FACULTY
OF INFORMATION
TECHNOLOGY
CTU IN PRAGUE

Assignment of bachelor's thesis

Title:	Parameterized Algorithms for the Truncated Metric Dimension problem
Student:	Jiří Jirásek
Supervisor:	RNDr. Ondřej Suchý, Ph.D.
Study program:	Informatics
Branch / specialization:	Computer Science
Department:	Department of Theoretical Computer Science
Validity:	until the end of summer semester 2023/2024

Instructions

Get familiar with the Metric Dimension problem of graphs and its Truncated variant.
Get familiar with the basic notions and ideas of Parameterized Complexity.
Survey known results about parameterized complexity of Metric Dimension, especially parameterized algorithms and get familiar with the most important of them.

Inspired by these algorithms, develop parameterized algorithms for Truncated Metric Dimension or find major obstacles in developing such algorithms.

After consulting with the supervisor select one of the algorithms and implement it in a suitable language.

Test the resulting program on a suitable data, evaluate its performance.

Electronically approved by doc. Ing. Jan Janoušek, Ph.D. on 2 February 2023 in Prague.

Bachelor's thesis

PARAMETERIZED ALGORITHMS FOR THE TRUNCATED METRIC DIMENSION PROBLEM

Jiří Jirásek

Faculty of Information Technology
Department of Theoretical Computer Science
Supervisor: RNDr. Ondřej Suchý, Ph.D.
January 11, 2024

Citation of this thesis: Jirásek Jiří. *Parameterized Algorithms for the Truncated Metric Dimension problem.* Bachelor's thesis. Czech Technical University in Prague, Faculty of Information Technology, 2024.

Contents

List of Figures

List of Tables

First, I would especially like to express my gratitude to my supervisor RNDr. Ondřej Suchý, Ph.D. for his extraordinary amount of patience, invaluable insights and for all the time and effort he put towards helping me with the thesis. Second, I would like to thank Arara∼, Chipex and Veronika for their help and for supporting me throughout, not only writing this thesis, but all my studies.

Declaration

I hereby declare that the presented thesis is my own work and that I have cited all sources of information in accordance with the Guideline for adhering to ethical principles when elaborating an academic final thesis. I acknowledge that my thesis is subject to the rights and obligations stipulated by the Act No. 121/2000 Coll., the Copyright Act, as amended, in particular that the Czech Technical University in Prague has the right to conclude a license agreement on the utilization of this thesis as a school work under the provisions of Article 60 (1) of the Act.

In Prague on January 11, 2024 ....

Abstract

This thesis is about FPT algorithms that solve the Truncated Metric Dimension problem. First, two already known algorithms solving the metric dimension problem are described.
Then we present two counterexamples to the algorithm bounded by modular width, showing the algorithm does not work correctly. Following that, we propose a possible solution, that might fix the described issues. The algorithm is then implemented and tested.
It is also argued as to why the algorithm bounded by tree-length and max-degree is not suitable to solve the truncated metric dimension problem.

Keywords Metric Dimension, Truncated Metric Dimension, Resolving set, Parameterized algorithms, Complexity

Abstrakt

Tato práce se zabývá FPT algoritmy řešící problém zkrácené metrické dimenze. Nejdříve představíme dva již známé algoritmy řešící problém metrické dimenze.
Poté představíme dva jednoduché protipříklady algoritmu, jehož časová složitost je vázaná šířkou modulu, čímž ukážeme že nepracuje korektně. Dále navrhneme možnou úpravu, jež by mohla řešit popsané problémy s výpočtem. Algoritmus je dále implementován a testován.
Ukážeme také proč algoritmus, jehož časová složitost je vázaná délkou stromu a největším stupněm vrcholu, není vhodný pro úpravu aby řešil problém zkrácené metrické dimenze.

Klíčová slova Metrická dimenze, zkrácená metrická dimenze, rozlišující množina, parametrizované algoritmy, složitost

List of Acronyms

MD	METRIC DIMENSION
k-MD	TRUNCATED METRIC DIMENSION
FPT	FIXED-PARAMETER TRACTABLE
SAT	BOOLEAN SATISFIABILITY
NP	NONDETERMINISTIC POLYNOMIAL TIME

Introduction

The METRIC DIMENSION (MD) problem, is an old problem, that asks, given a graph and a number k, if there is a (resolving) set of k vertices, such that every vertex can be uniquely identified by its distance from the vertices in the set. The TRUNCATED METRIC DIMENSION (k-MD) problem poses the same question with the simple modification that we only consider vertices from the set, whose distance is at maximum k from the vertex we want to identify.

Identifying such vertices in a graph may be useful when we can consider robots which are moving from a node to a node in a network. We assume that the robots can communicate with a set of landmarks (subset of nodes) which provide them the distance to the landmarks in order to facilitate the navigation. In this sense, the position of each robot is uniquely determined by the distance to the landmarks. [1] We may want to only consider vertices that are no further apart than some distance, because the communication between a robot and some landmark can get more costly, or even impossible as the distance increases.

Our contributions The goal of this thesis is to make use of the existing parameterized algorithms for the METRIC DIMENSION problem with respect to various structural parameters and, if possible alter them in such a way that they then compute the solution of the TRUNCATED METRIC DIMENSION problem. In this thesis we will focus on the two algorithms proposed in an article by Belmonte et al.[2]. For the algorithm for graphs bounded by tree-length and max-degree we show that it is unfit for such alteration and for the algorithm bounded by modular-width we show a counterexample to its correctness.

Outline In the Sections 1, 2 and 3, we provide necessary definitions and introduce the notation that will be used throughout this thesis. In Sections 4 and 5 we discuss already known results about both the METRIC DIMENSION and TRUNCATED METRIC DIMENSION. In Chapter 6 and 7, we introduce the two known algorithms, one parameterized by tree-length and max-degree and the other parameterized by modular-width, both invented by Belmonte et al.[2] and at the end of the Chapter 7 we show tow counterexamples to the algorithm. In Chapter 8 and 9 we discuss the viability of altering both of the algorithms. Finally in Chapters 10 and 11 we describe our implementation and testing of the algorithm bounded by modular-width.

Preliminaries

1 Graph Theory

First, we shall start with a definition of a graph.

▶ **Definition 1.1** (Graph, Inspired by [3])**.** All graphs considered for the purposes of this thesis are undirected, unweighted and simple, i.e. without loops or multiple edges. A *graph* $G = (V, E)$ consists of set V and E.

