The Parameterised Complexity of Computing the Maximum Modularity of a Graph

Kitty Meeks[1] (ORCID) · **Fiona Skerman**[2]

Abstract

The *maximum modularity* of a graph is a parameter widely used to describe the level of clustering or community structure in a network. Determining the maximum modularity of a graph is known to be NP-complete in general, and in practice a range of heuristics are used to construct partitions of the vertex-set which give lower bounds on the maximum modularity but without any guarantee on how close these bounds are to the true maximum. In this paper we investigate the parameterised complexity of determining the maximum modularity with respect to various standard structural parameterisations of the input graph G. We show that the problem belongs to FPT when parameterised by the size of a minimum vertex cover for G, and is solvable in polynomial time whenever the treewidth or max leaf number of G is bounded by some fixed constant; we also obtain an FPT algorithm, parameterised by treewidth, to compute any constant-factor approximation to the maximum modularity. On the other hand we show that the problem is W[1]-hard (and hence unlikely to admit an FPT algorithm) when parameterised simultaneously by pathwidth and the size of a minimum feedback vertex set.

✉ Kitty Meeks
  kitty.meeks@glasgow.ac.uk

1 School of Computing Science, University of Glasgow, Glasgow, UK

2 Department of Mathematics, Heilbronn Institute for Mathematical Research, University of Bristol, Bristol, UK

1 Introduction

The increasing availability of large network datasets has led to great interest in techniques to discover network structure. An important and frequently observed structure in networks is the existence of groups of vertices with many connections between them, often referred to as 'communities'.

Newman and Girvan introduced the modularity function in 2004 [24]. Modularity gives a measure of how well a graph can be partitioned into communities and is used in the most popular algorithms to cluster large networks. For example, the Louvain method, an iterative clustering technique, uses the modularity function to choose which parts from the previous step to fuse into larger parts at each step [2,16]. The widespread use of modularity and empirical success in finding communities makes modularity an important function to study from an algorithmic point of view.

In this paper we are concerned with the computational complexity of computing the maximum modularity of a given input graph, and specifically in the following decision problem.

MODULARITY
Input: A graph G and a constant $q \in [0, 1]$.
Question: Is the maximum modularity of G at least q?

This problem was shown to be NP-complete in general by Brandes et al. [4], using a construction that relies on the fact that all vertices of a sufficiently large clique must be assigned to the same part of an optimal partition. They also showed that a variation of the problem in which we wish to find the optimal partition into exactly two sets is hard; their proof for this relied again on the use of large cliques, but DasGupta and Desai [6] later showed that this 2-clustering problem remains NP-complete on d-regular graphs for any fixed $d \geq 9$. It has also been shown that it is NP-hard to approximate the maximum modularity within any constant factor [8], although there is a polynomial-time constant-factor approximation algorithm for certain families of scale-free networks [10]. The hardness of computing constant-factor multiplicative approximations in general has motivated research into approximation algorithms with an additive error [8,17]: the best known result is an approximation algorithm with additive error roughly 0.42084 [17].

In this paper we initiate the study of the parameterised complexity of MODULARITY, considering its complexity with respect to several standard structural parameterisations. On the positive side, we show that the problem is in FPT when parameterised by the cardinality of a minimum vertex cover for the input graph G, and that it belongs to XP when parameterised by either the treewidth or max leaf number of G. The XP algorithm parameterised by treewidth can easily be adapted to give an FPT algorithm, parameterised by treewidth, to compute any constant-factor approximation maximum modularity. On the other hand, we demonstrate that MODULARITY, parameterised by treewidth, is unlikely to belong to FPT: we prove that the problem is W[1]-hard even when parameterised simultaneously by the pathwidth of G and the size of a mini-

mum feedback vertex set for G. For background on parameterised complexity, and the complexity classes discussed here, we refer the reader to [5,11].

These results follow the same pattern as those obtained for the problem EQUITABLE CONNECTED PARTITION [12], and indeed our hardness result involves a reduction from a specialisation of this problem. There are clear similarities between the two problems: in a partition that maximises the modularity, every part will induce a connected subgraph and, in certain circumstances, we achieve the maximum modularity with a partition into parts that are as equal as possible. However, the crucial difference between the two problems is that the input to EQUITABLE CONNECTED PARTITION includes the required number of parts, whereas MODULARITY requires us to maximise over all possible partition sizes; in fact, if we restrict to partitions with a specified parts, it is no longer necessarily true that a partition maximising the modularity must induce connected subgraphs. This difference makes reductions between the two problems non-trivial.

1.1 The Modularity Function

The definition of modularity was first introduced by Newman and Girvan in [24]. Many or indeed most popular algorithms used to search for clusterings on large datasets are based on finding partitions with high modularity [15,19], and the heuristics within them sometimes also use local modularity optimisation, for example in the Louvain method [2]. See [14,25] for surveys on community detection including modularity based methods.

Knowledge on the maximum modularity for classes of graphs helps to understand the behaviour of the modularity function. There is a growing literature on this which began with cycles and complete graphs in [4]. Bagrow [1] and Montgolfier et al. [7] showed some classes of trees have high maximum modularity which was extended in [21] to all trees with maximum degree $o(n)$, and furthermore to all graphs where the product of treewidth and maximum degree grows more slowly than the number of edges. Many random graph models also have high modularity, see [22,23] for a treatment of Erdős-Renyi random graphs, [21] for random regular graphs and also [26] which includes the preferential attachment model.

Given a set A of vertices, let $e(A)$ denote the number of edges within A, and let $\mathrm{vol}(A)$ (sometimes called the volume of A) denote the sum of the degree d_v (in the whole graph G) over the vertices v in A. For a graph G with $m \geq 1$ edges and a vertex partition $\mathcal{A}$ of G, set the modularity score of $\mathcal{A}$ on G to be

$$q_{\mathcal{A}}(G) = \frac{1}{2m} \sum_{A \in \mathcal{A}} \sum_{u,v \in A} \left(\mathbf{1}_{uv \in E} - \frac{d_u d_v}{2m} \right) = \frac{1}{m} \sum_{A \in \mathcal{A}} e(A) - \frac{1}{4m^2} \sum_{A \in \mathcal{A}} \mathrm{vol}(A)^2;$$

the maximum modularity of G is $q^*(G) = \max_{\mathcal{A}}(G)$, where the maximum is over all partitions $\mathcal{A}$ of the vertices of G. Graphs with no edges are defined conventionally to have modularity 1. However note that if the modularity of graphs with no edges were defined to be 0 it would not change any of the results.

The modularity function is designed to score partitions highly when most edges fall within the parts and penalise partitions with very few or very big parts. These two objectives are encoded as the *edge contribution* or *coverage* $q_{\mathcal{A}}^{E}(G) = \frac{1}{m}\sum_{A \in \mathcal{A}} e(A)$, and *degree tax* $q_{\mathcal{A}}^{D}(G) = \frac{1}{4m^2}\sum_{A \in \mathcal{A}} \text{vol}(A)^2$, in the modularity of a vertex partition $\mathcal{A}$ of G.

Note that for any graph with $m \geq 1$ edges $0 \leq q^*(G) \leq 1$. To see the lower bound, notice that the trivial partition which places all vertices in the same part has modularity zero. For example, complete graphs and stars have modularity 0 as noted in [4]. A graph consisting of c disjoint cliques of the same size has modularity $1 - 1/c$ with the optimal partition taking each clique to be a part.

As modularity is at most 1 it is sometimes useful to consider the *modularity deficit* $\tilde{q}_{\mathcal{A}}(G) = 1 - q_{\mathcal{A}}(G)$. Denote by $\partial(A)$ the number of edges between vertex set A and the rest of the graph. Then

$$\tilde{q}_{\mathcal{A}}(G) = \frac{1}{2m}\sum_{A \in \mathcal{A}}\left(\partial(A) + \frac{\text{vol}(A)^2}{2m}\right)$$

and we may equivalently minimise the modularity deficit to maximise the modularity. In particular

$$\tilde{q}(G) = \min_{A \in \mathcal{A}}\tilde{q}_{\mathcal{A}}(G) = 1 - q^*(G).$$

We will make use of several facts about the maximum modularity of a graph.

Fact 1 (Lemma 1 of [9], Lemma 2.1 of [6]) *For any integer $c > 0$ and any graph G,*

$$\max_{|\mathcal{A}| \leq c} q_{\mathcal{A}}(G) > q^*(G)\left(1 - \frac{1}{c}\right).$$

Fact 2 (Lemma 3.4 of [4]) *Suppose that G is a graph that contains no isolated vertices. If $\mathcal{A}$ is a partition of $V(G)$ such that $q_{\mathcal{A}}(G) = q^*(G)$ then, for every $A \in \mathcal{A}$, $G[A]$ is a connected subgraph of G.*

Fact 3 (Corollary 1 of [4]) *Let $G = (V, E)$ and suppose that $V_0 \subseteq V$ is a set of isolated vertices. Then $q(G) = q(G \backslash V_0)$. Moreover, if partitions $\mathcal{A}$ and $\mathcal{A}'$ agree on all vertices of $V \backslash V_0$, then $q_{\mathcal{A}}(G) = q_{\mathcal{A}'}(G)$.*

Fact 4 (Lemma 1.6.5 of [27]) *If $\mathcal{A}$ is a partition of $V(G)$ such that $q_{\mathcal{A}}(G) = q^*(G)$ then no part A consists of a single non-isolated vertex.*

Proof Let u be a vertex with degree $d_u > 0$ and suppose (for a contradiction) that $\mathcal{A} = \{\{u\}, A_1, \ldots, A_k\}$ is an optimal partition of G. For each $i = 1, \ldots, k$ define the vertex partition $\mathcal{B}_i = \{A_1, \ldots, A_i \cup \{u\}, \ldots, A_k\}$. We can derive a simple expression for $q_{\mathcal{B}_i}(G) - q_{\mathcal{A}}(G)$ as most terms cancel:

$$q_{\mathcal{B}_i}(G) - q_{\mathcal{A}}(G) = \frac{1}{m}e(\{u\}, A_i) - \frac{1}{2m^2}d_u\,\text{vol}(A_i).$$

By assumption, $\mathcal{A}$ is an optimal partition so $q_{\mathcal{B}_i}(G) \leq q_{\mathcal{A}}(G)$ and thus for each i we have $2m \cdot e(\{u\}, A_i) \leq d_u \operatorname{vol}(A_i)$. Hence we can sum over $i = 1, \ldots, k$ and the inequality should hold. However for the LHS $2m \sum_i e(\{u\}, A_i) = 2md_u$ and the RHS is

$$d_u \sum_{i=1}^{k} \operatorname{vol}(A_i) = d_u(2m - d_u) < 2md_u$$

and so we have our contradiction.   □

Observe that Facts 2, 3 and 4 together imply that the search for an optimal partition can be restricted to those in which all parts are connected subgraphs and no part consists of a single node.

