Modularity Clustering Parameterized by Max Leaf Number

Jaroslav Garvardt
Institute of Computer Science, Friedrich Schiller University Jena, Germany

Christian Komusiewicz
Institute of Computer Science, Friedrich Schiller University Jena, Germany

Abstract

The modularity score is one of the most important measures for assessing the quality of clusterings of undirected graphs. In the notoriously difficult MODULARITY problem, one is given an undirected graph G and the task is to find a clustering with maximum modularity. We show that MODULARITY is fixed-parameter tractable with respect to the max leaf number of G. This improves on a previous result by Meeks and Skerman [Algorithmica '20] who showed an XP-algorithm for this parameter. In addition, we strengthen previous hardness results for MODULARITY by showing W[1]-hardness for the parameter vertex deletion distance to disjoint union of stars.

2012 ACM Subject Classification Theory of computation → Fixed parameter tractability

Keywords and phrases Graph clustering, parameterized complexity

Digital Object Identifier 10.4230/LIPIcs.IPEC.2024.16

Funding *Jaroslav Garvardt*: Supported by the Carl Zeiss Foundation, Germany, within the project "Interactive Inference".

1 Introduction

One of the most central topics in network science is the detection of community structure [18]. This can be achieved via graph clustering. In its most simple form, this is the task of searching for a partition of the vertex set into clusters and the underlying assumption is that edges are more likely to be present inside clusters than between them. The corresponding optimization goal is to maximize the edge coverage of the partition, that is, the number of intracluster edges. Of course, the trivial partition into one single cluster trivially maximizes the number of intracluster edges. To counter this, several approaches have been proposed, for example one may demand that clusters form highly connected subgraphs [11, 12] or one may penalize missing edges inside clusters [19].

The arguably most popular way of achieving clusterings with high edge coverage while avoiding the trivial clustering is to maximize modularity [17]. In the modularity measure, the contribution of a cluster consists of two parts: the first part corresponds to the edge coverage and the second part is a degree tax which penalizes clusters that have many high-degree vertices. The idea behind the degree tax is that such clusters are expected to contain many edges simply because there are many edges that are incident with the cluster vertices. Hence, for a positive contribution to the modularity score, the number of present edges should exceed the number of expected ones. More formally, a clustering of a graph $G = (V, E)$ is a partition of V and for a vertex set $C \subseteq V$, $E(C)$ denotes the set of edges with both endpoints in C. Now the definition of modularity reads as follows.

▶ **Definition 1.1.** *The* modularity *of a clustering $\mathcal{C}$ of a graph G with m edges is given by*

$$q(\mathcal{C}) = \sum_{C \in \mathcal{C}} \frac{|E(C)|}{m} - \frac{\left(\sum_{v \in C} \deg(v)\right)^2}{4m^2}.$$

© Jaroslav Garvardt and Christian Komusiewicz;
licensed under Creative Commons License CC-BY 4.0

19th International Symposium on Parameterized and Exact Computation (IPEC 2024).
Editors: Édouard Bonnet and Paweł Rzążewski; Article No. 16; pp. 16:1–16:14
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

The problem of computing an optimal clustering under this measure is now defined as follows.

Modularity
Input: A graph $G = (V, E)$.
Task: Find a clustering $\mathcal{C}$ with maximum modularity $q(\mathcal{C})$.

Modularity is NP-hard [4] and therefore heuristics, for example greedy algorithms [5] or local search [3], are prevalent in practice. The modularity measure has some counterintuitive behavior [4]. Consequently, some research focuses on better understanding the properties of optimal clusterings [4] or on providing bounds for the modularity values of certain graph classes [1, 14, 15, 8, 20]; for an overview, refer to the work of Skerman [20].

The importance of the modularity measure has motivated further research into the complexity of Modularity. For example, computing the best clustering with exactly two clusters is also NP-hard [4] even when the input graphs are restricted to be d-regular for any $d \geq 9$ [7]. Meeks and Skerman [16] initiated the analysis of Modularity within the framework of parameterized complexity obtaining the following results: They showed that Modularity can be solved in polynomial time on graphs with constant treewidth and that Modularity is fixed-parameter tractable with respect to the vertex cover number of the input graph G. Moreover, it was shown that Modularity is W[1]-hard with respect to the parameter pathwidth of G plus feedback vertex set number of G, so presumably there exists no FPT-algorithm for this parameter. For the parameter max leaf number of G, denoted $\lambda(G)$, an XP-algorithm was shown. That is, the algorithm has a polynomial running time for constant values of $\lambda(G)$, but the degree of the polynomial depends on $\lambda(G)$. The precise parameterized complexity of Modularity with respect to the max leaf number of G was left open.

In this work, we continue this line of research. We show that the XP-algorithm for the max leaf number $\lambda(G)$ can be improved to an FPT-algorithm. While the max leaf number, one of the most classic structural parameters [10], is quite restrictive, our result provides only the second nontrivial FPT-algorithm for this very important problem. Roughly speaking, our algorithm exploits that large graphs with bounded max leaf number contain very long paths consisting of degree-2 vertices and that the clustering for these paths follows a relatively regular pattern. The algorithm consists of three steps. In a first branching, the global structure of the clustering is constrained. In particular, it is determined how the vertices of degree at least 3 are clustered and with which degree-2 paths these clusters share vertices. To prepare the next step, it is shown that the clusters consisting of degree-2 vertices have roughly the same size and, based on this, that the clusters containing high-degree vertices also deviate by at most $22\,\lambda(G)^2$ from the size of the path clusters. This allows us to find the correct cluster sizes via branching. The remaining problem of computing an optimal clustering under these size constraints is then solved via an ILP formulation.

Our algorithm works also on disconnected graphs G, where we define $\lambda(G)$ to be the sum of the max leaf numbers of the connected components of G. The only proofs where we assume connectivity are those that bound the number of high-degree vertices in terms of $\lambda(G)$ and they are easily seen to also hold for disconnected graphs by summing over the connected components.

On the negative side, we strengthen the previous W[1]-hardness for Modularity parameterized by pathwidth plus feedback vertex set number by showing that Modularity is W[1]-hard with respect to the vertex deletion distance of G to a disjoint union of stars. This parameter is obviously at least as large as the feedback vertex set number of G. Moreover, the parameter is also lower-bounded by pathwidth + 1: Any vertex deletion set S to a disjoint union of stars gives a path decomposition where every bag contains S plus a star center plus one leaf of the star.

In our opinion, this W[1]-hardness for distance to stars puts the FPT-algorithm for the admittedly large parameter max leaf number into context by underlining once more that Modularity is resistant to quite large structural parameterizations. Due to space constraints for statements marked with a star (*), the proofs are deferred to the long version of this work.

2 Preliminaries

We consider undirected graphs $G = (V, E)$ and let n denote the number of vertices of G and m the number of edges of G. The *neighborhood* of a vertex $v \in V$ is defined as $N(v) = \{u \in V \mid \{u, v\} \in E\}$ and for a set of vertices $V' \subseteq V$ we define $N(V') = (\bigcup_{v \in V'} N(v)) \setminus V'$. The *degree* of a vertex v is denoted by $\deg(v) = |N(v)|$. We define $V_{=1} = \{v \in V \mid \deg(v) = 1\}$ and analogously $V_{=2} = \{v \in V \mid \deg(v) = 2\}$ and $V_{\geq 3} = \{v \in V \mid \deg(v) \geq 3\}$.