- V is a set of *vertices*, sometimes referred to as $V(G)$, when it is not obvious to which graph we refer.
- E is a set of *edges*, sometimes also denoted $E(G)$.
- Edge is a set that consists of exactly two vertices, which are called *endpoints*. An edge *joins* its endpoints.
- A vertex v is adjacent to a vertex u if $\{u, v\} \in E$.
- Adjacent vertices may be called *neighbours*, the set of all neighbours of vertex v is the (*open*) *neighbourhood* and denoted $N(v)$.
- The *closed* neighbourhood of vertex v is $N[v] = N(v) \cup \{v\}$.
- For a positive integer r let $N^r_G[v] = \{u \in V \mid \mathrm{dist}_G(u, v) \leq r\}$ be the set of vertices at distance at most r from v.
- An edge is *incident* to vertex v, if v is one of its endpoints.
- The *degree* of a vertex is number of its neighbours.
- The *maximum degree* of a graph is the maximum over the degrees of all the vertices.

By $G - U$ we denote the graph obtained by removal of all the vertices of U. $G[U]$ denotes graph *induced* by the set $U \subseteq V$, meaning $G[U] = G - (G \setminus U)$.

We will also use n and m to denote the number of verticies and edges respectively.

Following the definition of a graph, we define path.

▶ **Definition 1.2** (Path, Inspired by [3])**.** A *path* in a graph G is an alternating sequence of vertices and edges $P = v_0, e_1, v_1, e_2, ..., e_n, v_n$, where for each $j \in 1, 2, ..., n$, and v_{j-1} and v_j are endpoints of e_j, and no vertex is repeated in the sequence.

- The vertex v_0 is the *initial vertex*.

- The vertex v_n is the *terminal vertex.*
- A u-v-path is a path with initial vertex u and terminal vertex v.

Since we reference trees in this thesis, we will define a tree structure. For that we will also need a definition of a cycle and a definiton of a connected graph.

▶ **Definition 1.3** (Cycle [4])**.** A graph $G = (V, E)$ is a *cycle*, if $G = (\{1, ..., n\}, \{\{i, i+1\} \mid i \in \{1, ..., n-1\}\} \cup \{\{1, n\}\})$, for $n > 3$.

▶ **Definition 1.4** (Connected graph [5])**.** A graph $G = (V, E)$ is *connected* if for each two distinct vertices $u, v \in V(G)$ there is a u-v-path in G.

▶ **Definition 1.5** (Tree [5])**.** A graph $G = (V, E)$ is a *tree* if the graph is connected and does not have a cycle as a sub-graph. We call a vertex $v \in V(G)$ a *leaf* if $\deg_G(v) = 1$.

The problems we deal with are closely tied to distance in a graph, we shall define a distance in a graph too.

▶ **Definition 1.6** (Distance, l-truncated distance, Inspired by [6])**.** Distance denoted $\operatorname{dist}_G(u, v)$ between two vertices u and v in the graph G is the number of edges in a shortest u-v-path in the graph G.

- Let $\operatorname{dist}_{G,l}(u, v) = \min(\operatorname{dist}_G(u, v), l + 1)$ denote *l-truncated* distance.
- For a vertex $v \in V$ and a set $U \subseteq V$, let $\operatorname{dist}_G(v, U) = \min\{\operatorname{dist}_G(v, u) \mid u \in U\}$ be a minimal distance from a vertex v to any of the vertices from U.

▶ **Definition 1.7** (Diameter, Inspired by [2])**.** For a set $U \subseteq V$ of a graph $G = (V, E)$, we define its *diameter* as $diam_G(U) = \max\{\operatorname{dist}_G(u, v) \mid u, v \in U\}$. Then specifically we denote the diameter of a graph as $\operatorname{diam}(G) = \operatorname{diam}_G(V)$.

For the proof of correctness of an algorithm bounded by modular-width we will need to define a universal vertex.

▶ **Definition 1.8** (Universal vertex [2])**.** A vertex $v \in V$ is *universal* if $N_G(v) = V \setminus \{v\}$.

▶ **Definition 1.9** (Disjoint union and join of graphs [2])**.** For two graphs G_1, G_2, the disjoint union of G_1 and G_2 is the graph G that has $V(G_1) \cup V(G_2)$ as its vertices and $E(G_1) \cup E(G_2)$ as its edges.

A join of graphs G_1 and G_2 is the graph G that has $V(G_1) \cup V(G_2)$ as its vertices and $E(G_1) \cup E(G_2) \cup \{uv \mid u \in V_1, v \in V_2\}$ as its edges.

In the following we specify the structural parameters *modular-width* and *tree-length*, which we will use as the parameters for the algorithms.

▶ **Definition 1.10** (Modular-width [2])**.** A set $X \subseteq V(G)$ is a module of a graph G if for any $v \in V(G) \setminus X$, either $X \subseteq N_G(v)$ or $X \cap N_G(v) = \emptyset$. We shall define modular-width using a recursive definition as it is more suitable for our purpose. The modular-width of a graph G is at most t if one of the following holds:

1. G has one vertex;
2. G is disjoint union of two graphs of modular-width at most t;
3. G is a join of two graphs of modular-width at most t;
4. $V(G)$ can be partitioned into $s \leq t$ modules $X_1, ..., X_s$ such that modular-width $\operatorname{mw}(G[X_i]) \leq t$ for $i \in \{1, ..., s\}$.

▶ **Definition 1.11** (Tree decomposition [2]). A tree decomposition of a graph G is a pair (X, T) where T is a tree and $X = \{X_i \mid i \in V(T)\}$ is a collection of bags (subsets of a $V(G)$) such that:

1. $\cup_{i \in V(T)} X_i = V(G)$ has one vertex;
2. for each edge $uv \in E(G), x, y \in X_i$ for some $i \in V(T)$;
3. for each vertex $x \in V(G)$ the set $\{i \mid x \in X_i\}$ induces a connected sub-tree of T.

The width of a tree decomposition $(\{X_i \mid i \in V(T)\}, T)$ is $\max_{i \in V(T)} |X_i| - 1$. The length of a tree decomposition $(\{X_i \mid i \in V(T)\}, T)$ is $\max_{i \in V(T)} diam(X_i)$. The tree-length of a graph G denoted as $tl(G)$ is a minimum length over all tree decompositions of G.