1.2 Notation and Definitions

Given a graph $G = (V, E)$, and a set $U \subseteq V$ of vertices, we write $G[U]$ for the subgraph of G induced by U and $G \backslash U$ for $G[V \backslash U]$. Given two disjoint subsets of vertices $A, B \subseteq V$, we write $e(A, B)$ for the number of edges with one endpoint in A and the other in B. We shall often want to denote the number of edges between a set of vertices and the remainder of the graph so set $\partial(A) = e(A, \bar{A})$. If $\mathcal{P}$ is a partition of a set X, and $Y \subset X$, we write $\mathcal{P}[Y]$ for the restriction of $\mathcal{P}$ to Y.

A *vertex cover* of a graph $G = (V, E)$ is a set $U \subseteq V$ such that every edge has at least one endpoint in U; equivalently, $G \backslash U$ is an independent set (i.e. contains no edges). The *vertex cover number* of G is the smallest cardinality of any vertex cover of G. A *feedback vertex set* for G is a set $U \subseteq V$ such that $G \backslash U$ contains no cycles. Notice that the vertex cover number of G gives an upper bound on the size of the smallest feedback vertex set for G, written $\operatorname{fvs}(G)$. The *max leaf number* of G is the maximum number of leaves (degree one vertices) in any spanning tree of G.

A *tree decomposition* of a graph G is a pair $(T, \mathcal{D})$ where T is a tree and $\mathcal{D} = \{\mathcal{D}(t) : t \in V(T)\}$ is a collection of non-empty subsets of $V(G)$ (or *bags*), indexed by the nodes of T, satisfying:

1. $V(G) = \bigcup_{t \in V(T)} \mathcal{D}(t)$,
2. for every $e = uv \in E(G)$, there exists $t \in V(T)$ such that $u, v \in \mathcal{D}(t)$,
3. for every $v \in V(G)$, if $T(v)$ is defined to be the subgraph of T induced by nodes t with $v \in \mathcal{D}(t)$, then $T(v)$ is connected.

We will assume throughout that the indexing tree T has a distinguished root node r; if not we may choose an arbitrary node to be the root. Given any node $t \in V(T)$ we write V_t for the set of vertices of G that appear in bags indexed by t and the descendants of t.

If T is in fact a path, we say that $(T, \mathcal{D})$ is a *path decomposition* of G. The *width* of the tree decomposition $(T, \mathcal{D})$ is defined to be $\max_{t \in V(T)} |\mathcal{D}(t)| - 1$, and the *treewidth* of G, written $\operatorname{tw}(G)$, is the minimum width over all tree decompositions of G. The *pathwidth* of G, $\operatorname{pw}(G)$, is the minimum width over all path decompositions of G.

We note that there is an FPT algorithm to compute a minimum-width tree decom-
position of any graph G, where the treewidth of G is taken as the parameter [3].
Moreover, any such tree decomposition can be transformed into a so-called *nice* tree
decomposition (having certain algorithmically useful properties) in linear time, with-
out increasing the number of nodes by more than a constant factor [18].

2 Positive Results

In this section we identify a number of structural restrictions on the input graph that
allow us to compute the maximum modularity of a graph, or a good approximation to
this quantity, efficiently.

2.1 Parameterisation by Vertex Cover Number

In this section we demonstrate that MODULARITY is in FPT when parameterised by the
vertex cover number of the input graph.

Theorem 5 MODULARITY, *parameterised by cardinality of a minimum vertex cover
for the input graph G, is in* FPT.

To prove this result, we make use of recent work of Lokshtanov [20] which gives
an FPT algorithm for the following problem.

INTEGER QUADRATIC PROGRAMMING
Input: An $n \times n$ integer matrix Q, an $m \times n$ integer matrix A, and an m-dimensional
vector $\mathbf{b}$.
Parameter: $n + \alpha$, where α is the maximum absolute value of any entry in A or
Q.
Problem: Find a vector $\mathbf{x} \in \mathbb{Z}^n$ which minimises $\mathbf{x}^T Q \mathbf{x}$, subject to $A\mathbf{x} \le \mathbf{b}$.

Our strategy can be summarised as follows. We first observe that we may restrict
our attention to partitions in which every part intersects the vertex cover. Moreover,
the vertices outside the vertex cover can be classified into at most 2^k "types" according
to their neighbourhood (which by definition must be a subset of the vertex cover). We
then argue that the modularity of a partition depends only on (1) the inherited partition
of the vertex cover and (2) the number of (non-vertex-cover) vertices of each type that
belong to each of the parts. Using this characterisation, we can reduce the problem
of maximising the modularity to that of solving a collection of instances of INTEGER
QUADRATIC PROGRAMMING.

Before embarking on the proof of Theorem 5, we introduce some notation. Suppose
that the graph $G = (V, E)$ has $|E| = m$, and that $U = \{u_1, \ldots, u_k\}$ is a vertex cover
for G. Let $\mathcal{P} = \{P_1, \ldots, P_\ell\}$ be a partition of U, and set $W = V \setminus U$ (so W is an
independent set).

We can partition the vertices of W into 2^k sets based on their *type*: the type $\tau_U(w) \in
\{0, 1\}^k$ of a vertex $w \in W$ describes which of the vertices in U are neighbours of
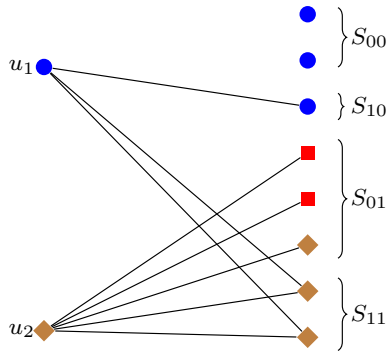
Fig. 1 An example of a graph with vertex cover $U = \{u_1, u_2\}$ and four sets of distinct types indicated for the vertices $W = V \setminus U$. For the vertex partition $\mathcal{A} = \{A_1, A_2, A_3\}$ indicated with circles (●), squares (■) and diamonds (◆) respectively the only non-zero values of $x_{\sigma,i}^{\mathcal{A}}$ are: $x_{00,1}^{\mathcal{A}} = 2$, $x_{10,1}^{\mathcal{A}} = 1$, $x_{01,2}^{\mathcal{A}} = 2$, $x_{01,3}^{\mathcal{A}} = 1$ and $x_{11,3}^{\mathcal{A}} = 2$. Note also that $\mathcal{A}$ extends the partition $\mathcal{P} = \{\{u_1\}, \{u_2\}\}$ of U but not the partition $\mathcal{P}' = \{\{u_1, u_2\}\}$ of U

w. Formally $\tau_U(w)_j = 1$ if $u_j w \in E(G)$ and $\tau_U(w)_j = 0$ otherwise. For each $\sigma \in \{0, 1\}^k$, we set S_σ to be the set of all vertices in W with type exactly σ, that is, $S_\sigma = \{w \in W : \tau(w) = \sigma\}$.

Now let $\mathcal{A} = \{A_1, \ldots, A_r\}$ be a partition of V. We write $x_{\sigma,i}^{\mathcal{A}}$ for the number of vertices of type σ which are assigned to A_i, that is, $x_{\sigma,i}^{\mathcal{A}} = |S_\sigma \cap A_i|$. Finally, we introduce 0-1 vectors to encode the sets $P_i \in \mathcal{P}$: for $1 \leq i \leq \ell$, we let $\pi^i \in \{0, 1\}^k$ be given by $\pi_j^i = 1$ if $u_j \in P_i$, and $\pi_j^i = 0$ otherwise. An example is given in Fig. 1.

We now argue that, if the partition $\mathcal{A}$ extends $\mathcal{P}$, we can compute the modularity of $\mathcal{A}$ using only the values $x_{\sigma,i}^{\mathcal{A}}$, together with information about $\mathcal{P}$.

Lemma 1 *Let $U = \{u_1, \ldots, u_k\}$ be a vertex cover for $G = (V, E)$, where $|E| = m$, and let $\mathcal{P}$ be a partition of U. If $\mathcal{A}$ is any partition of V which extends $\mathcal{P}$ and has the property that every $A \in \mathcal{A}$ has non-empty intersection with U, then*

$$q_{\mathcal{A}}^E(G) = \frac{1}{m} \sum_{i=1}^{\ell} e(P_i) + \frac{1}{m} \sum_{(\sigma,i)} x_{\sigma,i}^{\mathcal{A}} (\sigma \cdot \pi^i),$$

and

$$4m^2 q_{\mathcal{A}}^D = 4 \sum_i e(P_i)^2 + 4 \sum_{(\sigma,i)} x_{\sigma,i}^{\mathcal{A}} e(P_i)(\sigma \cdot (\mathbf{1} + \pi^i))$$

$$+ \sum_{(\sigma,i)(\sigma',j)} x_{\sigma,i}^{\mathcal{A}} x_{\sigma',j}^{\mathcal{A}} (\sigma \cdot (\mathbf{1} + \pi^i))(\sigma' \cdot (\mathbf{1} + \pi^j)).$$

Proof Suppose that $\mathcal{P} = \{P_1, \ldots, P_\ell\}$ and $\mathcal{A} = \{A_1, \ldots, A_\ell\}$, where $P_i \subseteq A_i$ for each i; we set $B_i = A_i \cap W$ for each $1 \leq i \leq \ell$. For any vertex $w \cap B_i$, we have that $e(w, P_i)$ is given by the dot product $\tau(w) \cdot \pi^i$; thus the number of edges between P_i and B_i for each i is given by

$$e(P_i, B_i) = \sum_{\sigma \in \{0,1\}^k} x_{\sigma,i}^{\mathcal{A}}(\sigma \cdot \pi^i). \qquad (1)$$

Since there are no edges inside any set B_i, it follows that

$$e(A_i) = e(P_i) + \sum_{\sigma \in \{0,1\}^k} x_{\sigma,i}^{\mathcal{A}}(\sigma \cdot \pi^i),$$

and hence we can write the edge contribution of $\mathcal{A}$ as

$$q_{\mathcal{A}}^{E}(G) = \frac{1}{m} \sum_{i=1}^{\ell} e(P_i) + \frac{1}{m} \sum_{(\sigma,i)} x_{\sigma,i}^{\mathcal{A}}(\sigma \cdot \pi^i). \qquad (2)$$