We denote the degree sum of a vertex set C, also called *volume* of C, by $\mathrm{vol}(C) := \sum_{v \in C} \deg(v)$. For two clusterings $\mathcal{C}$ and $\mathcal{C}'$ we say that $\mathcal{C}$ is *better* than $\mathcal{C}'$ if $q(\mathcal{C}) > q(\mathcal{C}')$. Since we are often not interested in the actual value of the modularity of a clustering, but only whether it is better than another clustering, we define the function $\widetilde{q}(\mathcal{C}) = 4m^2 q(\mathcal{C}) = 4m \sum_{C \in \mathcal{C}} |E(C)| - \sum_{C \in \mathcal{C}} \mathrm{vol}(C)^2$. Clearly, for two clusterings $\mathcal{C}$ and $\mathcal{C}'$ of the same graph we have $q(\mathcal{C}) \geq q(\mathcal{C}')$ if and only if $\widetilde{q}(\mathcal{C}) \geq \widetilde{q}(\mathcal{C}')$.

A 2-*path* is a path $(v_1, v_2, \ldots, v_k)$ with $\deg(v_i) \leq 2$ for all $i \in [k]$. A 2-path is *maximal* if it is not contained in a longer 2-path. For a maximal 2-path $P = (v_1, v_2, \ldots, v_k)$ we refer to v_1 and v_k as the endpoints of P and define $V(P) = \{v_1, v_2 \ldots, v_k\}$. A 2-path is *pendent* if $\deg(v_1) = 1$ or $\deg(v_k) = 1$. A *branch* of a graph G is a maximal path or cycle in which every *internal* vertex of the path has degree 2 in G. We denote with $\mathcal{B}_G$ the set of all branches in G and with $\beta(G) = |\mathcal{B}_G|$ the number of all branches in G. Note that $\beta(G)$ can be computed in $\mathcal{O}(n + m)$ time. Let G be a connected graph. The *maximum leaf number* $\lambda(G)$ of G (or just *max leaf number*) is the maximum number of leaves in any spanning tree of G. When the graph G is clear from the context we just write λ for the max leaf number.

▶ **Lemma 2.1** ([4]). *There is always a clustering with maximum modularity, in which each cluster induces a connected subgraph.*

▶ **Lemma 2.2** ([4]). *A clustering with maximum modularity has no cluster that consists of a single vertex with degree 1.*

In our correctness proofs, we are often concerned with the effect of removing one vertex from some cluster C_i and adding some vertex to another cluster C_j. In particular, we are interested in the change of the total degree tax for these two clusters. The following lemma describes a situation where the degree tax decreases.

▶ **Lemma 2.3.** *Let C_i and C_j be two clusters and let u and v be two vertices of the same degree such that $u \in C_i$ and $\deg(u) > 0$. If $\mathrm{vol}(C_i \setminus \{u\}) > \mathrm{vol}(C_j)$, then $\mathrm{vol}(C_i)^2 + \mathrm{vol}(C_j)^2 > \mathrm{vol}(C_i \setminus \{u\})^2 + \mathrm{vol}(C_j \cup \{v\})^2$.*

Proof. The claim holds trivially if $v \in C_j$, thus assume $v \notin C_j$. Since $u \in C_i$, we have $\mathrm{vol}(C_i) = \mathrm{vol}(C_i \setminus \{u\}) + \deg(u)$. Therefore,

$$\mathrm{vol}(C_i)^2 = (\mathrm{vol}(C_i \setminus \{u\}) + \deg(u))^2 = \mathrm{vol}(C_i \setminus \{u\})^2 + 2 \cdot \mathrm{vol}(C_i \setminus \{u\}) \cdot \deg(u) + \deg(u)^2.$$

Since $\deg(u) = \deg(v)$, we similarly have

$$\mathrm{vol}(C_j \cup \{v\})^2 = (\mathrm{vol}(C_j) + \deg(u))^2 = \mathrm{vol}(C_j)^2 + 2 \cdot \mathrm{vol}(C_j) \cdot \deg(u) + \deg(u)^2.$$

Thus,

$$\begin{aligned}&\operatorname{vol}(C_i)^2+\operatorname{vol}(C_j)^2-(\operatorname{vol}(C_i\setminus\{u\})^2+\operatorname{vol}(C_j\cup\{v\})^2)\\ =\;&2\cdot\operatorname{vol}(C_i\setminus\{u\})\cdot\deg(u)+\deg(u)^2-(2\cdot\operatorname{vol}(C_j)\cdot\deg(u)+\deg(u)^2)\\ =\;&2\cdot\deg(u)\cdot(\operatorname{vol}(C_i\setminus\{u\})-\operatorname{vol}(C_j))>0.\end{aligned}$$ ◀

We now show that the number of vertices $v \in V$ with $\deg(v) \geq 3$ is bounded by a function of the max leaf number. More precisely, we give a bound on the sum of the degrees of these vertices. This bound is obtained via bounding the number of branches in G. Eppstein [9] already showed that this number is $\mathcal{O}(\lambda(G)^2)$. We give a precise bound on the hidden constant since we will use it in our algorithm.

▶ **Lemma 2.4 (*).** *Let $G = (V, E)$ be a connected graph. Then we have $\beta(G) \leq 21\,\lambda(G)^2$.*

The next statement directly follows from Lemma 2.4, since each edge incident with a vertex $v \in V_{\geq 3}$ corresponds to a branch and each branch contains at most two vertices in $V_{\geq 3}$.

▶ **Corollary 2.5.** *Let $G = (V, E)$ be a connected graph. Then we have $\sum_{v\in V_{\geq 3}} \deg(v) \leq 42\,\lambda(G)^2$.*

3 Preclusterings and Cluster Sizes

3.1 Preclustering Branching

The general approach of the algorithm is to consider in a branching step all the possibilities of how some optimal clustering might interact with the vertices of degree at least 3. The structure containing this information is a partial clustering defined as follows.

▶ **Definition 3.1.** *A* preclustering *is a set $\mathcal{P} := \{C_1, \ldots, C_s\}$ of nonempty disjoint subsets of V.*

▶ **Definition 3.2.** *A clustering $\mathcal{C} = \{C'_1, \ldots, C'_t\}$* extends *a preclustering $\mathcal{P} = \{C_1, \ldots, C_s\}$ if for each $C_i \in \mathcal{P}$ there is exactly one cluster $C'_j \in \mathcal{C}$ such that $C_i \subseteq C'_j$ and each cluster $C'_j \in \mathcal{C}$ has nonempty intersection with at most one cluster of $\mathcal{P}$.*

To find an optimal solution efficiently from a preclustering, we will need to fix not only which vertices of degree 3 are contained in which clusters but also how these clusters interact with the potentially very long 2-paths connecting them. The necessary information is provided by what we call full preclusterings, defined as follows (see Figure 1 for an example).