▶ **Definition 1.12** (Nice tree decomposition [2]). We call tree decomposition (X, T) of a graph G with $X = \{X_i \mid i \in V(T)\}$ nice if T is a rooted binary tree such that nodes of T are of four types

1. a leaf node is a leaf of T and $|X_i| = 1$;
2. an introduce node i has one child i' with $X_i = X_{i'} \cup \{v\}$ for some vertex $v \in V(G)X_{i'}$;
3. a forget node i has one child i' with $X_i = X_{i'} \setminus \{v\}$ for some vertex $v \in X_{i'}$;
4. a join node i has two children i' and i'' with $X_i = X_{i'} = X_{i''}$ such that the subrees of T rooted in i' and i'' have at least one forget node each.

Modular width can be computed in linear time by the algorithm of Tedder et al. [7] This is not the case for tree-length as it has been proved that to decide whether $tl(G) < l$ for a graph G and $l \geq 2$ is a NP-complete problem [8]. However we can approximate l within a factor of 3, by utilizing the techniques developed by Dourisbourne and Gavoille [8]. It is also possible to show that nice tree decomposition of a graph can be computed from any tree decomposition in polynomial time [9]. In the original article, Belmonte et al. [2] reference the claim posed by Kloks [9], that a nice tree decomposition can be computed in a linear time from a tree decomposition. This however is not correct. Counterexample can be found in the book by Cygan et al.[10]. We can however find a nice tree decomposition in a polynomial time, which suffices for our use. Moreover the length l and width w of the nice tree decomposition shall be equal to the length and width of the original tree decomposition. Size of such tree is $O(wn)$ [2]. It is also possible to obtain such nice tree decomposition with some $v \in V(G)$ as a unique vertex in the root bag.

2 Parameterized Problems

This thesis examines problems with regard to some structural properties. We call such problems parameterized.

▶ **Definition 2.1** (Parameterized problem, Cygan et al. [10]). Parameterized problem is a language $L \subseteq \Sigma \times N$, where Σ is a fixed finite alphabet. For an instance $(x, k) \in \Sigma^* \times N, k$ is called parameter.

▶ **Definition 2.2** (FPT, Cygan et al. [10]). A parameterized problem $L \subseteq \Sigma^* \times N$ is called *fixed-parameter tracable* (FPT) if there exists an algorithm A, called *fixed-parameter tracable* algorithm, a computable functions $f : N \to N$, and a constant c such that, given $(x, k) \in \Sigma^* \times N$, the algorithm A correctly decides whether $(x, k) \in L$ in time bounded by $f(k) \cdot |(x, k)|^c$. The complexity class containing all fixed-parameter tractable problems is called FPT.

It would also be appropriate to introduce the 3-SAT problem, however this would require many definitions that are not fit for this thesis. The definition can be found in the book by Cygan et al.[10] along with the proof that 3-SAT is NP-complete.

3 Definitions of the Problems

Now we shall define the METRIC DIMENSION and the TRUNCATED METRIC DIMENSION problems. For simplicity we only define the decision version of the problems, however all the algorithms described in this thesis can be converted to find the resolving sets at no further cost of running time.

▶ **Definition 3.1** (METRIC DIMENSION [2])**.** Let $G = (V, E)$ be a graph. For two distinct vertices $u, v \in V$, let $R(x, y) = \{z \in V \mid \operatorname{dist}(x, z) \neq \operatorname{dist}(y, z)\}$. A subset $S \subseteq V(G)$ is a *resolving* set of G if $S \cap R(x, y) \geq 1$ for any pair of distinct vertices x and y in G. The metric dimension of G denoted by $\operatorname{md}(G)$ is the minimum cardinality over all resolving sets of G. METRIC DIMENSION asks if $\operatorname{md}(G) \leq d$, for a given positive integer d.

▶ **Definition 3.2** (TRUNCATED METRIC DIMENSION [6])**.** Let $G = (V, E)$ be a graph and k a positive integer. For two distinct vertices $u, v \in V$, let $R_k(x, y) = \{z \in V \mid \operatorname{dist}_k(x, z) \neq \operatorname{dist}_k(y, z)\}$. A subset $S \subseteq V(G)$ is a *k-truncated resolving* set of G if $S \cap R_k(x, y) \geq 1$ for any pair of distinct vertices x and y in G. The truncated metric dimension of G denoted by $\operatorname{md}_k(G)$ is the minimum cardinality over all resolving sets of G. TRUNCATED METRIC DIMENSION asks if $\operatorname{md}_k(G) \leq d$, for given positive integers d and k.

Known Results

4 Metric Dimension

The notion of resolving sets was first independently introduced by Slater [11] and Harary with Melter [12] as a way of uniquely identifying all the vertices in a graph. Khuller et al. [13] have in their paper on the topic shown that it is NP-hard to find or decide if there is such a set on general graphs and that it can be solved in linear time on trees.

As a result of this finding many algorithms that approximate the metric dimension have been developed. Most notably algorithms that make use of the generic algorithm [14] and a variable neighborhood search [15]. Even though these algorithms usually find small resolving sets, there are no approximation guarantees that would bound the distance between the approximation and some optimal resolving set.

In 2012 such algorithm was developed by Hauptmann et al. [16]. For a graph $G = (V, E)$ it guarantees the approximation ratio of $1 + (1 + o(1) \cdot \ln(|V(G)|))$. The running time complexity of this algorithm is $O\left(|V(G)|^3\right)$.

It is also important to mention that efficient exact algorithms, bounds and formulae have been discovered for variety of graphs.

5 Truncated Metric Dimension

In 2021 the notion of truncated metric dimension has been established by Frongillo et al.[6]. The motivation for the restriction on the distance was one, due to the cost of long distance communication in a network and two, reducing dependency on random variables in identifying the source of an infection in an epidemic [6, 17, 18].

Just like for metric dimension, it has been shown that this problem is NP-hard by reduction from 3-SAT. This should be obvious since we can set the parameter k to be strictly higher then the diameter of a given graph, for example as the number of vertices of the graph, and we get the exact definition of the non-truncated metric dimension. This means that this can be seen as somewhat of a generalization of the problem.

Bounds and formulae have also been discovered by Frongillo et al.[6] for specific types of graphs. For, trees Frongillo et al.[6] have developed an exact algorithm that runs in liner time.