Similarly for the degree tax, observe that a vertex $w \in W$ of type $\tau(w)$ has degree $\tau(w) \cdot \mathbf{1} \leq k$, and hence $\mathrm{vol}(B_i) = \sum_{\sigma} x_{\sigma,i}(\sigma \cdot \mathbf{1})$. Notice that $\mathrm{vol}(P_i) = 2e(P_i) + e(P_i, B_i)$ and we already have an expression for $e(P_i, B_i)$ in terms of the $x_{\sigma,i}^{\mathcal{A}}$ in (1). Hence, as $\mathrm{vol}(P_i \cup B_i) = \mathrm{vol}(P_i) + \mathrm{vol}(B_i)$, we have

$$4m^2 q_{\mathcal{A}}^{D} = \sum_{i} \mathrm{vol}(P_i \cup B_i)^2 = \sum_{i} \left(2e(P_i) + \sum_{\sigma} x_{\sigma,i}^{\mathcal{A}}((\sigma \cdot \pi^i) + (\sigma \cdot \mathbf{1})) \right)^2$$

and thus rearranging,

$$4m^2 q_{\mathcal{A}}^{D} = 4 \sum_{i} e(P_i)^2 + 4 \sum_{i} e(P_i) \sum_{\sigma} x_{\sigma,i}^{\mathcal{A}}(\sigma \cdot (\mathbf{1} + \pi^i))$$

$$+ \left(\sum_{\sigma} x_{\sigma,i}^{\mathcal{A}}(\sigma \cdot (\mathbf{1} + \pi^i)) \right)^2$$

$$= 4 \sum_{i} e(P_i)^2 + 4 \sum_{(\sigma,i)} x_{\sigma,i}^{\mathcal{A}} e(P_i)(\sigma \cdot (\mathbf{1} + \pi^i))$$

$$+ \sum_{(\sigma,i)(\sigma',j)} x_{\sigma,i}^{\mathcal{A}} x_{\sigma',j}^{\mathcal{A}}(\sigma \cdot (\mathbf{1} + \pi^i))(\sigma' \cdot (\mathbf{1} + \pi^j)),$$

as required. $\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad$ □

We are now ready to prove the main result of this section.

Proof of Theorem 5 We will assume that the input to our instance of MODULARITY is a graph $G = (V, E)$, where $|E| = m$. We may assume without loss of generality that we are also given as input a vertex cover $U = \{u_1, \ldots, u_k\}$ for G (as if not we can easily compute one in the allowed time). We may further assume that G does not contain any isolated vertices, as we can delete any such vertices (in polynomial time) without changing the value of the maximum modularity (by Fact 3).

Note that the total number of possible partitions of U into non-empty parts is equal to the kth *Bell number*, B_k (and hence is certainly less than k^k). It therefore suffices to describe an fpt-algorithm which determines, given some partition $\mathcal{P}$ of U,

$$q^{\mathcal{P}}(G) = \max\{q_{\mathcal{A}}(G) : \mathcal{A}[U] = \mathcal{P}\}.$$

The maximum modularity of G can then be calculated by taking

$$\max\{q^{\mathcal{P}}(G) : \mathcal{P} \text{ is a partition of } U\}.$$

From now on, we consider a fixed partition $\mathcal{P} = \{P_1, \ldots, P_\ell\}$ of U, and describe how to compute $q^{\mathcal{P}}(G)$. It follows from Facts 2 and 4, together with the fact that W is an independent set that, if $\mathcal{A} = \{A_1, \ldots, A_j\}$ is a partition of V which achieves the maximum modularity, then every part A_i has non-empty intersection with U. We will call a partition with this property a U-*partition* of G. It then suffices to maximise the modularity over all U-partitions in order to determine the value of $q^{\mathcal{P}}(G)$.

Now, by Lemma 1, we know that we can express the modularity of a U-partition $\mathcal{A}$ as

$$q_{\mathcal{A}}(G) = \frac{1}{m} \sum_{i=1}^{\ell} e(P_i) + \frac{1}{m} \sum_{(\sigma,i)} x_{\sigma,i}^{\mathcal{A}} (\sigma \cdot \pi^i) - \frac{1}{m^2} \sum_i e(P_i)^2$$

$$- \frac{1}{m^2} \sum_{(\sigma,i)} x_{\sigma,i}^{\mathcal{A}} e(P_i) (\sigma \cdot (\mathbf{1} + \pi^i))$$

$$- \frac{1}{4m^2} \sum_{(\sigma,i)(\sigma',j)} x_{\sigma,i}^{\mathcal{A}} x_{\sigma',j}^{\mathcal{A}} (\sigma \cdot (\mathbf{1} + \pi^i))(\sigma' \cdot (\mathbf{1} + \pi^j)). \qquad (3)$$

As we have fixed the partition $\mathcal{P}$, all values $e(P_i)$ can be regarded as fixed constants. In order to determine the maximum modularity we can obtain with a U-partition, we therefore need to find the values of $x_{\sigma,i}^{\mathcal{A}}$ which maximise this expression.

We can rewrite (3) as the sum of a constant term, two linear functions θ and ϕ of the $x_{\sigma,i}^{\mathcal{A}}$ and a quadratic function ψ of the $x_{\sigma,i}^{\mathcal{A}}$ (up to scaling by constants):

$$q_{\mathcal{A}}(G) = \underbrace{\frac{1}{m} \sum_{i=1}^{\ell} e(P_i) - \frac{1}{m^2} \sum_i e(P_i)^2}_{\text{constant}}$$

$$+ \underbrace{\frac{1}{m} \sum_{(\sigma,i)} x_{\sigma,i}^{\mathcal{A}} (\sigma \cdot \pi^i)}_{\theta(\mathcal{A})} \underbrace{- \frac{1}{m^2} \sum_{(\sigma,i)} x_{\sigma,i}^{\mathcal{A}} e(P_i) (\sigma \cdot (\mathbf{1} + \pi^i))}_{\phi(\mathcal{A})}$$

$$\underbrace{- \frac{1}{4m^2} \sum_{(i,\sigma)(j,\sigma')} x_{\sigma,i}^{\mathcal{A}} x_{\sigma',j}^{\mathcal{A}} (\sigma \cdot (\mathbf{1} + \pi^i))(\sigma' \cdot (\mathbf{1} + \pi^j))}_{\psi(\mathcal{A})}.$$

To find the maximum value of $q_{\mathcal{A}}(G)$ over all U-partitions it therefore suffices to determine, for all possible values of $\theta(\mathcal{A})$ and $\phi(\mathcal{A})$, the minimum possible value of $\psi(\mathcal{A})$. Before describing how to do this, we observe that the number of combinations of possible values for $\theta(\mathcal{A})$ and $\phi(\mathcal{A})$ and is not too large. Note that $0 \leq \sum_{\sigma,i} x_{\sigma,i}^{\mathcal{A}}$ $(\sigma \cdot \pi^i) < nk$, and $0 \leq \sum_{\sigma,i} x_{\sigma,i}^{\mathcal{A}} e(P_i)(\sigma \cdot (1+\pi^i)) < n\binom{k}{2}2k < nk^3$, so the number of possible pairs $(\theta(\mathcal{A}), \phi(\mathcal{A}))$ is at most n^2k^4. Thus, if we know the minimum possible value of $\psi(\mathcal{A})$ corresponding to each possible pair $(\theta(\mathcal{A}), \phi(\mathcal{A}))$, we can compute the maximum modularity achieved by any U-partition $\mathcal{A}$ such that $(\theta(\mathcal{A}), \phi(\mathcal{A})) = (y, z)$, and maximising over the polynomial number of possible pairs (y, z) will give $q^{\mathcal{P}}(G)$.

Now, given a possible pair of values (y, z) for $(\theta(\mathcal{A}), \phi(\mathcal{A}))$, we describe how to compute

$$\min\{\psi(\mathcal{A}) : \mathcal{A} \text{ is a } U\text{-partition with } \theta(\mathcal{A}) = y \text{ and } \phi(\mathcal{A}) = z\}.$$

Our strategy is to express this minimisation problem as an instance of INTEGER QUADRATIC PROGRAMMING and then apply the FPT algorithm of [20].

In this instance, we have $n = \ell 2^k \leq k2^k$, and our vector of variables $\mathbf{x} = (x_1, \ldots, x_n)^T$ is given by

$$x_i = x_{(\sigma_{i \mod 2^k}), \lceil i/2^k \rceil}^{\mathcal{A}},$$

where $\sigma_1, \ldots, \sigma_{2^k}$ is a fixed enumeration of all vectors in $\{0, 1\}^k$. The matrix Q expresses the value of $\psi(\mathcal{A})$ in terms of $\mathbf{x}$: if we set $Q = \{q_{i,j}\}$ where

$$q_{i,j} = \left(\sigma_{(i \mod 2^k)} \cdot \left(1 + \pi^{\lceil i/2^k \rceil}\right)\right)\left(\sigma_{(j \mod 2^k)} \cdot \left(1 + \pi^{\lceil j/2^k \rceil}\right)\right),$$

then it is easy to see that $\psi(\mathcal{A}) = \mathbf{x}^T Q \mathbf{x}$. Note also that the maximum absolute value of any entry in Q is at most $4k^2$.

We now use the linear constraints to express the conditions that

1. $\theta(\mathcal{A}) = y$,
2. $\phi(\mathcal{A}) = z$, and
3. the values $x_{i,\sigma}$ correspond to a valid U-partition $\mathcal{A}$.

The first of these conditions can be expressed as a single linear constraint:

$$\sum_{(\sigma,i)} x_{\sigma,i}^{\mathcal{A}}(\sigma \cdot \pi^i) = y,$$

or equivalently $\mathbf{a}_1 \mathbf{x} = y$ where $\mathbf{a}_1$ is the $1 \times n$ row vector with i^{th} entry equal to

$$\sigma_{(i \mod 2^k)} \cdot \pi^{\lceil i/2^k \rceil}.$$

We can similarly express the second condition as a single linear constraint:

$$\sum_{(\sigma,i)} x_{\sigma,i}^{\mathcal{A}} \, e(P_i)(\sigma \cdot (\mathbf{1} + \pi^i)) = z,$$

or equivalently $\mathbf{a}_2 \mathbf{x} = z$, where $\mathbf{a}_2$ is the $1 \times n$ row vector with i^{th} entry equal to

$$e\left(P_{\lceil i/2^k \rceil}\right)\left(\sigma_{(i \mod 2^k)} \cdot \left(\mathbf{1} + \pi^{\lceil i/2^k \rceil}\right)\right).$$

Note that every entry in the vectors $\mathbf{a}_1$ and $\mathbf{a}_2$ has absolute value no more than $2k^3$. For the third condition, note that the values $x_{i,\sigma}$ correspond to a valid U-partition if and only if every $x_{i,\sigma}$ is non-negative, and for each σ we have $\sum_{i=1}^{\ell} x_{i,\sigma}^{\mathcal{A}} = |S_\sigma|$.

