. Brandes, U., Erlebach, T., eds.: Network Analysis: Methodological Foundations. Volume 3418 of Lecture Notes in Computer Science. Springer-Verlag (2005)
2. Newman, M.E.J., Girvan, M.: Finding and evaluating community structure in networks. Physical Review E **69** (2004)
3. Fortunato, S., Barthelemy, M.: Resolution Limit in Community Detection. Proceedings of the National Academy of Sciences **104** (2007) 36–41
4. Ziv, E., Middendorf, M., Wiggins, C.: Information-Theoretic Approach to Network Modularity. Physical Review E **71** (2005)
5. Muff, S., Rao, F., Caflisch, A.: Local Modularity Measure for Network Clusterizations. Physical Review E **72** (2005)
6. 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)
7. 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) 11–26
8. Newman, M.E.J.: Fast Algorithm for Detecting Community Structure in Networks. Physical Review E **69** (2004)
9. Clauset, A., Newman, M.E.J., Moore, C.: Finding community structure in very large networks. Physical Review E **70** (2004)
10. Newman, M.: Modularity and Community Structure in Networks. In: Proceedings of the National Academy of Sciences. (2005) 8577–8582
11. White, S., Smyth, P.: A Spectral Clustering Approach to Finding Communities in Graph. In: SIAM Data Mining Conference. (2005)
12. Guimerà, R., Sales-Pardo, M., Amaral, L.A.N.: Modularity from Fluctuations in Random Graphs and Complex Networks. Physical Review E **70** (2004)
13. Reichardt, J., Bornholdt, S.: Statistical Mechanics of Community Detection. Physical Review E **74** (2006)
14. Duch, J., Arenas, A.: Community Detection in Complex Networks using Extremal Optimization. Physical Review E **72** (2005)
15. Gaertler, M.: Clustering. [1] 178–215
16. Brandes, U., Delling, D., Gaertler, M., Görke, R., Hoefer, M., Nikoloski, Z., Wagner, D.: On Modularity - NP-Completeness and Beyond. Technical Report 2006-19, ITI Wagner, Faculty of Informatics, Universität Karlsruhe (TH) (2006)
17. Danon, L., Díaz-Guilera, A., Duch, J., Arenas, A.: Comparing community structure identification. Journal of Statistical Mechanics (2005)
18. Garey, M.R., Johnson, D.S.: Computers and Intractability. A Guide to the Theory of $\mathcal{NP}$-Completeness. W. H. Freeman and Company (1979)
19. Newman, M.: Analysis of Weighted Networks. Technical report, Cornell University, Santa Fe Institute, University of Michigan (2004)
20. Alpert, C.J., Kahng, A.B.: Recent Directions in Netlist Partitioning: A Survey. Integration: The VLSI Journal **19** (1995) 1–81
21. Hartuv, E., Shamir, R.: A Clustering Algorithm based on Graph Connectivity. Information Processing Letters **76** (2000) 175–181
22. Vempala, S., Kannan, R., Vetta, A.: On Clusterings - Good, Bad and Spectral. In: Proceedings of the 41st Annual IEEE Symposium on Foundations of Computer Science (FOCS'00). (2000) 367–378
23. 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
24. Bui, T.N., Chaudhuri, S., Leighton, F.T., Sipser, M.: Graph bisection algorithms with good average case behavior. Combinatorica **7** (1987) 171–191
25. Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: a review. ACM Computing Surveys **31** (1999) 264–323
26. Zachary, W.W.: An Information Flow Model for Conflict and Fission in Small Groups. Journal of Anthropological Research **33** (1977) 452–473
27. 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
28. 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