Algorithms For Metric Dimension

6 Max-degree and Tree-length

In this section, we present an FPT algorithm for the METRIC DIMENSION *parameterized by the max-degree and the tree-length, that was developed by Belmonte et al. [2]*

By the following lemma we bound the tree-width by the tree-length and the max-degree.

▶ **Lemma 6.1** ([2])**.** Let G be a connected graph with $\Delta(G) = \Delta$ and let (X,T) be a tree decomposition of G with the length at most l. Then the width of (X,T) is at most $w(\Delta,l) = \Delta(\Delta-1)^{(l-1)}$.

By the following lemma we bound the number of bags of (X,T) a particular vertex can appear in. Belmonte et al. [2] then use this to prove Lemma 6.3.

▶ **Lemma 6.2** ([2])**.** Let G be a connected graph with $\Delta(G) = \Delta$, and let (X,T), where $X = \{X_i \mid i \in V(T)\}$, be a nice tree decomposition of G of length at most l. Furthermore, let P be a path in T such that for some vertex $z \in V(G), z \in X_i$ for every $i \in V(P)$. Then $|V(P)| \leq \alpha(\Delta,l) = 2\left(\Delta^l(\Delta+2)+4\right)$.

▶ **Lemma 6.3** ([2])**.** Let G be a connected graph with max-degree $\Delta(G) = \Delta$ and let (X,T), where $X = \{X_i \mid i \in V(T)\}$, be a nice tree decomposition of G of length at most l. Then for every $i,j \in V(T)$ and any $x \in X_i, y \in X_j$ the following holds:

$$\mathrm{dist}_T(i,j) \leq \alpha(\Delta,l)(\mathrm{dist}_G(x,y)+1)-1.$$

The next lemma essentially approximates distance between pairs of vertices in the graph by factor only depending on Δ and l. This lemma will be further explored in the next chapter and so we provide a proof.

▶ **Lemma 6.4** (Locality Lemma [2])**.** Let (X,T), where $X = \{X_i \mid i \in V(T)\}$, be a nice tree decomposition of G of length at most l such that T is rooted in $r, X_r = \{u\}$. Let $\Delta = \Delta(G)$ be the max-degree of G and let $s = \alpha(\Delta,l)(2l+1)$. Then the following holds:

1. If $i \in V(G)$ is an introduce node with the child i' and v is the unique vertex of $X_i \setminus X_{i'}$ then for any $x \in V(G_j)$ for a node $j \in V(T_i)$ such that $\mathrm{dist}_T(i,j) \geq s$, u resolves v and x.
2. If $i \in V(G)$ is a join node with the children i', i'' and $x \in V(G_j) \setminus X_j$ for $j \in T_{i'}$ such that $dist_T(i',j) \geq s-1$ and $y \in V(G_{i''}) \setminus X_{i''}$ then u or an arbitrary vertex $v \in (V(G_j) \setminus X_j)$ resolves x and y.

Proof.

1. Consider $x \in V(G_j)$ for some $j \in V(T_{i'})$ such that $\mathrm{dist}_T(i', j') \geq s$. As either $u \in X_i$ or u is separated from x by X_i,
$$\mathrm{dist}_G(u, x) = \min \{\mathrm{dist}_G(u, y) + \mathrm{dist}_G(y, z) + \mathrm{dist}_G(z, x) \mid y \in X_i, z \in X_j\}.$$
Let $y \in X_i$ and $z \in X_j$ be vertices such that $\mathrm{dist}_G(u, x) = \mathrm{dist}_G(u, y) + \mathrm{dist}_G(y, z) + \mathrm{dist}_G(z, x)$. Then by Lemma 6.3,
$$\mathrm{dist}_G(u, x) \geq \mathrm{dist}_G(u, y) + \mathrm{dist}_G(y, z) \geq \mathrm{dist}_G(u, y) + \frac{s+1}{\alpha(\Delta, l)} - 1.$$
Because $v \in X_i$ and $\mathrm{diam}_G(X_i) \leq l$,
$$\mathrm{dist}_G(u, v) \leq \mathrm{dist}_G(u, y) + \mathrm{dist}_G(y, v) \leq \mathrm{dist}_G(u, y) + l.$$
Because $s = \alpha(\Delta, l)(2l + 1)$, we obtain that $\mathrm{dist}_G(u, v) < \mathrm{dist}_G(u, x)$, completing the proof of the first statement.

2. Let $x \in V(G_j)$ for $j \in T_{i'}$ such that $\mathrm{dist}_T(i', j) = s - 1$, and let $y \in V(G_{i''}) \setminus X_{i''}$. Assume also that $v \in V(G_j) \setminus X_j$. Suppose that u does not resolve x and y. It means that $\mathrm{dist}_G(u, x) = \mathrm{dist}_G(u, y)$. Because either $u \in X_i$ or u and $\{x, y\}$ are separated by X_i, there are $x', y' \in X_i$ such that $\mathrm{dist}_G(u, x) = \mathrm{dist}_G(u, x') + \mathrm{dist}_G(x', x)$ and $\mathrm{dist}_G(u, y) = \mathrm{dist}_G(u, y') + \mathrm{dist}_G(y', y)$. As $\mathrm{dist}_G(u, x) = \mathrm{dist}_G(u, y)$ and $\mathrm{diam}_G(X_i) \leq l$,
$$\mathrm{dist}_G(x', x) - \mathrm{dist}_G(y', y) = \mathrm{dist}_G(u, y') - \mathrm{dist}_G(u, x') \leq l.$$
Notice that $\mathrm{dist}_G(x, X_i) \leq \mathrm{dist}_G(x, x')$ and $\mathrm{dist}_G(y, X_i) \geq \mathrm{dist}_G(y, y') - l$, because $\mathrm{diam}_G(X_i) \leq l$. Hence, $\mathrm{dist}_G(x, X_i) - \mathrm{dist}_G(y, X_i) \leq 2l$. There are $z, z' \in X_j$ such that $\mathrm{dist}_G(x, X_i) = \mathrm{dist}_G(x, z) + \mathrm{dist}_G(z, X_i)$ and $\mathrm{dist}_G(v, X_i) = \mathrm{dist}_G(z', X_i)$. Because $\mathrm{diam}_G(X_j) \leq l$, $\mathrm{dist}_G(v, z') + l$ and $\mathrm{dist}_G(z, X_i) \leq \mathrm{dist}_G(z', X_i) + l$. Hence
$$\mathrm{dist}_G(v, z) + \mathrm{dist}_G(z, X_i) \leq \mathrm{dist}_G(v, z') + \mathrm{dist}_G(z', X_i) + 2l \leq \mathrm{dist}_G(v, X_i) + 2l.$$
Since X_i separates v and y,
$$\begin{aligned}\mathrm{dist}_G(v, y) &\geq \mathrm{dist}(v, X_i) + \mathrm{dist}(y, X_i)\\ &\geq \mathrm{dist}(v, z) + \mathrm{dist}(z, X_i) - 2l + \mathrm{dist}(y, X_i)\\ &\geq \mathrm{dist}(v, z) + \mathrm{dist}(z, X_i) - 2l + \mathrm{dist}(x, X_i) - 2l\\ &\geq \mathrm{dist}(v, z) + 2 \cdot \mathrm{dist}(z, X_i) + \mathrm{dist}(y, z) - 4l\end{aligned}$$
Clearly, $\mathrm{dist}(v, x) \leq \mathrm{dist}(x, z) + \mathrm{dist}(v, z)$. Hence,
$$\begin{aligned}\mathrm{dist}(v, y) - \mathrm{dist}(v, x) &\geq (\mathrm{dist}(v, z) + 2\mathrm{dist}(z, X_i) + \mathrm{dist}(x, z) - 4l)\\ &\quad - (\mathrm{dist}(x, z) + \mathrm{dist}(v, z))\\ &\geq 2 \cdot \mathrm{dist}(z, X_i) - 4l.\end{aligned}$$
All that remains is to observe that $\mathrm{dist}(z, X_i) \geq \frac{x+1}{\alpha(\Delta,l)} - 1 > 2l$. With that we obtain that $\mathrm{dist}(v, y) - \mathrm{dist}(v, x) > 0$, this means that v resolves x and y.