We can therefore express all three conditions in the form $A\mathbf{x} = \mathbf{b}$, where A is a $\left(4 + (\ell + 1)2^k\right) \times n$ and $\mathbf{b}$ is a $\left(4 + (\ell + 1)2^k\right)$-dimensional vector (notice that we use two inequalities to express each of the linear equality constraints).

Altogether, this means that the solution to this INTEGER QUADRATIC PROGRAM-MING instance will determine the values of $x_{i,\sigma}^{\mathcal{A}}$ which minimize (out of all values corresponding to some U-partition $\mathcal{A}$) the value of $\psi(\mathcal{A})$, subject to the additional requirement that $\theta(\mathcal{A}) = y$ and $\phi(\mathcal{A}) = z$. Note that the number of variables n is at most $k2^k$ and the largest absolute value of any entry in A or Q is at most $2k^3$, so the parameter in the instance of INTEGER QUADRATIC PROGRAMMING is bounded by a function of k. This completes the proof. □

We note the algorithm described can easily be modified to output an optimal partition.

2.2 Parameterisation by Treewidth

In this section we demonstrate that MODULARITY, when parameterised by the treewidth of the input graph G, belongs to XP and so is solvable in polynomial time on graph classes whose treewidth is bounded by some fixed constant. We further show that for any fixed $\varepsilon > 0$ there is an FPT-algorithm, parameterised by treewidth, which computes a factor $(1 - \varepsilon)$-approximation; i.e. returning a value between $(1 - \varepsilon)q^*$ and q^* where q^* is the maximum modularity of the graph.

Theorem 6 MODULARITY *parameterised by the treewidth of the input graph G is in* XP.

Proof As the proof makes use of standard dynamic programming techniques on tree decompositions, we only give an outline proof here. Suppose that G has n vertices and m edges, and has treewidth k. We will assume that we are given a nice tree decomposition $(T, \mathcal{D})$ (where T is a tree and $\mathcal{D} = \{\mathcal{D}(t) : t \in V(T)\}$) of G, of width k, as part of the input (if not we can compute one in FPT time).

The proof relies heavily on Fact 2. This means we can compute the optimum modularity without considering partitions that induce disconnected subgraphs; hence, for any node $t \in V(T)$, we need only consider partitions $\mathcal{A}$ with the property that, if

$A \in \mathcal{A}$ does not intersect $\mathcal{D}(t)$, then all vertices in A only appear in bags indexed by nodes in precisely one connected component of $T \setminus t$.

We compute the modularity by working upwards from the leaves in the standard way. As we do this, we need to keep track of relevant statistics for the parts that intersect the current bag (*liquid* parts) and also the total contribution to the modularity from the parts (*frozen* parts) which contain only vertices from bags indexed by descendants of the current node (and so by the reasoning above cannot accept more vertices from elsewhere in the graph).

For any node $t \in V(T)$, a valid *state* of t consists of the following:

1. a partition $\mathcal{P}$ of $\mathcal{D}(t)$;
2. a function $\alpha : \mathcal{P} \to [m]$ such that $\alpha(P_i) \geq e(P_i)$ for each $P_i \in \mathcal{P}$;
3. a function $\beta : \mathcal{P} \to [2m]$ such that $\beta(P_i) \geq \text{vol}(P_i)$ for each $P_i \in \mathcal{P}$.

Here $\mathcal{P}$ records the restriction of a partition to $\mathcal{D}(t)$, α keeps track of the number of edges captured so far in each of the liquid parts, and β keeps track of the volume so far of each of the liquid parts. Notice that the total number of possible states for any node t is at most $(k+1)^{(k+1)} \cdot m^{(k+1)} \cdot (2m)^{(k+1)} = m^{\mathcal{O}(k)}$.

For each possible state of a node t, we need to keep track of the maximum contribution to modularity from frozen parts we can achieve consistent with the liquid parts having the specified state: this is done with a function σ_t, the *signature* of t. Given any state $(\mathcal{P}, \alpha, \beta)$ of t, we first define a $(t, \mathcal{P}, \alpha, \beta)$-*partition* to be any partition $\mathcal{A}$ of V_t such that:

1. $\mathcal{P} = \mathcal{A}[\mathcal{D}(t)]$;
2. for all $A \in \mathcal{A}$ with $A \cap \mathcal{D}(t) \neq \emptyset$:

    - $\alpha(A \cap \mathcal{D}(t)) = e(A)$, and
    - $\beta(A \cap \mathcal{D}(t)) = \text{vol}(A)$.

We then set

$$\sigma_t(\mathcal{P}, \alpha, \beta) = \max \left\{ \frac{1}{m} \sum_{B \in \mathcal{B}} e(B) - \frac{1}{m^2} \sum_{B \in \mathcal{B}} \text{vol}(B)^2 : \mathcal{A} \text{ is a } (t, \mathcal{P}, \alpha, \beta)\text{-partition} \right.$$

$$\left. \text{and } \mathcal{B} = \{A \in \mathcal{A} : A \cap \mathcal{D}(t) = \emptyset\} \right\}.$$

Throughout the proof we adopt the convention that the maximum value of an empty set is $-\infty$.

It is clear that, with knowledge of σ_r for the root r of the tree decomposition, we can easily determine the maximum modularity of G. It therefore remains to outline how we compute σ_t for the four types of node in the nice tree decomposition, using only information about the values of $\sigma_{t'}$ where t' is a child of t. We begin by observing that if t is a leaf node then we can exhaustively consider all possibilities in time depending only on k.

Now suppose t is an introduce node with child t', where $\mathcal{D}(t) = \mathcal{D}(t') \cup \{v\}$. Given any state $(\mathcal{P}, \alpha, \beta)$ of t, we say that a state $(\mathcal{P}', \alpha', \beta')$ of t' is introduce-compatible with $(\mathcal{P}, \alpha, \beta)$ if:

- $\mathcal{P}' = \mathcal{P}\backslash\{v\}$;
- for every $P \in \mathcal{P}$, if $v \notin P$ then $\alpha'(P) = \alpha(P)$, and if $v \in P$ (but $P\backslash\{v\} \neq \emptyset$) then $\alpha'(P) = \alpha(P) - |\{u \in P : uv \in E(G)\}|$;
- for every $P \in \mathcal{P}$, if $v \notin P$ then $\beta'(P) = \beta(P)$, and if $v \in P$ (but $P\backslash\{v\} \neq \emptyset$) then $\beta'(P) = \beta(P) - d(v)$.

It then follows that $\sigma_t(\mathcal{P}, \alpha, \beta)$ is equal to

$$\max\{\sigma_{t'}(\mathcal{P}', \alpha', \beta') : (\mathcal{P}', \alpha', \beta') \text{ is introduce-compatible with } (\mathcal{P}, \alpha, \beta)\}.$$

Next, suppose that t is a forget node with child t', where $\mathcal{D}(t) = \mathcal{D}(t')\backslash\{v\}$. Given any state $(\mathcal{P}, \alpha, \beta)$ of t, we define two functions σ_t^1 and σ_t^2; these functions correspond to the case where one of the parts that is liquid at t' becomes frozen at t (if v was the last vertex in its part), and the case where all parts that are liquid at t' remain liquid at t, respectively. We set

$$\sigma_t^1(\mathcal{P}, \alpha, \beta) = \max\left\{\sigma_{t'}(\mathcal{P}', \alpha', \beta') + \frac{1}{m}\alpha'(\{v\}) - \frac{1}{4m^2}\beta'(\{v\})^2 : \right.$$

$$\mathcal{P}' = \mathcal{P} \cup \{v\} \text{ and, for all } P \in \mathcal{P},$$

$$\left. \alpha'(P) = \alpha(P) \text{ and } \beta'(P) = \beta(P)\right\},$$

and

$$\sigma_t^2(\mathcal{P}, \alpha, \beta) = \max\left\{\sigma_{t'}(\mathcal{P}', \alpha', \beta') : \mathcal{P} = \mathcal{P}'\backslash\{v\}, |\mathcal{P}'| = |\mathcal{P}| \text{ and,} \right.$$

$$\text{for all } P \in \mathcal{P}', \alpha'(P) = \alpha(P\backslash v)$$

$$\left. \text{and } \beta'(P) = \beta(P\backslash v)\right\}.$$

We then see that

$$\sigma_t(\mathcal{P}, \alpha, \beta) = \max\left\{\sigma_t^1(\mathcal{P}, \alpha, \beta), \sigma_t^2(\mathcal{P}, \alpha, \beta)\right\}.$$

Finally, suppose that t is a join node with children t_1 and t_2, where $\mathcal{D}(t_1) = \mathcal{D}(t_2) = \mathcal{D}(t)$. In this case we see that

$$\sigma_t(\mathcal{P}, \alpha, \beta) = \max\left\{\sigma_{t_1}(\mathcal{P}, \alpha_1, \beta_1) + \sigma_{t_2}(\mathcal{P}, \alpha_2, \beta_2) : \text{ for all } P \in \mathcal{P}, \right.$$

$$\alpha(P) = \alpha_1(P) + \alpha_2(P) - e(P) \text{ and}$$

$$\left. \beta(P) = \beta_1(P) + \beta_2(P) - \text{vol}(P)\right\}.$$

$\square$

To obtain our FPT approximation result, we use a very similar approach; the key is to restrict our attention to partitions with only a constant number of parts. For any

constant $c \in \mathbb{N}$, we write $q_{\leq c}(G)$ for the maximum modularity for G achievable with a partition into at most c parts, that is

$$q_{\leq c}(G) = \max_{|\mathcal{A}| \leq c} q_{\mathcal{A}}(G).$$

We refer to the problem of deciding whether $q_{\leq c}(G) \geq q$ for a given input graph G and constant $q \in [0, 1]$ as c-MODULARITY. We now argue that c-MODULARITY is in FPT parameterised by the treewidth of the input graph. The crucial difference from our XP algorithm above is the fact that, when we fix the number of parts in the partition, we can no longer assume that every part is connected. However, if the maximum number of parts c is a constant, we can keep track of the necessary statistics for every possible part, not just those that intersect the bag under consideration.