▶ **Definition 3.3.** *A preclustering $\mathcal{P} = \{C_1, \ldots, C_s\}$ is a* full *preclustering if every vertex of $V_{\geq 3}$ is contained in some cluster of $\mathcal{P}$ and for every maximal 2-path $P = (v_1, v_2, \ldots, v_k)$ of G either*

- *all vertices of P are contained in some common cluster C_i,*
- *no vertex of P is contained in any cluster C_i, or*
- *$k \geq 2$, for each cluster C_i we have $C_i \cap V(P) \subseteq \{v_1, v_k\}$ and if $v_1 \in C_i$ then also $u_1 \in C_i$, where u_1 is the unique neighbor of v_1 in $V_{\geq 3}$, and if $v_k \in C_i$ then also $u_k \in C_i$, where u_k is the unique neighbor of v_k in $V_{\geq 3}$.*

The idea of full preclusterings is as follows. For the fully contained maximal 2-paths P, the cluster is already fixed. For the endpoints of the other maximal 2-paths, we know that they are either 1) in different clusters than their high-degree neighbors which helps us to separate the instance in smaller pieces, or 2) in the same cluster as their high-degree neighbors which allows us to use the clusters with these high-degree vertices in some exchange arguments because they also contain some degree-2 vertices.

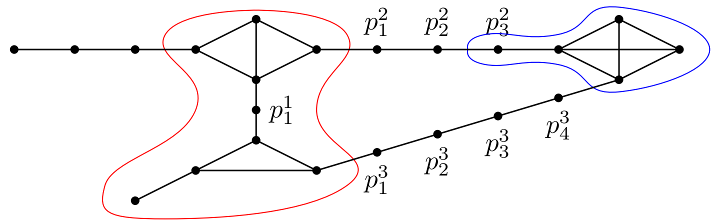

Figure 1 Example of a full preclustering $\mathcal{P} = \{C_1, C_2\}$. The cluster C_1 is encircled in red, the cluster C_2 is encircled in blue. Cluster C_1 contains the complete maximal 2-path (p_1^1), cluster C_2 contains one endpoint of the maximal 2-path (p_1^2, p_2^2, p_3^2) and no vertex of the maximal 2-path $(p_1^3, p_2^3, p_3^3, p_4^3)$ is contained in any cluster in $\mathcal{P}$.

▶ **Definition 3.4.** *A clustering $\mathcal{C}$* legally extends *a full preclustering $\mathcal{P}$ if*

- *$\mathcal{C}$ extends $\mathcal{P}$,*
- *a cluster $C \in \mathcal{C}$ contains $\{u, v\}$ where u is an endpoint of a maximal 2-path P and v is a neighbor of u in $V_{\geq 3}$ only if some cluster of $\mathcal{P}$ does.*

Note that for a given clustering $\mathcal{C}$, there is exactly one full preclustering $\mathcal{P}$ such that $\mathcal{C}$ legally extends $\mathcal{P}$. We say that $\mathcal{P}$ is the preclustering that *corresponds* to $\mathcal{C}$.

Let us first show that we may indeed consider all full preclusterings within FPT time.

▶ **Lemma 3.5.** *Any graph G has $\lambda(G)^{\mathcal{O}(\lambda(G)^2)}$ full preclusterings.*

Proof. A full preclustering $\mathcal{P}$ can be identified by

1. the partition of $V_{\geq 3}$ that it induces,
2. for each 2-path, the information whether that 2-path is fully contained in some cluster of $\mathcal{P}$, disjoint from all clusters of $\mathcal{P}$, or whether its endpoints are contained in some cluster of $\mathcal{P}$.

In the latter case, the cluster which contains an endpoint is uniquely determined to be the cluster containing the neighbor of the endpoint in $V_{\geq 3}$. By Corollary 2.5, the number of vertices in $V_{\geq 3}$ is $\mathcal{O}(\lambda(G)^2)$ and thus the number of partitions of $V_{\geq 3}$ is $\lambda(G)^{\mathcal{O}(\lambda(G)^2)}$. By Lemma 2.4, the number of branches and thus the number of 2-paths is $\mathcal{O}(\lambda(G)^2)$. For each 2-path, we need to distinguish altogether five cases, hence there are $2^{\mathcal{O}(\lambda(G)^2)}$ possibilities for the 2-path information. The total number of full preclusterings is thus $\lambda(G)^{\mathcal{O}(\lambda(G)^2)} \cdot 2^{\mathcal{O}(\lambda(G)^2)} = \lambda(G)^{\mathcal{O}(\lambda(G)^2)}$. ◀

A full preclustering $\mathcal{P}$ constrains some edges of the graph to not be contained in any cluster of a clustering that legally extends $\mathcal{P}$. This set of edges is defined as follows.

▶ **Definition 3.6.** *Let $\mathcal{P} = \{C_1, \ldots, C_s\}$ be a full preclustering of G. The* separation induced by $\mathcal{P}$ *is the edge set*

$$S(\mathcal{P}) := \bigcup_{i \in [s]} \{\{u, v\} \in E \mid u \in C_i \cap V_{\geq 3} \text{ and } v \notin C_i\}.$$

As the name suggests a separation fully separates some parts of the instance. These are exactly the connected components of $G - S(\mathcal{P})$, they are called the *separated components* of $\mathcal{P}$. By Lemma 2.1 it is sufficient to consider clusterings such that every cluster induces a connected subgraph. For any such clustering $\mathcal{C}$ that legally extends a full preclustering $\mathcal{P}$, we have that every cluster C is completely contained in some separated component of $\mathcal{P}$. We thus compute an optimal clustering of each separated component of $\mathcal{P}$ individually.

We will distinguish those clusters that contain at least one vertex from $V_{\geq 3}$, these are called *base clusters*, from those clusters that are contained in the 2-paths, these are called *path clusters.* The two main parts that are not yet determined by a full preclustering are how far each base cluster extends into the neighboring 2-paths and how large the clusters which are fully contained in 2-paths are. The next step is now to show that the 2-path clusters inside a separated component have roughly the same size. Note that this is not true for all full preclusterings but rather that there is some preclustering which has a globally optimal legal extension for which this is the case.

▶ **Lemma 3.7.** *There exists an optimal clustering $\mathcal{C}$ such that*

- *$\mathcal{C}$ legally extends some full preclustering $\mathcal{P}$, and*
- *in every separated component S of $\mathcal{P}$, there is some number p such that the path clusters in S have size p or $p + 1$.*

The approach to show Lemma 3.7 is, roughly speaking, to show that a big size difference between path clusters leads to suboptimality because we can exchange some degree-2 vertices to balance the cluster sizes. This exchange may need to involve base clusters which contain some vertices of 2-paths. To distinguish whether a base cluster contains some vertices of a 2-path or not, we say a base cluster C *extends* into a 2-path P if $|C \cap P| \geq 1$. Note that per definition a base cluster C extends into a 2-path P if and only if in the corresponding preclustering there is a cluster $C_P \subseteq C$ such that $|C_P \cap P| \geq 1$.

▶ **Definition 3.8.** *Two clusters $C \in \mathcal{C}$ and $C' \in \mathcal{C}$ are* neighboring clusters *or* neighbors *if $\{u, v\} \in E$ for some $u \in C, v \in C'$.*

For a clustering $\mathcal{C}$ with neighboring path clusters $C_1 = \{u_i, \dots, u_j\}$ and $C_2 = \{u_{j+1}, \dots, u_{j+\ell}\}$ on a 2-path $P = (u_1, \dots, u_t)$, we define the clustering $\mathcal{C}'$ obtained by the *swap* of C_1 and C_2 as the clustering that is the same as $\mathcal{C}$ except for clusters C_1 and C_2 which are replaced by the clusters $C_1' = \{u_i, \dots, u_{i+\ell-1}\}$ and $C_2' = \{u_{i+\ell}, \dots, u_{j+\ell}\}$. In other words, the swap exchanges the lengths of two neighboring path clusters. Clearly, the clustering resulting from applying a swap to $\mathcal{C}$ has the same modularity as $\mathcal{C}$.