◀

With the conclusion of this proof, we have presented the necessary structural properties of graphs bounded by tree-length and max-degree. In the next part we introduce projection and resolving sets.

▶ **Definition 6.5** (Projection and resolving set [2])**.** Let $X \subseteq V(G)$, and let d be a positive integer such that $\text{diam}_G(X) \leq d$. For a vertex $v \in V(G)$, we say that $\mathcal{P}r_{v,d}(X) = (X_0, ..., X_d)$, where $X_i = \{x \in X \mid \text{dist}_G(v, x) = \text{dist}_G(v, X) + i\}$ is the *projection* of v on X. Notice that $(X_0, ..., X_d)$ form an ordered partition of X, because $\text{diam}_G(X) \leq d$. Some sets could be empty. For a set $U \subseteq V(G)$, the set $\mathcal{P}r_{U,d}(X) = \{\mathcal{P}r_{v,d} \mid v \in U\}$. It can happen that $\mathcal{P}r_{v,d}(X) = \mathcal{P}r_{u,d}(X)$ for $u, v \in U$, but as $\mathcal{P}r_{U,d}(X)$ is a set, it contains only one copy of $\mathcal{P}r_{v,d}(X)$.

The algorithm uses the following properties of separators of bounded diameter. For the next lemma and two definitions, let X be the separator of connected graph G such that $\text{diam}_G(X) \leq d$, and let V_1, V_2 be partition of the vertex set of $G - X$ such that no edge of G joins a vertex of V_1 with the vertex of V_2.

▶ **Lemma 6.6** ([2])**.** If for $u, v \in V_1$, $Pr_{u,d}(X) = Pr_{v,d}(X)$, then u resolves verticex $x, y \in V_2$ if and only if v resolves x, y. Moreover, for a given ordered partition $(X_0, ..., X_d)$ of X, it can be decided in polynomial time whether a vertex $v \in V_1$ with $Pr_{v,d}(X) = (X_0, ..., X_d)$ resolves x and y.

▶ **Definition 6.7** (Ordered partition [2])**.** Let $X' \subseteq X \cup V_2$ with $\text{diam}_G(X) \leq d$. We define the *ordered partition* $(X'_0, ..., X'_d)$ of X' as:

$$X'_i = \left\{ x \in X' \mid \min_{i \in 0,...,d} (i + \text{dist}_G(X_i, x)) = s + i \right\}, \text{where}$$

$$s = \min_{x \in X'} \min_{i \in \{0,...,d\}} (i + \text{dist}_G(X_i, x)).$$

This implies that if for $u, v \in V_1, Pr_{u,d}(X)$, then u resolves x and y. And since for any $x \in V_2$ we can compute $min_{i \in \{0,...d\}}(i + \text{dist}_G(X_i, x))$ by making use of the Dijkstra's algorithm if $(X_0, ..., X_d)$ is given, we obtain the second pair of the statement.

▶ **Definition 6.8** (d-cover [2])**.** We say that $(X_0, ..., X_d)$ is a $d-$*cover* of $(X'_0, ..., X'_d)$ with respect to V_1, and we say that $(X'_0, ..., X'_d)$ is $d-$*covered by* $(X_0, ..., X_d)$ with respect to V_1. We also say that a set $\mathcal{P}$ of ordered partitions $(X_0, ..., X_d)$ of X is a $d-$*cover* of a set $\mathcal{P}'$ of ordered partition $(X'_0, ..., X'_d)$ of X' with respect to V_1, if $\mathcal{P}'$ is the set of all ordered partitions of X' that are $d-$*covered* by the partitions of $\mathcal{P}$.

▶ **Lemma 6.9** ([2])**.** Let $X' \subseteq X \cup V_2$ with $\text{diam}_G(X) \leq d$. Let also $(X_0, ..., X_d)$ and $(X'_0, ..., X'_d)$ be ordered partitions of X and X' respectively such that $(X_0, ..., X_d)$ is a $d-$*cover* of $(X'_0, ..., X'_d)$ with respect to V_1. If $\mathcal{P}r_{v,d}(X) = (X_0, ..., X_d)$ for some $v \in V_1$, then $\mathcal{P}r_{v,d} = (X') = (X'_0, ..., X'_d)$.

▶ **Theorem 6.10** ([2])**.** METRIC DIMENSION is FPT when parameterized by $\Delta + tl$, where Δ is max-degree and tl is tree-length of the input graph.