Lemma 2 *c-MODULARITY is in* FPT *when parameterised by the treewidth of the input graph.*

Proof The strategy is broadly the same as that used in the proof of Theorem 6, however when the number of parts is fixed we can no longer assume that every part in the optimal partition is connected. Thus, instead of recording statistics relating to each part that intersects the bag currently under consideration, we keep track of the same statistics for each of the c (possibly empty) parts allowed in the partition. Formally, for any node $t \in V(T)$, a valid state of t consists of:

1. a function $\pi : \mathcal{D}(t) \rightarrow [c]$;
2. a function $\alpha : [c] \rightarrow [m]$ such that $\alpha(i) \geq e(\pi^{-1}(i))$ for all $i \in [c]$;
3. a function $\beta : [c] \rightarrow [2m]$ such that $\beta(i) \geq \mathrm{vol}(\pi^{-1}(i))$ for all $i \in [c]$.

Here π records the mapping of vertices of $\mathcal{D}(t)$ to the c possible parts, α keeps track of the number of edges captured so far in each of the c parts, and β the volume so far of each part. Notice that the number of possible states for any node t is at most $c^{k+1} \cdot m^c \cdot (2m)^c = c^{k+1} m^{\mathcal{O}(c)}$.

Given any state (π, α, β) of t, we define a (t, π, α, β)-*partition* to be any partition $\mathcal{A} = \{A_1, \ldots, A_c\}$ of V_t such that:

1. $v \in A_{\pi(v)}$ for each $v \in \mathcal{D}(t)$;
2. for each $i \in [c]$:

   – $\alpha(i) = e(A_i)$, and
   – $\beta(i) = \mathrm{vol}(A_i)$.

We then set

$$\theta_t(\pi, \alpha, \beta) = \begin{cases} 1 & \text{if there exists a } (t, \pi, \alpha, \beta)\text{-partition of } V_t, \\ 0 & \text{otherwise.} \end{cases}$$

It is clear that, if r is the root of the tree decomposition,

$$q_{\leq c}(G) = \max_{\theta_r(\pi, \alpha, \beta) = 1} \left\{ \frac{1}{m} \sum_{i=1}^{c} \alpha(i) + \frac{1}{m^2} \sum_{i=1}^{c} \beta(i)^2 \right\}.$$

Thus it suffices to compute all values of θ_r. Note that if t is a leaf node we can consider all possibilities in time depending only on k and c; we now outline how to compute the values of θ_t for a node t, given the values for its children.

Suppose first that t is an introduce node with child t', where $\mathcal{D}(t) = \mathcal{D}(t') \cup \{v\}$. Given any state (π, α, β) of t, we say that a state (π', α', β') of t' is introduce-compatible with (π, α, β) if:

- $\pi' = \pi|_{\mathcal{D}(t')}$;
- for every $i \in [c]$, if $\pi(v) \neq c$ then $\alpha'(i) = \alpha(i)$, and if $\pi(v) = i$ then $\alpha'(i) = \alpha(i) - |\{u \in \pi^{-1}(i) : uv \in E(G)\}|$;
- for every $i \in [c]$, if $\pi(v) \neq i$ then $\beta'(i) = \beta(i)$, and if $\pi(v) = P$ then $\beta'(i) = \beta(i) - d(v)$.

It then follows that $\theta_t(\mathcal{P}, \alpha, \beta)$ is equal to

$$\max\{\theta_{t'}(\pi', \alpha', \beta') : (\pi', \alpha', \beta') \text{ is introduce-compatible with } (\pi, \alpha, \beta)\}.$$

Next, suppose that t is a forget node with child t', where $\mathcal{D}(t) = \mathcal{D}(t')\backslash\{v\}$. In this case we have

$$\theta_t(\pi, \alpha, \beta) = \max\{\theta_{t'}(\pi', \alpha', \beta') : \pi = \pi'|_{\mathcal{D}(t)}, \alpha' = \alpha \text{ and } \beta' = \beta\}.$$

Finally, suppose that t is a join node with children t_1 and t_2, where $\mathcal{D}(t_1) = \mathcal{D}(t_2) = \mathcal{D}(t)$. In this case we see that

$$\theta_t(\pi, \alpha, \beta) = \max\left\{\theta_{t_1}(\pi, \alpha_1, \beta_1) \cdot \theta_{t_2}(\pi, \alpha_2, \beta_2) : \right.$$
$$\forall i \in [c], \alpha(i) = \alpha_1(i) + \alpha_2(i) - e(\pi^{-1}(i))$$
$$\left. \text{and } \beta(i) = \beta_1(i) + \beta_2(i) - \text{vol}(\pi^{-1}(i))\right\}.$$

□

Recall (Fact 1) that $q^*(G) \geq q_{\leq c}(G) > q^*(G)\left(1 - \frac{1}{c}\right)$; thus, for any constant $\epsilon > 0$, we obtain a factor $(1 - \varepsilon)$-approximation by solving $\lceil\frac{1}{\epsilon}\rceil$-MODULARITY. This immediately gives the following result.

Corollary 1 *Given any constant $\epsilon > 0$, there is an FPT-algorithm, parameterised by the treewidth of the input graph G, that returns a partition $\mathcal{A}$ with $q_{\mathcal{A}}(G) > (1 - \varepsilon)q^*(G)$.*

We conclude this section by noting that sparse graphs, in particular graphs G with low tree width, $\text{tw}(G)$, and maximum degree, $\triangle(G)$, can have high maximum modularity. In particular Theorem 1.11 of [21] shows $q^*(G) \geq 1 - 2((\text{tw}(G) + 1)\triangle(G)/|E(G)|)^{1/2}$.

2.3 Parameterisation by Max Leaf Number

In this section we demonstrate that MODULARITY can be solved in time linear in the number of connected subgraphs of the input graph G; as a consequence of this result, we deduce that the problem belongs to XP when parameterised by the max leaf number of G.

Theorem 7 *Let G be a graph on n vertices with m edges and at most h connected subgraphs. Then* MODULARITY *can be solved in time $\mathcal{O}(h^2 n)$.*

Proof We will assume without loss of generality (by Fact 3) that G contains no isolated vertices. For any induced subgraph H of G, and partition $\mathcal{A}_H$ of $V(H)$, we write

$$q_{\mathcal{A}_H}(H, G) = \frac{1}{m} \sum_{A \in \mathcal{A}_H} e(A) - \frac{1}{4m^2} \sum_{A \in \mathcal{A}_H} \text{vol}(A)^2,$$

where $\text{vol}(A)$ denotes the volume of A in G. We then set

$$q^*(H, G) = \max_{\mathcal{A}_H} q_{\mathcal{A}_H}(H, G),$$

where the maximum is taken over all partitions $\mathcal{A}_H$ of $V(H)$. Thus, $q^*(H, G)$ can be seen as the maximum possible contribution of parts contained in H to the modularity of G, if we only consider partitions of $V(G)$ such that every part is either completely contained in $V(H)$ or does not intersect $V(H)$.

Let H be a connected subgraph of G. Then, for any partition $\mathcal{A}_H$ of $V(H)$ with $|\mathcal{A}_H| > 1$, such that each part induces a connected subgraph, it is clear that there exists a partition (X, Y) of $V(H)$ into two nonempty sets such that $H[X]$ and $H[Y]$ are both connected, and every element of $\mathcal{A}_H$ is completely contained in either X or Y. Conversely, if (X, Y) is a partition with this property it is immediate that partitions of X and Y can be combined to give a partition of $V(H)$. For any connected graph H, we write $\mathcal{P}(H)$ for the set of all partitions (X, Y) of $V(H)$ into two non-empty sets such that $G[X]$ and $G[Y]$ are both connected. Since we need only consider partitions in which every part induces a connected subgraph (by Fact 2), it follows that

$$q^*(H, G) = \max \left\{ \left(\frac{1}{m} e(H) - \frac{1}{m^2} \text{vol}(H)^2 \right), \right.$$
$$\left. \max_{(X,Y) \in \mathcal{P}(H)} \left\{ q^*(G[X], G) + q^*(G[Y], G) \right\} \right\}, \qquad (4)$$

again adopting the convention that the maximum, taken over an empty set, is equal to $-\infty$.

By assumption, G has only h connected induced subgraphs. We note that, with suitable data structures, we can compute a list of all such subgraphs in time $\mathcal{O}(nh)$. To enumerate all connected induced subgraphs containing the vertex v, we can explore a search tree as follows: we associate the pair $(\{v\}, V(G)\backslash\{v\})$ with the root and, on

reaching a node associated with the pair (U, W), we select an arbitrary vertex $x \in W$ such that $N(x) \cap U \neq \emptyset$ (if such a vertex exists), and create two child nodes associated with $(U \cup \{x\}, W\backslash\{x\})$ and $(U, W\backslash\{x\})$ respectively. When this process terminates, the vertex-set of every connected induced subgraph appears as the first element of the tuple for exactly one leaf node. Repeating the process for each vertex in the graph (after deleting those starting vertices already considered) will produce a list of all connected induced subgraphs.

From now on we will assume that we have computed a list $H_1, \ldots, H_h$ of all connected induced subgraphs of G; without loss of generality we may further assume that these subgraphs are listed in non-decreasing order of their number of vertices. In particular, this means that there is no connected induced subgraph that is strictly contained in H_1, so $\mathcal{P}(H_1) = \emptyset$ and $q^*(H_1, G) = \frac{1}{m}e(H_1) - \frac{1}{m^2}\text{vol}(H_1)^2$. We can reformulate (4) as follows:

$$q^*(H_j, G) = \max \left\{ \left(\frac{1}{m}e(H_j) - \frac{1}{m^2}\text{vol}(H_j)^2 \right), \\ \max_{\substack{i < j \\ V(H_i) \subset V(H_j) \\ H_j \backslash V(H_i) \text{ connected}}} \left\{ q^*(H_i, G) + q^*(H_j \backslash V(H_i), G) \right\} \right\}.$$

Note that, if $H_j \backslash V(H_i)$ is connected, then $H_j \backslash V(H_i)$ is H_ℓ for some $\ell < j$. Thus, if we know the values $q^*(H_1, G), \ldots, q^*(H_{j-1}, G)$, we can compute $q^*(H_j, G)$ in time $\mathcal{O}(j|H_j|)$. It follows that, by considering the connected subgraphs $H_1, \ldots, H_h$ in order, we can compute $q^*(H)$ for every connected induced subgraph in time $\mathcal{O}(h^2 n)$.