3.2 Difference in Cluster Sizes is bounded in λ

We now prove a series of lemmas which are needed for the proof of Lemma 3.7. We distinguish those path clusters that contain a degree-1 vertex which we call *pendent* path clusters and those that do not contain a degree-1 vertex which are called *nonpendent* path clusters.

Note that a separated component consisting of a 2-path with two vertices of degree 1 is an isolated path, a graph with constant treewidth, for which the optimal clustering can be computed directly in polynomial time [16]. We therefore assume in the following that there is at most one pendent path cluster per 2-path.

The first lemma shows that in optimal solutions, pendent clusters are at least as large as neighboring nonpendent clusters.

▶ **Lemma 3.9 (*).** *Let $G = (V, E)$ be a graph and $\mathcal{C}$ a clustering of G. Let P be a pendent 2-path and $C_1 \in \mathcal{C}$ and $C_2 \in \mathcal{C}$ be path clusters in P such that C_2 is pendent, C_1 is nonpendent, and $|C_1| > |C_2|$. Then, $\mathcal{C}$ is not optimal.*

The next lemma shows that path clusters from the same 2-path in G can only differ in size by at most one vertex.

▶ **Lemma 3.10.** *Let $G = (V, E)$ be a graph and $\mathcal{C}$ a clustering of G. Let P be a 2-path and $C_1 \in \mathcal{C}$ and $C_2 \in \mathcal{C}$ be path clusters in P with $|C_1| > |C_2| + 1$. Then, $\mathcal{C}$ is not optimal.*

Proof. Recall that we can assume that not both of C_1 and C_2 are pendent, since otherwise P is an isolated path for which the optimal clustering can be computed directly in polynomial time. Since P is a 2-path, we can swap neighboring clusters of P without changing the modularity, so we can assume that C_1 and C_2 are neighboring clusters.

Let $v_1 \in C_1 \cap N(C_2)$ be the neighbor of C_2 in C_1. Consider the clustering $\mathcal{C}'$ where C_1 and C_2 are replaced by clusters $C_1' = C_1 \setminus \{v_1\}$ and $C_2' = C_2 \cup \{v_1\}$, respectively, and all other clusters are unchanged.

Note that we only have to consider the contribution of C_1 and C_2 to $\widetilde{q}(\mathcal{C})$ and the contribution of C_1' and C_2' to $\widetilde{q}(\mathcal{C}')$ since the other clusters are identical for both clusterings. Furthermore, since v_1 is part of a 2-path, we have $|E(C_1')| = |E(C_1)| - 1$ and $|E(C_2')| = |E(C_2)| + 1$, so the total number of intracluster edges remains the same.

Moreover, by Lemma 3.9, we may assume that C_2 is nonpendent. Now, if C_1 is nonpendent, then $\mathrm{vol}(C_1 \setminus \{v_1\}) = 2|C_1| - 2 > 2|C_2| = \mathrm{vol}(C_2)$ since $|C_1| > |C_2| + 1$. Thus Lemma 2.3 implies $\mathrm{vol}(C_1)^2 + \mathrm{vol}(C_2)^2 > \mathrm{vol}(C_1 \setminus \{v_1\})^2 + \mathrm{vol}(C_2 \cup \{v_1\})^2$. Finally, if C_1 is pendent, then again $\mathrm{vol}(C_1 \setminus \{v_1\}) = 2|C_1| - 3 \geq 2|C_2| + 1 > 2|C_2| = \mathrm{vol}(C_2)$ since $|C_1| > |C_2| + 1$ and thus $|C_1| \geq |C_2| + 2$. Hence, $\mathcal{C}'$ is a better clustering. ◀

The next lemma shows that two path clusters with the same base cluster as a neighbor can only differ in size by at most one vertex.

▶ **Lemma 3.11 (*).** *Let $G = (V, E)$ be a graph and $\mathcal{C}$ a clustering of G. Let $C \in \mathcal{C}$ be a base cluster that extends into a 2-path P_1 and a 2-path P_2. Let $C_{P_1} \in \mathcal{C}$ be a path cluster in P_1 and $C_{P_2} \in \mathcal{C}$ be a path cluster in P_2 with $|C_{P_1}| > |C_{P_2}| + 1$. Then, $\mathcal{C}$ is not optimal.*

We are now ready to show Lemma 3.7.

▶ **Lemma 3.7.** *There exists an optimal clustering $\mathcal{C}$ such that*

- *$\mathcal{C}$ extends some full preclustering $\mathcal{P}$, and*
- *in every separated component of $\mathcal{P}$, there is some number p such that the path clusters have size p or $p + 1$.*

For the proof of Lemma 3.7 we need the following definition.

▶ **Definition 3.12.** *Let $\mathcal{C}$ be a clustering that extends some full preclustering $\mathcal{P}$ and let S be a separated component of $\mathcal{P}$. Let C and $\widetilde{C}$ be path clusters in S contained in the 2-paths P and $\widetilde{P}$, respectively. Since C and $\widetilde{C}$ are part of the same separated component, there is a smallest number $\ell \geq 1$ for which there is a sequence of 2-paths $P_1, \ldots, P_\ell$ and a sequence of extended base clusters $B_1, \ldots, B_{\ell-1}$ such that $P = P_1$, $\widetilde{P} = P_\ell$ and B_i extends into P_i and P_{i+1} for each $i \in [\ell - 1]$. We then say that C and $\widetilde{C}$ have* path cluster distance *ℓ.*

Proof (of Lemma 3.7). Assume towards a contradiction that for every optimal clustering $\mathcal{C}$ and its corresponding full preclustering $\mathcal{P}$, there is a separated component of $\mathcal{P}$ that contains path clusters C_1 and C_2 with $|C_1| - |C_2| \geq 2$. We choose $\mathcal{C}$ in such a way that there is a separated component S of $\mathcal{P}$ such that S contains path clusters C and $\widetilde{C}$ with

- $p := |C|$ and $\widetilde{p} := |\widetilde{C}| = p + c$ for some $c > 1$,
- the clusters C and $\widetilde{C}$ have path cluster distance r, and
- r is the minimal path cluster distance of any two path clusters C_1, C_2 with $|C_1| - |C_2| \geq 2$ in the same separated component of any optimal clustering.

Let P be the 2-path that contains C and $\widetilde{P}$ be the 2-path that contains $\widetilde{C}$. Let $P_1, \ldots, P_r$ and $B_1, \ldots, B_{r-1}$ be the sequences of 2-paths and extended base clusters for C and $\widetilde{C}$ as described in Definition 3.12.