The algorithm. [2] From now on, we assume that $u \in V(G)$ is given. Using the techniques of Kloks [9] we construct a nice tree decomposition from (X, T) of the same width and the length at most l such that the root bag only contains the vertex u. For simplification, we assume that (X, T) is such a decomposition and T is rooted in r. By Lemma 6.2, for any path P in T, any $z \in V(G)$ occurs in at most $\alpha(\Delta, l)$ bags of X_i for $i \in V(P)$.

Now we introduce the dynamic programming algorithm that checks the existence of a resolving set of size at most k that includes u.

Let $s = \alpha(\Delta, l)(2l + 1)$. For $i \in V(T)$, we define $Y_i = \cup_{j \in N^s_{T_i}[i]} X_j$ and

$$I_i = \{j \in V(T_i) \mid \text{dist}_G(i, j) = s\}.$$

Let also $I'_i = I_i \cup \{0\}$. For each $i \in V(T)$, the algorithm constructs the table of values of the function $w_i\left(Z, \left\{\mathcal{P}^j \mid j \in I'_i\right\}\right)$, where

1. $Z \subseteq Y_i$ and $|Z| \leq k$,
2. $\mathcal{P}^0$ is a set of ordered partitions $(Y_0, ..., Y_l)$ of X_i such that $\mathcal{P}r_{u,l}(X_i) \in \mathcal{P}^0$ if $u \notin X_i$,
3. For $j \in I_i$, $\mathcal{P}^j$ is a set of ordered partitions $(Y_0, ..., Y_l)$ of X_j, and $w_i\left(Z, \left\{\mathcal{P}^j \mid j \in I'_i\right\}\right)$ is the minimum cardinality of a set $W \subseteq V(G_i)$ such that
   a. For any two distinct $x, y \in V(G_i)$, there is a vertex $v \in W$ that resolves x and y or there is an ordered partition $(Y_0, ..., Y_l) \in \mathcal{P}^0$ of X_i such that a vertex $v \in V(G) \setminus V(G_i)$ with $\mathcal{P}r_{v,l}(X_i) = (Y_0, ..., Y_l)$ resolves x and y.
   b. $W \cap Y_i = Z$,
   c. For $j \in I_i$, $\mathcal{P}^j = \mathcal{P}r_{W \cap (V(G_j) \setminus X_j), l}(X_j)$.

If no such set W exists, then $w_i\left(Z, \left\{\mathcal{P}^j \mid j \in I'_i\right\}\right) = +\infty$.

It can be observed that the table with an entry $w_r\left(Z, \left\{\mathcal{P}^j \mid j \in I'_r\right\}\right) \leq k$ for the root node r is a necessary and sufficient condition for the existence of resolving set W of size at most k. Now we shall explain how we construct table for each node $i \in V(T)$.

Let $i \in V(T)$. We define $J_i = \{j \in V(T) \mid \mathrm{dist}_{T_i}(i, j) = s - 1\}$. For Z and $\left\{\mathcal{P}^j \mid j \in I_i\right\}$ satisfying 1 and 3,

$$\mathcal{R}\left(Z, \left\{\mathcal{P}^j \mid j \in I_i\right\}\right) = \left\{\mathcal{R}^j \mid j \in J_i\right\},$$

where $\mathcal{R}^j$ is a set of ordered partitions $(Y_0, ..., Y_l)$ of X_j, constructed as follows. Let $j \in J_i$.

- If j is a leaf node of T, then $\mathcal{R}^j = \emptyset$,
- If j is an introduce node of T with the unique child j', then $\mathcal{R}^j$ is the set of ordered partitions $(Y'_0, ..., Y'_l)$ of X_j such that $\mathcal{P}^{j'}$ is an l-cover of $\mathcal{R}^j$ with respect to $V(G_{j'}) \setminus X_{j'}$,
- If j is a forget node of T with the unique child j' and $\{v\} = X_{j'} \setminus X_j$, then we first reconstruct $\mathcal{R}^j$ as the set of ordered partitions $(Y'_0, ..., Y'_l)$ of X_j such that $P^{j'}$ is an l-cover of $\mathcal{R}^j$ with respect to $V(G_{j'}) \setminus X_{j'}$, and then we set $\mathcal{R}^j = \mathcal{R}^j \cup \mathcal{P}r_{v,l}(X_i)$ if $x \in Z$,
- If j is a join node of T with the two children j' and j'', set $\mathcal{R}^j = \mathcal{P}^{j'} \cup \mathcal{P}^{j''}$.

Construction of a leaf node. Let $X_i = \{x\}$. It is easy to verify if for any $\left\{\mathcal{P}^j \mid j \in I_i\right\}$ satisfying 2, $w_i\left(\emptyset, \left\{\mathcal{P}^j \mid j \in I'_i\right\}\right) = 0$ and $w_i\left(\{x\}, \left\{\mathcal{P}^j \mid j \in I'_i\right\}\right) = 1$.

For the description of the rest of the types of nodes, we assume that the table values have already been calculated for all the descendants of i in T. Before performing the computation we also set $w_i\left(\{x\}, \left\{\mathcal{P}^j \mid j \in I'_i\right\}\right) = +\infty$ for all Z and $\left\{\mathcal{P}^j \mid j \in I'_i\right\}$ that satisfy 1–3.

Construction of an introduce node. Let i' be the child of i and $\{v\} = X_i \backslash X_{i'}$. Let us consider every Z and $\left\{\mathcal{P}^j \mid j \in I'_i\right\}$ that satisfy 1–3 for the node i' such that $w_{i'}\left(Z, \left\{\mathcal{P}^j \mid j \in I_{i'}\right\}\right) \leq +\infty$.

We construct $\mathcal{R}\left(Z, \left\{\mathcal{P}^j \mid j \in I_{i'}\right\}\right) = \left\{\mathcal{R}^j \mid j \in J_{i'}\right\}$ and for $j \in I_i$, set $\hat{\mathcal{P}}^j = \mathcal{R}^j$. Notice $J_{i'} = I_i$. There are two cases we consider:

1. If $u \neq v$, $\hat{Z} = Z \cap Y_i$. Consider every set $\hat{\mathcal{P}}^0$ of ordered partitions $\left(\hat{Y}_0, ..., \hat{Y}_l\right)$ of X_i satisfying 2 for node i such that $\hat{\mathcal{P}}^0$ is an l-cover of $\mathcal{P}^0$ with respect to $V(G) \setminus V(G_i)$.