Now suppose that G has connected components $C_1, \ldots, C_\ell$, where $V(C_i) = V_i$ for each i. By Fact 2 (see also Lemma 1.6.2 of [27]), we can restrict our attention to partitions $\mathcal{A}$ of $V(G)$ such that every part is completely contained in some V_i,

$$q^*(G) = \sum_{i=1}^{\ell} q^*(C_i, G).$$

Since each connected component C_i is a connected induced subgraph of G, it occurs in the list $H_1, \ldots, H_h$ of connected induced subgraphs. Thus, once we have computed $q^*(H, G)$ for each connected induced subgraph H, we can immediately determine $q^*(G)$ by summing the appropriate values. Hence the overall time required to compute $q^*(G)$ is $\mathcal{O}(h^2 n)$.   □

It is known that, if the max leaf number of G is c, then G is a subdivision of some graph H on at most $4c$ vertices [13]; a graph on n vertices that is a subdivision of such a graph H has at most $2^{4c}n^{(4c)^2}$ connected subgraphs (once we have decided which branch vertices belong to a subgraph, it remains only to decide where to cut each path from one of the chosen branch vertices to one we have not chosen). Thus, if the max leaf number of G is bounded by a constant it follows that G has at most a polynomial number of connected subgraphs, and the following result is an immediate consequence of Theorem 7.

Corollary 2 MODULARITY *is in* XP *when parameterised by the max leaf number of the input graph G.*

We conjecture that this result is not optimal, and that MODULARITY is in fact in FPT with respect to this parameterisation.

3 Hardness results

In this section we complement our positive result about the FPT approximability of the problem parameterised by treewidth by demonstrating that computing the exact value of the maximum modularity is hard even in a more restricted setting.

Theorem 8 MODULARITY, *parameterised simultaneously by the pathwidth and the size of a minimum feedback vertex set for the input graph, is* W[1]-*hard.*

Our proof of this result relies on the hardness of the following problem.

EQUITABLE CONNECTED PARTITION (ECP)
Input: A graph $G = (V, E)$ and $r \in \mathbb{N}$.
Question: Is there a partition of V into r classes $V_1, \ldots, V_r$ such that $|V_i| - |V_j| \leq 1$ for all $1 \leq i < j \leq r$, and the induced subgraph $G[V_i]$ is connected for each $i \in 1, \ldots, r$?

The parameterised complexity of ECP was investigated thoroughly in [12]. Among other results, the problem is shown to be W[1]-hard even when parameterised simultaneously by r, $\mathrm{pw}(G)$ and $\mathrm{fvs}(G)$. In proving this hardness result, the authors implicitly consider the following variation of ECP.

ANCHORED EQUITABLE CONNECTED PARTITION (AECP)
Input: A graph $H = (V_H, E_H)$, and a set of distinguished *anchor vertices* $a_1, \ldots, a_r \in V$.
Question: Is there a partition of V_H into r classes $V_1, \ldots, V_r$ such that $a_i \in V_i$ for all i, $\left| |V_i| - |V_j| \right| \leq 1$ for all $1 \leq i < j \leq r$, and the induced subgraph $G[V_i]$ is connected for each $i \in 1, \ldots, r$?

From the proof of [12, Theorem 1] we can extract the following statement about the hardness of AECP.

Lemma 3 ([12], implicit in proof of Theorem 1) *AECP is* W[1]-*hard, parameterised simultaneously by* $\mathrm{pw}(H)$ *and* $\mathrm{fvs}(H)$, *even if the following conditions hold simultaneously:*

1. *H is connected;*
2. *the graph H' obtained from H by deleting all vertices of degree one is a subdivision of a 3-regular graph $\tilde{H}$;*

3. *the branch vertices of H' (i.e. vertices of $\tilde{H}$) are precisely the anchor vertices $a_1, \ldots, a_r$;*
4. *$r \geq 4$ is even and divides $|V_H|$;*
5. *$H \backslash \{a_1, \ldots, a_r\}$ is a disjoint union of isolated vertices and paths with pendant edges.*

In the proof of Theorem 8, it is useful to analyse the 'per unit modularity deficit' $f_m(B)$ of vertex subsets B. For $m \geq 1$ and vertex subset B with $\mathrm{vol}(B) \geq 1$ we define

$$f_m(B) = \frac{\partial(B)}{\mathrm{vol}(B)} + \frac{\mathrm{vol}(B)}{2m}. \tag{5}$$

Intuitively, minimising the per unit modularity deficit $f_m(B)$ maximises the modularity (see (7) for a precise statement). Hence, loosely, the following lemma says that if we are restricted to parts B with $\delta(B) = 4$ the modularity maximising volume is $\mathrm{vol}(B) = 2\sqrt{2m}$. Moreover, while it would usually be better to take parts with $\delta(B) < 4$ these parts are actually worse (i.e. higher $f_m(B)$ value) if their volumes are too big or too small. The function $f_m(B)$ plays a similar role to the *n-cost* in Proposition 1 of [21].

Lemma 4 *Let $m \geq 1$, $\mathrm{vol}(B) \geq 1$ and let $f_m(B)$ be as defined in (5). Then the following properties hold:*

0: *if $\partial(B) = 0$ and $\mathrm{vol}(B) > 4\sqrt{2m}$ then $f_m(B) > 2\sqrt{2/m}$.*
1: *if $\partial(B) = 1$ and $\mathrm{vol}(B) > 3.7321\sqrt{2m}$ or $\mathrm{vol}(B) < 0.2679\sqrt{2m}$ then $f_m(B) > 2\sqrt{2/m}$.*
2: *if $\partial(B) = 2$ and $\mathrm{vol}(B) > 3.4143\sqrt{2m}$ or $\mathrm{vol}(B) < 0.5857\sqrt{2m}$ then $f_m(B) > 2\sqrt{2/m}$.*
3: *if $\partial(B) = 3$ and $\mathrm{vol}(B) > 3\sqrt{2m}$ or $\mathrm{vol}(B) < \sqrt{2m}$ then $f_m(B) > 2\sqrt{2/m}$.*
4: *if $\partial(B) = 4$ and $\mathrm{vol}(B) \geq 2\sqrt{2m}$ then $f_m(B) \geq 2\sqrt{2/m}$ with equality iff $\mathrm{vol}(B) = 2\sqrt{2m}$.*
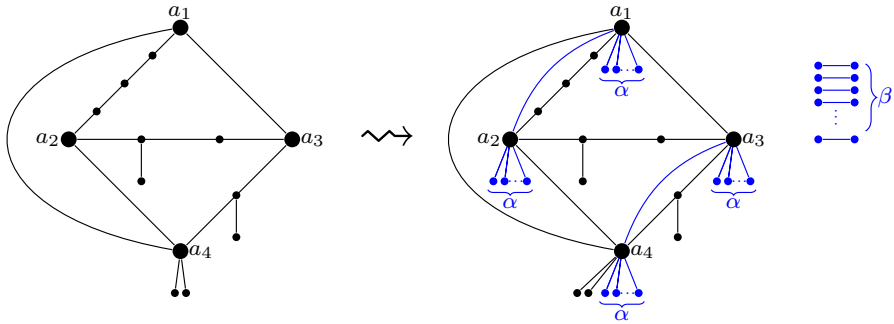5: *if $\partial(B) \geq 5$ then $f_m(B) > 2\sqrt{2/m}$.*

Proof Fix a vertex set B with a constant number, ℓ, of edges to the rest of the graph (so $\partial(B) = \ell$). For $\ell = 0$ one can check that directly that if $\mathrm{vol}(B) > 4\sqrt{2m}$ then $f_m(B) > 2\sqrt{2/m}$ which establishes part 0. Thus we may assume $\ell \geq 1$. By definition, $f_m(B) = \ell/\mathrm{vol}(B) + \mathrm{vol}(B)/(2m)$ and so

$$f_m(B) \geq 2\sqrt{\frac{2}{m}} \quad \Leftrightarrow \quad \left(\frac{\mathrm{vol}(B)}{\sqrt{2m}} - 2\right)^2 \geq 4 - \ell. \tag{6}$$

Hence for $\ell = 4$ we get equality iff $\mathrm{vol}(B) = 2\sqrt{2m}$ which immediately implies part 4 of the lemma. Also for $\ell \geq 5$ the RHS of (6) is negative which gives part 5 of the lemma. It remains to prove parts 1, 2 and 3 of the lemma.

Now suppose $\ell \in \{1, 2, 3\}$ then $\sqrt{4 - \ell}$ is real and so we may rearrange as the difference of two squares,

$$f_m(B) > 2\sqrt{\frac{2}{m}} \quad \Leftrightarrow \quad \left(\frac{\mathrm{vol}(B)}{\sqrt{2m}} - 2 - \sqrt{4 - \ell}\right)\left(\frac{\mathrm{vol}(B)}{\sqrt{2m}} - 2 + \sqrt{4 - \ell}\right) > 0.$$

Fig. 2 Possible input graph H with anchors a_1, a_2, a_3, a_4 and the graph G constructed from it by adding α new leaves adjacent to each anchor, β isolated edges and a perfect matching between the anchors

Observe $f_m(B) > 2\sqrt{2/m}$ if the terms in the product above are either both positive or both negative. Hence $f_m(B) > 2\sqrt{2/m}$ if

$$\text{vol}(B) > 2\sqrt{2m}\left(1 + \tfrac{1}{2}\sqrt{4 - \ell}\right) \qquad \text{or} \qquad \text{vol}(B) < 2\sqrt{2m}\left(1 - \tfrac{1}{2}\sqrt{4 - \ell}\right).$$

Therefore for $\ell = 1$ we get $f_m(B) > 2\sqrt{2m}$ if $\text{vol}(B) > \sqrt{2m}(2 + \sqrt{3})$ and $2 + \sqrt{3} \sim 3.7320508 \leq 3.7321$. Likewise, keeping $\ell = 1$, $f_m(B) > 2\sqrt{2m}$ if $\text{vol}(B) < \sqrt{2m}(2 - \sqrt{3})$ and $2 - \sqrt{3} \sim 0.26794919 \geq 0.2679$. This establishes part 1 of the lemma. The parts 2 and 3 follow in the same fashion. □

We are now ready to prove Theorem 8.

Proof of Theorem 8 We give a reduction from AECP. Suppose that $(H, \{a_1, \ldots, a_r\})$ is the input to an instance of AECP; we will describe how to construct a graph G, where $\text{pw}(G)$ and $\text{fvs}(G)$ are both bounded by a function of r, together with an explicit $q_0 \in (0, 1)$ such that (G, q_0) is a yes-instance for MODULARITY if and only if $(H, \{a_1, \ldots, a_r\})$ is a yes-instance for AECP.

We may assume without loss of generality that our instance of AECP satisfies all of the conditions of Lemma 3.

We define a new graph G, obtained from H by adding the following (see Fig. 2):

- α new leaves adjacent to each anchor vertex $a_1, \ldots, a_r$,
- β isolated edges disjoint from G, and
- an arbitrary perfect matching on the anchor vertices $a_1, \ldots, a_r$,

where the values of α and β will be determined later.