If $r = 1$, then we have $\widetilde{p} \in \{p, p+1\}$ by Lemma 3.10 and the optimality of $\mathcal{C}$, contradicting the assumption $\widetilde{p} = p + c$. If $r = 2$, then there is a base cluster B_1 that extends into both P and $\widetilde{P}$, so we have again $\widetilde{p} \in \{p, p+1\}$, since otherwise $\mathcal{C}$ would not be optimal according to Lemma 3.11, contradicting the assumption $\widetilde{p} = p + c$. Now consider the case $r \geq 3$. First, we show that we can assume that C and B_1 are neighboring clusters: If C is pendent and has a neighboring nonpendent path cluster C' of size $p + 1$, then the clustering $\mathcal{C}$ is not optimal by Lemma 3.9. Hence, if C is pendent and not a neighboring cluster of B_1, then we may choose the nonpendent cluster C' instead of C. Now, if C is nonpendent, since P_1 is a 2-path, we can swap neighboring clusters of P_1 until we reach B_1 without changing the modularity. Altogether, we can assume that C and B_1 are neighboring clusters. Let $\widehat{C}$ be the path cluster of P_2 neighboring B_1. Note that $|\widehat{C}| \in \{p-1, p, p+1\}$ due to Lemma 3.11. If $|\widehat{C}| \in \{p-1, p\}$, then we have a contradiction to C and $\widetilde{C}$ having minimal path cluster distance, since $|\widetilde{C}| - |\widehat{C}| \geq 2$ and $\widehat{C}$ and $\widetilde{C}$ have path cluster distance $r - 1$. Thus, let $|\widehat{C}| = p + 1$. Let $v_1 \in B_1 \cap N(C)$ be the neighbor of C in B_1 and let $v_2 \in \widehat{C} \cap N(B_1)$ be the neighbor of B_1 in $\widehat{C}$. Consider the clustering $\mathcal{C}'$ where B_1, C, and $\widehat{C}$ are replaced by clusters $B_1' = (B_1 \setminus \{v_1\}) \cup \{v_2\}$, $C' = C \cup \{v_1\}$, and $\widehat{C}' = \widehat{C} \setminus \{v_2\}$, respectively. Clearly, we have $\widetilde{q}(\mathcal{C}') = \widetilde{q}(\mathcal{C})$, so $\mathcal{C}'$ is also an optimal clustering. Note that in $\mathcal{C}'$ the clusters $\widehat{C}'$ and $\widetilde{C}$ are in the same separated component of the preclustering $\mathcal{P}'$ corresponding to $\mathcal{C}'$. Moreover, in $\mathcal{C}'$ the clusters $\widehat{C}'$ and $\widetilde{C}$ have path cluster distance $r - 1$ and $|\widehat{C}'| = p$. Altogether, we thus have a contradiction to C and $\widetilde{C}$ having minimal path cluster distance. ◀

By Lemma 3.7 for each separated component we can distinguish between *small* and *big* path clusters of size p and $p + 1$, respectively. The next lemma shows that for an extended base cluster and a neighboring path cluster the size difference is bounded by a function of $\lambda(G)$.

▶ **Lemma 3.13.** *Let $G = (V, E)$ be a graph and $\mathcal{C}$ a clustering of G. Let $C \in \mathcal{C}$ be a base cluster that extends into a 2-path P and let $C_P \in \mathcal{C}$ be a path cluster in P. If $|C_P| > |C| + 22\,\lambda(G)^2$ or $|C_P| < |C| - 2$, then $\mathcal{C}$ is not optimal.*

Proof. Without loss of generality, we may assume that C_P is a neighboring cluster of the base cluster C, as otherwise all exchanges between C and C_P can be carried out by moving the path clusters between C and C_P.

First, consider the case where $|C_P| > |C| + 22\,\lambda(G)^2$. Let $v \in C_P \cap N(C)$ be the neighbor of C in C_P. Consider the clustering $\mathcal{C}'$ where C_P and C get replaced by clusters $C_P' = C_P \setminus \{v\}$ and $C' = C \cup \{v\}$, respectively, and all other clusters are not changed.

Note that we only have to consider the contribution of C_P and C to $\widetilde{q}(\mathcal{C})$ and the contribution of C_P' and C' to $\widetilde{q}(\mathcal{C}')$ since the other clusters are identical for both clusterings. Furthermore, since v is part of a 2-path, we have $|E(C_P')| = |E(C_P)| - 1$ and $|E(C')| = |E(C)| + 1$, so the total number of intracluster edges remains the same. We can express the volume of the base cluster C as the sum $\mathrm{vol}(C) = \mathrm{vol}(C \cap V_{=1}) + \mathrm{vol}(C \cap V_{=2}) + \mathrm{vol}(C \cap V_{\geq 3})$. According to Lemma 2.5 we have $\mathrm{vol}(V_{\geq 3}) \leq 42\,\lambda(G)^2$ and therefore also $\mathrm{vol}(C \cap V_{\geq 3}) \leq 42\,\lambda(G)^2$. Thus, if C_P is nonpendent, we have

$$\mathrm{vol}(C_P \setminus \{v\}) = 2|C_P| - 2 > 2|C| + 44\,\lambda(G)^2 - 2 > \mathrm{vol}(C).$$

Similarly, if C_P is pendent, then

$$\mathrm{vol}(C_P \setminus \{v\}) = 2|C_P| - 3 > 2|C| + 44\,\lambda(G)^2 - 3 > \mathrm{vol}(C).$$

Thus, in both cases Lemma 2.3 implies $\mathrm{vol}(C_P)^2 + \mathrm{vol}(C)^2 > \mathrm{vol}(C_P \setminus \{v\})^2 + \mathrm{vol}(C \cup \{v\})^2$ and $\mathcal{C}'$ is a better clustering.

Second, consider the case $|C_P| < |C| - 2$. Let $v \in C \cap N(C_P)$ be the neighbor of C_P in C. Consider the clustering $\mathcal{C}'$ where C and C_P get replaced by clusters $C' = C \setminus \{v\}$ and $C'_P = C_P \cup \{v\}$, respectively, and all other clusters are not changed.

Again, we only have to consider the contribution of C_P and C to $\widetilde{q}(\mathcal{C})$ and the contribution of C'_P and C' to $\widetilde{q}(\mathcal{C}')$ since the other clusters are identical for both clusterings. Furthermore, note that since v is part of a 2-path, we have $|E(C'_P)| = |E(C_P)|+1$ and $|E(C')| = |E(C)|-1$, so the total number of intracluster edges remains the same. Moreover, since C is connected we have $\text{vol}(C) > 2|C| - 2$. Hence, $\text{vol}(C \setminus \{v\}) > 2|C| - 4 = 2(|C| - 2) > 2|C_P| \geq \text{vol}(C_P)$. Thus, Lemma 2.3 implies $\text{vol}(C)^2 + \text{vol}(C_P)^2 > \text{vol}(C \setminus \{v\})^2 + \text{vol}(C_P \cup \{v\})^2$ and therefore the clustering $\mathcal{C}$ is not optimal. ◀

▶ **Lemma 3.14.** *Let $G = (V, E)$ be a graph and $\mathcal{C}$ a clustering of G. Let $C \in \mathcal{C}$ and $\widehat{C} \in \mathcal{C}$ be base clusters of the same separated component such that $|C| + 22\,\lambda(G)^2 < |\widehat{C}|$. Then, $\mathcal{C}$ is not optimal.*

Proof. Let $(C = C_1, C_2, \ldots, C_t = \widehat{C})$ be a sequence of clusters such that C_i and C_{i+1} extend into the same 2-path P_i. Consider the clustering $\mathcal{C}'$ obtained as follows: Cluster C_1 gains one vertex from P_1, all path clusters on P_1 are shifted by one position on the path, C_2 loses one vertex on P_1 and gains one vertex on P_2 and so on until we reach $\widehat{C}$ which only loses one vertex on P_{t-1}.