   We verify that one of the following conditions holds:

   - There is $z \in \hat{Z}$ that resolves x, v,
   - There is an ordered partition $(Y_0, ..., Y_l) \in \mathcal{P}^0$ of X_i such that a vertex $z \in V(G)(G_i)$ with $\mathcal{P}r_{z,l}(X_i) = (Y_0, .., Y_l)$ resolves x, v,

- There is an ordered partition $(Y_0, ..., Y_l) \in \mathcal{P}^h$ of X_h for $h \in I_i'$ such that a vertex $z \in V(G_h) \setminus X_h$ with $\mathcal{P}r_{z,l}(X_h) = (Y_0, ..., Y_l)$ resolves x, v.

Using Lemma 6.6 we can verify the conditions in polynomial time and set $w_i\left(\hat{Z}, \left\{\hat{\mathcal{P}}^j \mid j \in I_i'\right\}\right) = w_{i'}\left(\hat{Z}, \left\{\hat{\mathcal{P}}^j \mid j \in I_{i'}'\right\}\right)$ and if the condition holds we set

$$w_i\left(\hat{Z}, \left\{\hat{\mathcal{P}}^j \mid j \in I_i'\right\}\right) = \min\left(w_i\left(\hat{Z}, \left\{\hat{\mathcal{P}}^j \mid j \in I_i'\right\}\right), w_{i'}\left(\hat{Z}, \left\{\hat{\mathcal{P}}^j \mid j \in I_{i'}'\right\}\right)\right).$$

2. A set $\hat{Z} = (Z \cap Y_i) \cup \{v\}$ if $|Z \cap Y_I| \leq k - 1$. Consider every set $\hat{\mathcal{P}}^0$ of ordered partitions $\left(\hat{Y}_0, ..., \hat{Y}_l\right)$ of X_i that satisfies 2 for the node i such that $\hat{\mathcal{P}}^0$ is an l-cover if $\mathcal{P}^0$ or $\mathcal{P}^0 \setminus \{\mathcal{P}r_{v,l}(X_{i'})\}$ with respect to $V(G) \setminus V(G_i)$. We set

$$w_i\left(\hat{Z}, \left\{\hat{\mathcal{P}}^j \mid j \in I_i'\right\}\right) = \min\left(w_i\left(\hat{Z}, \left\{\hat{\mathcal{P}}^j \mid j \in I_i'\right\}\right), w_{i'}\left(\hat{Z}, \left\{\hat{\mathcal{P}}^j \mid j \in I_{i'}'\right\}\right) + 1\right).$$

Construction of a forget node. Let i' be the child of i and $\{v\} = X_i' \setminus X_i$. Consider every Z and $\left\{\mathcal{P}^j \mid j \in I_{i'}'\right\}$ satisfying 1–3 for the node i' such that $w_{i'}\left(Z, \left\{\mathcal{P}^j \mid j \in I_{i'}'\right\}\right)$ is finite. We construct $\mathcal{R}\left(Z, \left\{\mathcal{P}^j \mid j \in I_{i'}\right\}\right) = \left\{\mathcal{R}^j \mid j \in J_{i'}\right\}$ and for $j \in I_i$, set $\hat{\mathcal{P}}^j = \mathcal{R}^j$. Set $\hat{Z} = Z \cap Y_i$. We consider every set $\hat{\mathcal{P}}^0$ of ordered partitions $\left(\hat{Y}_0, ..., \hat{Y}_l\right)$ of X_i that satisfies 2 for the node i such that $\hat{\mathcal{P}}^0$ is an l-cover of $\mathcal{P}^0$ with respect to $V(G) \setminus V(G_i)$. We set

$$w_i\left(\hat{Z}, \left\{\hat{\mathcal{P}}^j \mid j \in I_i'\right\}\right) = \min\left(w_i\left(\hat{Z}, \left\{\hat{\mathcal{P}}^j \mid j \in I_i'\right\}\right), w_{i'}\left(\hat{Z}, \left\{\hat{\mathcal{P}}^j \mid j \in I_{i'}'\right\}\right)\right).$$

Construction of a join node. Let i' and i'' be the children of i. Recall that $X_i = X_{i'} = X_{i''}$. Consider every Z_1 and $\left\{\hat{\mathcal{P}}_1^j \mid j \in I_{i'}'\right\}$ satisfying 1–3 for the node i' such that $w_{i'}\left(\hat{Z}, \left\{\hat{\mathcal{P}}_1^j \mid j \in I_{i'}'\right\}\right)$ is finite and every Z_2 and $\left\{\hat{\mathcal{P}}_2^j \mid j \in I_{i''}'\right\}$) satisfying 1–3 for the node i'' such that $w_{i''}\left(\hat{Z}, \left\{\hat{\mathcal{P}}_1^j \mid j \in I_{i''}'\right\}\right)$ and $Z_1 \cap X_i = Z_2 \cap X_i$.

We set $Z = (Z_1 \cup Z_2) \cap Y_i$. For every $j \in I_{i'}$, we can construct the set $\mathcal{S}_1^j$ of ordered partitions $(Y_0, ..., Y_l)$ of X_i such that $\mathcal{P}_1^j$ is an l-cover of $\mathcal{S}_1^j$, and set

$$S_1 = \left(\cup_{j \in I_{i'}} \mathcal{S}_1^j\right) \cup \left(\cup_{v \in Z_1 \setminus X_i} \mathcal{P}r_{v,l}(X_i)\right).$$

In the same way for every $j \in I_{i''}$ we construct the set $\mathcal{S}_1^j$ of ordered partitions $(Y_0, ..., Y_l)$ of X_i such that $\mathcal{P}_2^j$ is an l-cover of $\mathcal{S}_2^j$, and set

$$S_2 = \left(\cup_{j \in I_{i''}} \mathcal{S}_2^j\right) \cup \left(\cup_{v \in Z_2 \setminus X_i} \mathcal{P}r_{v,l}(X_i)\right).$$

Consider every set $\mathcal{P}^0$ of ordered partitions $(Y_0, ..., Y_l)$ of X_i that satisfy 2 for the node i such that $\mathcal{P}_1^0 = \mathcal{P}^0 \cup \mathcal{S}_2$ and $\mathcal{P}_2^0 = \mathcal{P}^0 \cup \mathcal{S}_1$.