The idea of the construction is that the α edges help ensure that each anchor vertex is in a separate part of any modularity optimal partition and the β edges allow us to get the numbers to work at the end of the proof. Notice that, even with these modifications, $G \setminus \{a_1, \ldots, a_r\}$ is still a disjoint union of isolated vertices and paths with pendant edges; hence $\text{pw}(G) \leq r + 1$ and $\text{fvs}(G) \leq r$. We set $m = |E(G)|$ so $m = |E(H)| + \alpha r + \beta + r/2$.

Define our instance of MODULARITY to be (G, q_0), where

$$q_0 = 1 - \frac{\beta}{m^2} - \frac{2\sqrt{2}(m - \beta)}{m^{3/2}}.$$

We now argue that (G, q_0) is a yes-instance if and only if $(H, \{a_1, \ldots, a_r\})$ is a yes-instance for AECP. Recall that

$$q^*(G) = 1 - \min_{\mathcal{A}} \sum_{A \in \mathcal{A}} \left(\frac{\partial(A)}{2m} + \frac{\text{vol}(A)^2}{4m^2} \right),$$

and that the partition $\mathcal{A}$ which achieves the minimum in the expression above is exactly the modularity maximal $\mathcal{A}$. In any modularity optimal partition, $\mathcal{A}$, each isolated edge will form its own part: this follows from Facts 2 and 4. Write V' for vertices of G without the vertices supporting the β isolated edges, and let the minimisation be over $\mathcal{A}'$ which are vertex partitions of V'. We then have

$$q^*(G) = 1 - \frac{\beta}{m^2} - \min_{\mathcal{A}'} \sum_{A \in \mathcal{A}'} \frac{\partial(A)}{2m} + \frac{\text{vol}(A)^2}{4m^2}.$$

Rearranging, we see that

$$1 - \frac{\beta}{m^2} - q^*(G') = \frac{m - \beta}{m} \min_{\mathcal{A}'} \sum_{A \in \mathcal{A}'} \frac{\text{vol}(A)}{2(m - \beta)} \left(\frac{\partial(A)}{\text{vol}(A)} + \frac{\text{vol}(A)}{2m} \right)$$

$$= \frac{m - \beta}{m} \min_{\mathcal{A}'} \sum_{A \in \mathcal{A}'} \frac{\text{vol}(A)}{2(m - \beta)} f_m(A) \tag{7}$$

$$\geq \frac{m - \beta}{m} \min_{A \subseteq V'} f_m(A). \tag{8}$$

The last inequality holds because $\sum_A \text{vol}(A) = 2(m - \beta)$ and so (7) is a weighted sum of the $f_m(A)$ with total weight one. This, together with the fact that no A has zero volume, also implies that (7) $\geq$ (8) with equality if and only if $f_m(A) = \min_{B \subset V'} f_m(B)$ for every $A \in \mathcal{A}'$.

Note that, since $\mathcal{A}'$ is the restriction of some modularity optimal partition $\mathcal{A}$ to a connected component of G, we may assume that, for all $A \in \mathcal{A}'$, $G[A]$ is connected. Moreover, if v is a pendant vertex adjacent to u then u and v are in the same part in $\mathcal{A}'$; we call a partition with this last property (or, abusing notation, a set that would not violate this condition in a partition) 'pendant-consistent'.

We now make the following claim, writing $s = |H|/r$ for the desired part size in our instance of AECP.

Claim 9 *Suppose that $\alpha > 32|E(H)|^2$ and that we have $\sqrt{2m} = s + \alpha + 1$. Then:*

a) *for any connected, pendant-consistent set $B \subseteq V'$ we have $f_m(B) \geq 2\sqrt{2/m}$, and if $f_m(B) = 2\sqrt{2/m}$ then B contains exactly one anchor and $\text{vol}(B) = 2\sqrt{2m}$;*

b) *if $(H, \{a_1, \ldots, a_r\})$ is a yes-instance, then there is a vertex partition $\mathcal{A}'$ of V' so that $f_m(A) = 2\sqrt{2/m}$ for all $A \in \mathcal{A}'$;*

c) *if there is a vertex partition $\mathcal{A}' = \{A_1, \ldots, A_r\}$ of V' so that for all $A_i \in \mathcal{A}$, $f_m(A_i) = 2\sqrt{2/m}$, A is pendant-consistent and $G[A]$ is connected for all $A \in \mathcal{A}'$, then $(H, \{a_1, \ldots, a_r\})$ is a yes-instance.*

We defer the proof of Claim 9. For now we assume that Claim 9 holds and that we have $\alpha > 32|E(H)|^2$ and $\sqrt{2m} = s + \alpha + 1$ and prove the theorem holds under these assumptions. By Claim 9(a) and line (7), we have

$$q^*(G) \le q_0 = 1 - \frac{\beta}{m^2} - \frac{2\sqrt{2}(m - \beta)}{m^{3/2}}.$$

Hence in particular (G, q_0) is a yes-instance if and only if there is a partition $\mathcal{A}'$ of V' such that $\forall A \in \mathcal{A}'$ $f_m(A) = 2\sqrt{2/m}$.

Claim 9(b), together with line (7), implies that, if $(H, \{a_1, \ldots, a_r\})$ is a yes-instance, then so is (G, q_0). Conversely, if (G, q_0) is a yes-instance, it follows from Claim 9(c), that $(H, \{a_1, \ldots, a_r\})$ is a yes-instance.

It remains only to show the claim holds and we can choose suitable values of α and β to ensure $\alpha > 32|E(H)|^2$ and $\sqrt{2m} = s + \alpha + 1$. Set α to be the least integer such that

$$\alpha \ge 32|E(H)|^2, \quad (\alpha + s + 1)^2 > 2|E(H)| + 2\alpha r + r \quad \text{and} \quad \alpha = s + 1 \pmod 2. \quad (9)$$

Recall that r is even. This, along with our parity constraint between α and s, implies that $(\alpha + s + 1)^2 - r$ is even. Thus we can choose β to be

$$\beta = \frac{1}{2}\left((\alpha + s + 1)^2 - r\right) - |E(H)| - \alpha r; \quad (10)$$

note β is positive because we set α so that $(\alpha + s + 1)^2 > 2|E(H)| + 2\alpha r + r$. Finally, observe that we do have $\sqrt{2m} = s + \alpha + 1$ because, by the chosen value of β,

$$m = |E(H)| + \alpha r + \beta + r/2 = (s + \alpha + 1)^2/2.$$

This concludes the proof of the theorem except that we must still establish Claim 9.

Proof of Claim 9(a): We begin by showing that our two assumptions $\alpha > 32|E(G)|^2$ and $\alpha + s + 1 = \sqrt{2m}$ imply that $\alpha > 0.969\sqrt{2m}$. Recall that H without pendant edges is a subdivision of a cubic graph and so the average degree in H is at least two. Thus $|E(H)| \ge |H|$. Also $r \ge 4$, so $|H| \ge 4s \ge s + 1$ so $|E(H)| \ge s + 1$. By assumption $\alpha > 32|E(H)|^2 \ge 32(s + 1)$. But also by assumption $\alpha + s + 1 = \sqrt{2m}$ and so $\alpha \ge 32/33\sqrt{2m} > 0.969\sqrt{2m}$.

We now show that if $f_m(B) \le 2\sqrt{2/m}$ then B must contain exactly one anchor. First suppose B contains no anchors: then B does not contain nor is B incident to any of the α extra edges added to anchors nor the $r/2$ extra edges in the perfect matching between anchors. Hence the volume of B in G is at most what it was in H,

i.e. $\mathrm{vol}_G(B) \leq \mathrm{vol}_H(B) \leq 2|E(H)|$. Also note that as $G[V']$ is connected, $\partial(B) \geq 1$, hence by Lemma 4 it is enough to show that $\mathrm{vol}(B) < 0.2679\sqrt{2m}$ and this will show that for B with no anchors, $f_m(B) > 2\sqrt{2/m}$. Clearly $m \geq \alpha$ and by assumption $\alpha > 32|E(H)|^2$. Hence for B with no anchors:

$$0.2679\sqrt{2m} \geq 0.2679\sqrt{64|E(H)|^2} = 2.1432 \ E(H) > \mathrm{vol}(B)$$

and so if $f_m(B) \leq 2\sqrt{2/m}$ then B must contain at least one anchor.

If B contains at least two anchors then there are two options: $B = V'$ and $B \subsetneq V'$. We rule out $B = V'$ and $f_m(B) \leq 2\sqrt{2/m}$ first. Note that $\mathrm{vol}(V') = 2|E(H)| + 2\alpha r + r$. But $r \geq 4$ and by earlier in the proof $\alpha > 0.969\sqrt{2m}$. Hence $\mathrm{vol}(V') \geq 8\alpha > 7.752\sqrt{2m}$ and so by Lemma 4 we get that $f_m(V') > 2\sqrt{2/m}$. Thus $B \neq V'$.

Now we show that for $B \subsetneq V'$ with at least two anchors in B we have $f_m(B) > 2\sqrt{2/m}$. In the case $B \neq V'$ because $G[V']$ is connected $\partial(B) \geq 1$. If B has at least two anchors then $\mathrm{vol}(B) \geq 4\alpha > 3.876\sqrt{2m}$. Therefore for $B \subsetneq V'$ with at least two anchors in B, $\partial(B) \geq 1$ and $\mathrm{vol}(B) > 3.878\sqrt{2m}$ hence $f_m(B) > 2\sqrt{2/m}$ by Lemma 4.

Thus to ensure $f_m(B) \leq 2\sqrt{2/m}$ we must have exactly one anchor in B. In particular we can now assume that B contains exactly one anchor. Let graph G' be G without the added perfect matching between anchors at the end of the construction of G from H. Now $G'[B]$ is connected, B has exactly one anchor and after stripping pendant vertices that anchor has degree 3 in G' so we have $\partial_{G'}(B) \geq 3$. And because B is pendant-consistent $\partial_{G'}(B) = 3$, after re-adding the perfect matching between anchors $\partial_G(B) = 4$.