The number of edges covered by $\mathcal{C}'$ is the same as for $\mathcal{C}$. Moreover, the only two clusters whose volume has changed are C and $\widehat{C}$ with C gaining a degree-2 vertex u and $\widehat{C}$ losing a degree-2 vertex v. By Corollary 2.5, we have $\text{vol}(C) \leq 2|C| + 42\lambda(G)^2$ and $\text{vol}(\widehat{C} \setminus \{v\}) \geq 2(|C| + 22\lambda(G)^2 - 1) = 2|C| + 44\lambda(G)^2 - 2 > 2|C| + 42\lambda(G)^2$ since $\widehat{C} \setminus \{v\}$ is connected and $|\widehat{C} \setminus \{v\}| \geq |C| + 22\lambda(G)^2$ and $\lambda(G) \geq 2$. Thus, C, $\widehat{C}$, u, and v fulfill the conditions of Lemma 2.3 and $\mathcal{C}'$ is a better clustering than $\mathcal{C}$. ◀

▶ **Lemma 3.15.** *There exists an optimal clustering $\mathcal{C}$ such that*

- *$\mathcal{C}$ extends some full preclustering $\mathcal{P}$, and*
- *in every separated component of $\mathcal{P}$ there is some number p such that each path cluster has size p or $p + 1$ and the base clusters have a size in the range $[p - 22\,\lambda^2, p + 2]$.*

Proof. Note that the second statement is true for every separated component S that does not contain any path clusters, since we can set p as the size of the largest base cluster and all other base clusters in S then have a size in the range $[p - 22\,\lambda^2, p]$ according to Lemma 3.14. Thus it is sufficient to consider separated components that contain a path cluster.

Moreover, due to Lemma 3.7, we can assume that there is a non-empty family $\mathcal{F}$ of optimal clusterings where in every separated component of the corresponding preclustering there is some number p such that the path clusters have size p or $p + 1$. We thus assume towards a contradiction that for every optimal clustering $\mathcal{C} \in \mathcal{F}$ and its corresponding full preclustering $\mathcal{P}$, there is a separated component S of $\mathcal{P}$ that contains a base cluster C of size $|C| \notin [p - 22\,\lambda^2, p + 2]$, where p and $p + 1$ are the sizes of path clusters in S.

Now, let $\mathcal{C} \in \mathcal{F}$ be an optimal clustering with its corresponding full preclustering $\mathcal{P}$ and let S be a separated component of $\mathcal{P}$ such that in S there is a base cluster C with $|C| \notin [p - 22\,\lambda^2, p + 2]$. Let P be a 2-path in S that C extends into and let C_P be a path cluster in P of size p. Since $|C_P| = p > |C| + 22\,\lambda^2$ or $|C_P| = p < |C| - 2$, according to Lemma 3.13 the clustering $\mathcal{C}$ is not optimal, a contradiction to the assumption. ◀

4 Solving Separated Components

We now show how to compute an optimal clustering extending a given full preclustering $\mathcal{P}$ under the assumption that the full preclustering can be legally extended to an optimal clustering. The algorithm considers the separated components one by one. We thus assume in the following, that we are given one separated component H. Let $C_1, \dots, C_t$ denote the base clusters of the full preclustering that are contained in H, and let $P_1, \dots, P_q$ denote the maximal 2-paths of vertices in H that are not contained in any cluster C_i. The problem is thus to determine how far the clusters extend into the paths P_i and how large the path clusters in each 2-path P_i are.

The main observations from Section 3.2 are that for each separated component there is a number p such that the path clusters have size p or $p+1$ and that the size of each base cluster C_i is in $[p - 22\,\lambda^2, p+2]$. The algorithm to compute the optimal clustering will now consist of two main steps. First, we perform a branching to fix p and the size of each base cluster. Afterwards, we formulate the problem as an ILP.

For the branching step, first observe that the number of choices for p is less than n. Now the number of different choices for the base clusters is $\lambda^{\mathcal{O}(\lambda^2)}$ since there are $\mathcal{O}(\lambda^2)$ base clusters, and for each the number of possible sizes is $22\,\lambda^2 + 3$. Hence, the total number of created branches is $n \cdot \lambda^{\mathcal{O}(\lambda^2)}$.

Now, for each branch we search for an optimal clustering of H that legally extends the preclustering and fulfills all the cluster size constraints of the branch. Let $c_1, \dots, c_t$ denote the cluster size constraints for the base clusters.

The first observation now is that the modularity of a cluster C_i' containing a cluster C_i is determined by the branch assumption: the only aspect of the cluster C_i that is not fixed by the preclustering is the total number of vertices from neighboring 2-paths of C_i that are contained in $C_i' \setminus C_i$. This number is fixed by the branching, it is precisely $c_i - |C_i|$. Each of these additional vertices contributes a value of 2 to $\mathrm{vol}(C_i')$ and one additional edge to $|E(C_i')|$. Hence, the contribution of the final clusters $C_i' \supseteq C_i$ is fixed for all base clusters C_i.

Moreover, for each path cluster C the modularity contribution is

- $q_1 := (p-1)/m - (2p)^2/4m^2$ when C is nonpendent and $|C| = p$, and
- $q_2 := p/m - (2p+2)^2/4m^2$ when C is nonpendent and $|C| = p+1$.
- $q_1' := (p-1)/m - (2p-1)^2/4m^2$ when C is pendent and $|C| = p$, and
- $q_2' := p/m - (2p+1)^2/4m^2$ when C is pendent and $|C| = p+1$.

Consequently, the only unknown quantity that influences the modularity of the clustering is the number of pendent and nonpendent path clusters that have size p and the number of pendent and nonpendent path clusters that have size $p+1$.

With this discussion in mind, we find the optimal clustering by the following ILP. For each 2-path P_i we introduce variables $x_{1,i}$ and $x_{2,i}$ representing the number of path clusters contained in P_i of size p and $p+1$, respectively. If P_i is pendent, then we also introduce variables $x_{1,i}'$ and $x_{2,i}'$ representing the number of pendent clusters of size p and $p+1$, respectively. For each 2-path P_i, we declare one endpoint to be the right endpoint of P_i and one to be the left endpoint of P_i, we also introduce variables e_i^r and e_i^ℓ that represent the number of vertices of P_i that do not belong to path clusters but to the base clusters that extend into P_i containing the right and left endpoint, respectively. Now, for a base cluster C_i, we let N_i^r denote the set of 2-paths P_j such that C_i extends from the right into P_j (that is, C_i contains the neighbor of the right endpoint of P_j) and N_i^ℓ denote the set of 2-paths P_j such that C_i extends from the left into P_j. All variables are constrained to be nonnegative integers. Then, the ILP reads as follows.

$$\begin{aligned}
&\max \sum_{P_i} q_1 \cdot x_{1,i} + q_2 \cdot x_{2,i} + q'_1 \cdot x'_{1,i} + q_2 \cdot x'_{2,i} && && (1)\\
\text{s.t.}\quad & p \cdot x_{1,i} + (p+1) \cdot x_{2,i} + e^r_i + e^\ell_i &&= |P_i| \ \forall \text{ nonpendent } P_i && (2)\\
& p \cdot x_{1,i} + (p+1) \cdot x_{2,i} + p \cdot x'_{1,i} + (p+1) \cdot x'_{2,i} + e^r_i &&= |P_i| \qquad \forall \text{ pendent } P_i && (3)\\
& \sum_{P_j \in N^r_i} e^r_j + \sum_{P_j \in N^\ell_i} e^\ell_j &&= c_i - |C_i| \qquad \forall\ C_i && (4)\\
& x'_{1,i} + x'_{2,i} &&\leq 1 \qquad \forall \text{ 1-pendent } P_i && (5)\\
& x'_{1,i} + x'_{2,i} &&\leq 2 \qquad \forall \text{ 2-pendent } P_i && (6)
\end{aligned}$$

Here, a 1-pendent path is a pendent path with one vertex of degree 1, and a 2-pendent path is a path with two vertices of degree 1.[1]

By the discussion above, the objective function (1) maximizes the modularity of the clustering for the separated component given the size constraints. Constraint (2) guarantees that the number of length-p and length-$(p+1)$ paths together with the path vertices that end up in base clusters gives the total path length for nonpendent paths. Constraint (3) guarantees the same for pendent paths. Constraint (4) guarantees that the base clusters fulfill the size constraints of the current branch. Finally, observe that Lemma 2.4 implies that the ILP has $\mathcal{O}(\lambda^2)$ variables since we have a constant number of variables for each branch of the separated component.

We now have all the necessary parts to prove the main result of this work.

▶ **Theorem 4.1.** *Modularity can be solved in* $\lambda^{\mathcal{O}(\lambda^2)} \cdot n^{\mathcal{O}(1)}$ *time.*

Proof. The algorithm enumerates all full preclusterings. For each full preclustering $\mathcal{P}$, a clustering is computed that legally extends $\mathcal{P}$. The correctness of the algorithm can be seen as follows. Fix an optimal clustering $\mathcal{C}$. Then, there is a full preclustering $\mathcal{P}$ such that $\mathcal{C}$ legally extends $\mathcal{P}$. By Lemma 3.15, there exists for each separated component of $\mathcal{P}$ a number p such that all path clusters of the component have size p or $p+1$ and the size of each base cluster C_i is in $[p - 22\,\lambda^2, p+2]$. For each separated component, the algorithm considers one branch where p and the sizes of the base clusters in the component are the same as the sizes of the corresponding clusters in $\mathcal{C}$. For this branch, the ILP computes a clustering of the component which has maximum modularity under the constraints. Thus, the modularity of the computed clustering for each separated component is the same as the modularity of $\mathcal{C}$ for this component, and the returned clustering is globally optimal.

It remains to show the running time bound. By Lemma 3.5, the number of full preclusterings is $\lambda^{\mathcal{O}(\lambda^2)}$. For each of them, the algorithm branches for each separated component into $n \cdot \lambda^{\mathcal{O}(\lambda^2)}$ cases for the sizes of the path and base clusters. For each branch, an ILP with $\mathcal{O}(\lambda^2)$ variables is solved. This can be done in $\lambda^{\mathcal{O}(\lambda^2)} \cdot n^{\mathcal{O}(1)}$ time [6]. The overall running time follows. ◀

5 Parameterization by distance to stars

In this section, we strengthen previous hardness results for Modularity by showing W[1]-hardness for the parameter vertex deletion distance to disjoint union of stars. This parameter is defined as follows. Let $G = (V, E)$ be a graph. A *modulator set to a disjoint union of*

[1] These are isolated paths for which the optimal clustering can be also computed directly in polynomial time, but for the sake of brevity, we decided to describe a unified approach that can solve all separated components.

stars for G is a set of vertices $S \subseteq V$, such that $G[V \setminus S]$ is a disjoint union of stars. For a graph G, the vertex deletion distance to disjoint union of stars $\mathrm{dts}(G)$ is the size of a smallest modulator set to a disjoint union of stars for G.

We show W[1]-hardness for Modularity parameterized by $\mathrm{dts}(G)$ by presenting a reduction from Unary Bin Packing defined as follows.

Unary Bin Packing
Input: A number of bins r, a capacity of a single bin k, and a multi-set of integers $A = \{a_1, \dots, a_n\}$ such that $\sum_{a \in A} a = rk$ and r and k are encoded in unary.
Question: Is there a surjective mapping $\alpha : A \to [r]$ such that for every $j \in [r]$ we have $\sum_{a \in \alpha^{-1}(j)} a = k$?

Unary Bin Packing is W[1]-hard for the parameter number of bins r [13]. Our reduction is an adaption of a parameterized reduction for showing the hardness of Equitable Connected Partition parameterized by the vertex deletion distance to various graph classes [2]. A main difficulty that needs to be overcome for our proof is that the sizes of the different gadgets need to be carefully balanced to achieve that a clustering corresponds to a bin packing and that a size-balanced clustering achieves the optimal modularity. In our proof, we consider the decision variant of Modularity where we ask if there is a clustering $\mathcal{C}$ for G with a modularity score $q(\mathcal{C})$ (or equivalently $\widetilde{q}(\mathcal{C})$) of at least some threshold value q^*.

Construction: Let $I = (A = \{a_1, \dots a_n\}, r, k_0)$ be an instance of Unary Bin Packing. Let $k = k_0 \cdot r^2 \cdot n^2$ and $a_i^* = a_i \cdot r^2 \cdot n^2$. Clearly, $\sum_{i=1}^{n} a_i^* = r \cdot k$. Note that the instance $I^* = (A^* = \{a_1^*, \dots, a_n^*\}, r, k)$ of Unary Bin Packing is equivalent to I, since each item and bin size is scaled by the same factor $r^2 \cdot n^2$. We construct an instance $I' = (G, q^*)$ of Modularity that is equivalent to I^* as follows. Let G be an initially empty graph. For every number $a_i^* \in A^*$, we create an item gadget S_i which is a star with $a_i^* - 1$ leaf vertices and star center vertex c_i. Next, we create r bin gadgets $B_1, \dots, B_r$. Each of these gadgets B_j consists of a star with $p := 5r^2k^2n^2$ leafs and a star center vertex b_j. We add an edge between every center vertex b_j of a bin gadget and every center vertex c_i of an item gadget. Finally, we add $x := 8pk = 40r^2k^3n^2$ isolated edges $e_1, \dots, e_x$. Since $\sum_{i=1}^{n} a_i^* = r \cdot k$, the constructed graph G has $m := rn + rp + (rk - n) + x$ edges. This concludes the construction, except for the concrete modularity threshold q^* whose definition is deferred to the long version of this work.

Observe that after deleting all star centers b_j of bin gadgets B_j for $j \in [r]$ each connected component of the resulting graph $G' = G - (\bigcup_{j \in [r]} \{b_j\})$ is either an isolated edge e_ℓ, an item gadget S_i or an isolated vertex that was a leaf vertex of a bin gadget B_j, all of which are stars. Thus $\mathrm{dts}(G) \leq r$ where r is the number of bins for I^*. Since Unary Bin Packing is W[1]-hard for the number of bins, it thus remains to show the correctness of the construction.