But now, because $\partial_G(B) = 4$, by Lemma 4 we have that $f_m(B) \geq 2\sqrt{2/m}$. Also by Lemma 4 to get equality $f_m(B) = 2\sqrt{2/m}$ we must have $\mathrm{vol}(B) = 2\sqrt{2m}$ which establishes the last part of the claim. $\qquad \Box$ *Claim 9(a)*

Proof of Claim 9(b): Suppose $(H, \{a_1, \ldots, a_r\})$ is a yes-instance. We prove there exists a vertex partition $\mathcal{A}'$ of V' such that, for all $A \in \mathcal{A}'$, $f_m(A) = 2\sqrt{2/m}$. By assumption, there is a connected equipartition $\mathcal{B} = \{B_1, \ldots, B_r\}$ of $V(H)$ such that $a_i \in B_i$ for each i. In the construction of the graph G from H we added α pendant vertices, say $u_1^i, \ldots, u_\alpha^i$, to each anchor a_i. Define $\mathcal{A}' = \{B_i \cup \{u_1^i, \ldots, u_\alpha^i\} : B_i \in \mathcal{B}\}$. Observe that that $\mathcal{A}'$ is a vertex partition of V' as the set V' consists exactly of $V(H)$ together with the extra αr vertices added with the pendant edges on each anchor. It now remains to prove that $f_m(A_i) = 2\sqrt{2/m}$ for each i.

Consider G', the graph formed from G by removing the arbitary perfect matching added in the last step of the construction of G from H. Recall the graph G' is the subdivision of a 3-regular graph with the anchors as the branch vertices. Fix i and note that $H[B_i]$ connected implies that $G'[A_i]$ is connected. But as $G'[A_i]$ is connected, contains exactly one anchor and contains every vertex pendant to a vertex in A_i it must be the case that $\partial_{G'}(A_i) = 3$. Now re-add the perfect matching and we get that $\partial_G(A_i) = 4$.

It suffices now to show that $\mathrm{vol}(A_i) = 2\sqrt{2m}$. To see this, first observe that $G[A_i]$ is a tree, and so $|E_G(A_i)| = |A_i| - 1$. But by the construction $|A_i| = |B_i| + \alpha = s + \alpha$. Recall the volume of a vertex set is twice the number of internal edges plus the number

of edges between the set and the rest of the graph. Thus, because $\partial_G(A_i) = 4$, we get

$$\mathrm{vol}(A_i) = 2(s + \alpha - 1) + 4 = 2(s + \alpha + 1) = 2\sqrt{2m},$$

which establishes the claim.   □ *Claim 9(b)*

Proof of Claim 9(c): Suppose there exists a vertex partition $\mathcal{A}' = \{A'_1, \ldots, A'_r\}$ of V' such that, for all $A'_i \in \mathcal{A}'$, $f_m(A'_i) = 2\sqrt{2/m}$, A'_i is pendant-consistent, and $G[A'_i]$ is connected. By Claim 9(a) we may also assume that for all $A'_i \in \mathcal{A}'$ we have $\mathrm{vol}(A'_i) = 2\sqrt{2m}$. We will show this implies that $(H, \{a_1, \ldots, a_r\})$ is a yes-instance.

Fix some i. The induced subgraph $G[A'_i]$ is connected and contains exactly one anchor, say a_i, so we can remove the perfect matching between the anchors and $G'[A'_i]$ is still connected. Let B_i be the vertex set obtained from A'_i by removing the α added leaves pendant on the anchor a_i. Then $B_i \subseteq V(H)$ and $H[B_i]$ is connected.

It remains only to show that $|B_i|$ is exactly $s = |H|/r$. Since $G(A'_i)$ is a tree with volume $2\sqrt{2/m}$ and $\partial_G(A'_i) = 4$, $\mathrm{vol}(A'_i) = 2(|A_i| - 1) + 4 = 2|A_i| + 2$. But $|A_i| = |B_i| + \alpha$ and so

$$|B_i| = |A_i| - \alpha = \mathrm{vol}(A'_i)/2 - 1 - \alpha;$$

by design this is precisely $\mathrm{vol}(A'_i)/2 - 1 - \alpha = s$ and so we are done.

                                                                          □ *Claim 9(c)*

This completes the proof.   □

4 Conclusions and Open Problems

We have shown that MODULARITY belongs to FPT when parameterised by the vertex cover number of the input graph, and that the problem is solvable in polynomial time on input graphs whose treewidth or max leaf number is bounded by some fixed constant; we also showed that there is an FPT algorithm, parameterised by treewidth, which computes any constant-factor approximation to the maximum modularity. In contrast with the positive approximation result, we demonstrated that the problem is unlikely to admit an exact FPT algorithm when the treewidth is taken to be the parameter, as it is W[1]-hard even when parameterised simultaneously by the pathwidth and size of a minimum feedback vertex set for the input graph.

We conjecture that our XP algorithm parameterised by max leaf number is not optimal, and that MODULARITY in fact belongs to FPT with respect to this parameterisation. Another open question arising from our work is whether the problem belongs to FPT with respect to other parameters for which this is not ruled out by our hardness result, including treedepth, modular width and neighbourhood diversity.

It is also natural to ask whether our approximation result can be extended to larger classes of graphs, for example those of bounded cliquewidth or bounded expansion. Moreover, when considering treewidth as the parameter, it would be interesting to investigate the existence or otherwise of an ϵ-approximation in time $f(\mathrm{tw}, \epsilon)n^{\mathcal{O}(1)}$.

References

1. Bagrow, J.P.: Communities and bottlenecks: trees and treelike networks have high modularity. Phys. Rev. E **85**(6), 066118 (2012)
2. Blondel, V.D., Guillaume, J.L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. J. Stat. Mech Theory Exp. **2008**(10), P10008 (2008)
3. Bodlaender, H.L.: A linear time algorithm for finding tree-decompositions of small treewidth. In: Proceedings of the Twenty-Fifth Annual ACM Symposium on Theory of Computing, STOC '93, pp. 226–234. ACM, New York, NY, USA (1993). https://doi.org/10.1145/167088.167161
4. Brandes, U., Delling, D., Gaertler, M., Gorke, R., Hoefer, M., Nikoloski, Z., Wagner, D.: On modularity clustering. IEEE Trans. Knowl. Data Eng. **20**(2), 172–188 (2008). https://doi.org/10.1109/TKDE.2007.190689
5. Cygan, M., Fomin, F.V., Kowalik, Ł., Lokshtanov, D., Marx, D., Pilipczuk, M., Pilipczuk, M., Saurabh, S.: Parameterized Algorithms. Springer, Cham (2015)
6. DasGupta, B., Desai, D.: On the complexity of newman's community finding approach for biological and social networks. J. Comput. Syst. Sci. **79**(1), 50–67 (2013). https://doi.org/10.1016/j.jcss.2012.04.003
7. de Montgolfier, F., Soto, M., Viennot, L.: Asymptotic modularity of some graph classes. In: Asano, T., Nakano, S., Okamoto, Y., Watanabe, O. (eds.) Algorithms and Computation. ISAAC 2011. Lecture Notes in Computer Science, vol. 7074. Springer, Berlin, Heidelberg (2011)
8. Dinh, T.N., Li, X., Thai, M.T.: Network clustering via maximizing modularity: Approximation algorithms and theoretical limits. In: 2015 IEEE International Conference on Data Mining, pp. 101–110 (2015). https://doi.org/10.1109/ICDM.2015.139
9. Dinh, T.N., Thai, M.T.: Finding community structure with performance guarantees in scale-free networks. In: Privacy, Security, Risk and Trust (PASSAT) and 2011 IEEE Third Inernational Conference on Social Computing (SocialCom), 2011 IEEE Third International Conference on, pp. 888–891. IEEE (2011)
10. Dinh, T.N., Thai, M.T.: Community detection in scale-free networks: approximation algorithms for maximizing modularity. IEEE J. Sel. Areas Commun. **31**(6), 997–1006 (2013). https://doi.org/10.1109/JSAC.2013.130602
11. Downey, R.G., Fellows, M.R.: Fundamentals of Parameterized Complexity. Springer, London (2013)
12. Enciso, R., Fellows, M.R., Guo, J., Kanj, I., Rosamond, F., Suchý, O.: What makes equitable connected partition easy. In: Chen, J., Fomin, F.V., (eds.) Parameterized and Exact Computation: 4th International Workshop, IWPEC 2009, Copenhagen, Denmark, September 10–11, 2009, Revised Selected Papers, pp. 122–133. Springer, Berlin, (2009). https://doi.org/10.1007/978-3-642-11269-0_10
13. Estivill-Castro, V., Fellows, M., Langston, M., Rosamond, F.: FPT is P-time extremal structure I. In: Algorithms and Complexity in Durham 2005, Proceedings of the first ACiD Workshop, volume 4 of Texts in Algorithmics, pp. 1–41. King's College Publications (2005)
14. Fortunato, S.: Community detection in graphs. Phys. Rep. **486**(3), 75–174 (2010)
15. Fortunato, S., Hric, D.: Community detection in networks: a user guide. Phys. Rep. **659**, 1–44 (2016)
16. Jutla, I.S., Jeub, L.G.S., Mucha, P.J.: A generalized Louvain method for community detection implemented in MATLAB. (2011) http://netwiki.amath.unc.edu/GenLouvain
17. Kawase, Y., Matsui, T., Miyauchi, A.: Additive Approximation Algorithms for Modularity Maximization. In: Hong, S.H. (ed.) 27th International Symposium on Algorithms and Computation (ISAAC 2016), Leibniz International Proceedings in Informatics (LIPIcs), vol. 64, pp. 43:1–43:13. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany (2016). https://doi.org/10.4230/LIPIcs.ISAAC.2016.43

18. Kloks, T.: Treewidth—Computations and Approximations, Lecture Notes in Computer Science, vol. 842. Springer, Berlin (1994)
19. Lancichinetti, A., Fortunato, S.: Limits of modularity maximization in community detection. Phys. Rev. E **84**(6), 066122 (2011)
20. Lokshtanov, D.: Parameterized integer quadratic programming: Variables and coefficients. (2015) arXiv:1511.00310 [cs.DS]
21. McDiarmid, C., Skerman, F.: Modularity of regular and treelike graphs. J. Complex Netw. **6**, 596 (2017)
22. McDiarmid, C., Skerman, F.: Modularity of Erdős-Rényi random graphs. Random Struct. Algorithms, (to appear)
23. McDiarmid, C., Skerman, F.: Modularity of Erdős-Rényi random graphs. In: 29th International Conference on Probabilistic, Combinatorial and Asymptotic Methods for the Analysis of Algorithms, vol. 1 (2018)
24. Newman, M.E.J., Girvan, M.: Finding and evaluating community structure in networks. Phys. Rev. E **69**(2), 026113 (2004)
25. Porter, M., Onnela, J.P., Mucha, P.: Communities in networks. Not AMS **56**(9), 1082–1097 (2009)
26. Prokhorenkova, L.O., Prałat, P., Raigorodskii, A.: Modularity of models of complex networks. Electron. Notes Discrete Math. **61**, 947–953 (2017)
27. Skerman, F.: Modularity of networks. Ph.D. thesis, University of Oxford (2016)