First, observe that, by Lemmas 2.1 and 2.2, in every optimal clustering the vertices of a star S_i belong to the same cluster. The same is true for a star B_j. Moreover, each of the isolated edges e_ℓ forms a separate cluster and we denote $\mathcal{E} := \{e_\ell \mid \ell \in [x]\}$. Thus from here on out we can assume that an optimal clustering $\mathcal{C}$ for G has the form $\mathcal{C} = \{C_1, \dots, C_t\} \cup \mathcal{E}$, where C_i contains $r_i \in [0, r]$ bin gadgets $B_1^i, \dots, B_{r_i}^i$ as well as $n_i \in [0, n]$ items gadgets, where the item gadgets have s_i leafs in total. For the value of $\widetilde{q}(\mathcal{C})$ we thus get

$$\widetilde{q}(\mathcal{C}) = 4m \left(\sum_{i=1}^{t} |E(C_i)| \right) - \sum_{i=1}^{t} \mathrm{vol}(C_i)^2 + \widetilde{q}(\mathcal{E}) \tag{7}$$

$$= 4m \left(\sum_{i=1}^{t} r_i p + r_i n_i + s_i \right) - \sum_{i=1}^{t} (2r_i p + r_i n + r n_i + 2s_i)^2 + \widetilde{q}(\mathcal{E}), \tag{8}$$

where $\widetilde{q}(\mathcal{E})$ is the contribution of the partial clustering $\mathcal{E}$ to the total modularity score.

The idea of the reduction is as follows. A modularity score for G of at least q^* can only be achieved by a clustering where each cluster (that is not an isolated edge e_ℓ) contains exactly one bin gadget and some item gadgets with total number of vertices k. Such a clustering corresponds to a partition of the items in A into r bins of size k_0, scaled by the factor $r^2 \cdot n^2$. The values for p and x as well as the scaling factor $r^2 \cdot n^2$ for the items and bin sizes are chosen accordingly.

▶ **Theorem 5.1** (*). *Modularity is W[1]-hard when parameterized by the vertex deletion distance to disjoint union of stars* dts.

6 Conclusion

We provided an FPT-algorithm for Modularity parameterized by a classic graph parameter, the max leaf number. Clearly, improvements of the running time for the max leaf number parameterization and FPT-algorithms for smaller structural parameters are desirable. In terms of running time improvements, it would also be interesting to reconsider and improve the FPT-algorithm for Modularity parameterized by the vertex cover number of G [16]. A particularly interesting question is whether one can replace the quadratic programming part for the vertex cover parameterization by a purely combinatorial algorithm or by an ILP formulation. The W[1]-hardness for the parameterization by distance to stars underlines once more the algorithmic difficulty of the problem. One approach that is not ruled out by our reduction would be to combine parameterizations by vertex deletion distance to tractable graph classes with other parameterizations, for example the maximum degree of the input graph. Another approach could be to consider FPT-approximation algorithms for Modularity with structural parameterizations.

References

1 James P Bagrow. Communities and bottlenecks: Trees and treelike networks have high modularity. *Physical Review E*, 85(6):066118, 2012. doi:10.1103/PhysRevE.85.066118.

2 Václav Blazej, Dusan Knop, Jan Pokorný, and Simon Schierreich. Equitable connected partition and structural parameters revisited: N-fold beats lenstra. *CoRR*, abs/2404.18968, 2024. doi:10.48550/arXiv.2404.18968.

3 Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, 2008. doi:10.1088/1742-5468/2008/10/P10008.

4 Ulrik Brandes, Daniel Delling, Marco Gaertler, Robert Görke, Martin Hoefer, Zoran Nikoloski, and Dorothea Wagner. On modularity clustering. *IEEE Transactions on Knowledge and Data Engineering*, 20(2):172–188, 2008. doi:10.1109/TKDE.2007.190689.

5 Aaron Clauset, Mark EJ Newman, and Cristopher Moore. Finding community structure in very large networks. *Physical Review E*, 70(6):066111, 2004. doi:10.1103/PhysRevE.70.066111.

6 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015. doi:10.1007/978-3-319-21275-3.

7 Bhaskar DasGupta and Devendra Desai. On the complexity of Newman's community finding approach for biological and social networks. *Journal of Computer and System Sciences*, 79(1):50–67, 2013. doi:10.1016/J.JCSS.2012.04.003.

8 Fabien de Montgolfier, Mauricio Soto, and Laurent Viennot. Asymptotic modularity of some graph classes. In *Proceedings of the 22nd International Symposium on Algorithms and Computation (ISAAC '11)*, volume 7074 of *Lecture Notes in Computer Science*, pages 435–444. Springer, 2011. doi:10.1007/978-3-642-25591-5_45.

9 David Eppstein. Metric dimension parameterized by max leaf number. *Journal of Graph Algorithms and Applications*, 19(1):313–323, 2015. doi:10.7155/JGAA.00360.

10 Michael R. Fellows, Daniel Lokshtanov, Neeldhara Misra, Matthias Mnich, Frances A. Rosamond, and Saket Saurabh. The complexity ecology of parameters: An illustration using bounded max leaf number. *Theory of Computing Systems*, 45(4):822–848, 2009. doi:10.1007/S00224-009-9167-9.

11 Erez Hartuv and Ron Shamir. A clustering algorithm based on graph connectivity. *Information Processing Letters*, 76(4-6):175–181, 2000. doi:10.1016/S0020-0190(00)00142-3.

12 Falk Hüffner, Christian Komusiewicz, Adrian Liebtrau, and Rolf Niedermeier. Partitioning biological networks into highly connected clusters with maximum edge coverage. *IEEE ACM Transactions on Computational Biology and Bioinformatics*, 11(3):455–467, 2014. doi:10.1109/TCBB.2013.177.

13 Klaus Jansen, Stefan Kratsch, Dániel Marx, and Ildikó Schlotter. Bin packing with fixed number of bins revisited. *Journal of Computer and System Sciences*, 79(1):39–49, 2013. doi:10.1016/J.JCSS.2012.04.004.

14 Colin McDiarmid and Fiona Skerman. Modularity in random regular graphs and lattices. *Electronic Notes in Discrete Mathematics*, 43:431–437, 2013. doi:10.1016/j.endm.2013.07.063.

15 Colin McDiarmid and Fiona Skerman. Modularity of regular and treelike graphs. *Journal of Complex Networks*, 6(4):596–619, 2018. doi:10.1093/comnet/cnx046.

16 Kitty Meeks and Fiona Skerman. The parameterised complexity of computing the maximum modularity of a graph. *Algorithmica*, 82(8):2174–2199, 2020. doi:10.1007/s00453-019-00649-7.

17 M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69:026113, February 2004. doi:10.1103/PhysRevE.69.026113.

18 Mark Newman. *Networks*. Oxford university press, 2018.

19 Ron Shamir, Roded Sharan, and Dekel Tsur. Cluster graph modification problems. *Discrete Applied Mathematics*, 144(1-2):173–182, 2004. doi:10.1016/j.dam.2004.01.007.

20 Fiona Skerman. *Modularity of networks*. PhD thesis, University of Oxford, 2015.