Observe that $I_i = J_{i'} \cup J_{i''}$. We construct $\mathcal{R}\left(Z_1, \left\{\mathcal{P}_1^j \mid j \in I_{i'}\right\}\right) = \left\{\mathcal{R}^j \mid j \in J_{i'}\right\}$ and $\mathcal{R}\left(Z_2, \left\{\mathcal{P}_2^j \mid j \in I_{i''}\right\}\right) = \left\{\mathcal{R}^j \mid j \in J_{i''}\right\}$. By setting $\mathcal{P}^j = \mathcal{R}^j$ for $j \in I_i$ we define $\left\{\mathcal{P}^j \mid j \in I_i'\right\}$.

We verify that one of the following conditions hold:
For every $x \in V(G_{i'}) \setminus X_i$ and $y \in V(G_{i''}) \setminus X_i$

- There is $v \in Z$ that resolves x and y,
- There is an ordered partition $(Y_0, ..., Y_l) \in \mathcal{P}^0$ of X_i such that a vertex $v \in V(G) \setminus V(G_i)$ with $\mathcal{P}r_{v,l}(X_i) = (Y_0, ..., Y_l)$ resolves x and y,

- There is an ordered partition $(Y_0, ..., Y_l) \in \mathcal{P}^j$ of X_j for $j \in I_i$ such that $x, y \notin V(G_j) \setminus X_j$ and a vertex $v \in V(G_j) \setminus X_j$ with $\mathcal{P}r_{v,l}(X_j) = (Y_0, ..., Y_l)$ resolves x and y,
- $x \in V(G_j) \setminus X_j$ for $j \in I_i$ and $\mathcal{P}^j \neq \emptyset$,
- $y \in V(G_j) \setminus X_j$ for $j \in I_i$ and $\mathcal{P}^j \neq \emptyset$.

This can be verified in polynomial time using Lemma 6.6. If the conditions hold we set

$$w_i\left(Z, \left\{\mathcal{P}^j \mid j \in I'_i\right\}\right) = \min(w_i\left(Z, \left\{\mathcal{P}^j \mid j \in I'_i\right\}\right), \\ w_{i'}\left(Z_1, \left\{\mathcal{P}^j \mid j \in I'_{i'}\right\}\right) + w_{i''}\left(Z_2, \left\{\mathcal{P}^j \mid j \in I'_{i''}\right\}\right) - |Z_1 \cap X_i|)$$

Now we perform the running time analysis of the algorithm. This is done by setting an upper bound for each of the tables. Let $i \in V(T)$. The size of X_i is at most $w(\Delta, l)$. The size of $N^s_{T_i} \leq 2^{s+1} - 1$. Hence, $|Y_i| \leq \left(2^{s+1} - 1\right) \cdot w(\Delta, l)$, and there can be at most $2^{\left(2^{s+1}-1\right) \cdot w(\Delta,l)}$ possibilities to choose Z. We have that $|I'_i| \leq 2^s + 1$. The number of all ordered partitions $(Y_0, ..., Y_l)$ of any X_j is at most $(l+1)^{|x_j|} \leq (l+1)^{w(\Delta,l)}$. Hence, the table for the node i contains at most $2^{\left(2^{s+1}-1\right) \cdot w(\Delta,l)} \cdot (l+1)^{\left(2^s+1\right) \cdot w(\Delta,l)}$ values of the function $w_i\left(Z, \left\{P^j | j \in I'_i\right\}\right)$.

Since the number of ordered partitions $(Y_0, ..., Y_l)$ of X_i is at most $(l+1)^{w(\Delta,l)}$, we obtain that table can be constructed in

$$O^*\left(2^{2 \cdot \left(2^{s+1}-1\right) \cdot w(\Delta,l)} \cdot (l+1)^{\left(2^{s+1}+3\right) \cdot w(\Delta,l)}\right)$$

And since the preliminary steps of the algorithm can be done in polynomial time and the algorithm is run for at most n choices of u, the time of the construction of the tables is also the final running time of the algorithm.

7 Modular-width

In this section, we present an algorithm for the METRIC DIMENSION *problem that runs in linear time with respect to the modular-width as the original authors do, and later we show a counterexample to the algorithm.*

Let X be a module of a graph G and $v \in V(G) \setminus X$. We can make the observation that the distances in G between v and the vertices of X are the same. This is expressed by the next lemma.

▶ **Lemma 7.1** ([2])**.** Let $X \subseteq V(G)$ be a module of a connected graph G and $|X| \geq 2$. Let also H be a graph obtained from $G[X]$ by addition of a universal vertex. Then any $v \in V(G)$ resolving $x, y \in X$ is a vertex of X, and if $W \subseteq V(G)$ is a resolving set of G, then $W \cap X$ resolves X in H.

▶ **Theorem 7.2** ([2])**.** The metric dimension of a connected graph G of modular-width at most t can be computed in time $O\left(t^3 4^t n + m\right)$.

The authors [2] describe the intuition behind the function $w(\cdot)$ as follows.

To compute $\mathrm{md}(G)$, consider auxiliary values $w(H, p, q)$ for a graph H of modular-width t with at least two vertices and boolean variables p and q as follows. Let H' be a graph obtained from H by the addition of a universal vertex u. Notice that $\mathrm{diam}_{H'}(V(H)) \leq 2$. Then the minimum size of set $W \subseteq V(H)$ such that

1. W resolves $V(H)$ in H',
2. H has a vertex x such that $\mathrm{dist}_{H'}(x, v) = 1$ for every $v \in W$ if and only if $p = true$,

3. H has a vertex x such that $\text{dist}_{H'}(x, v) = 2$ for every $v \in W$ if and only if $q = true$.

The assumption is made that $w(H, p, q) = +\infty$ if such set does not exist.

Let G be a graph, X its module, $H = G[X]$ and let $H_1, ..., H_s$ be the partition of H into modules, of which t, $t \leq s$ are trivial. Assume Z is a hypothetical optimal resolving set and $Z' = Z \cap X$. Every pair of vertices in H must be resolved by a vertex in Z', by Lemma 7.1. This means that we need to compute a set that will, amongst others, satisfy the property that the set will be a resolving set for the vertices in X. As we have stated above, those vertices are either adjacent or at a distance 2 from each other in G. This is why it is required for W to be resolving set of $V(H)$ in H'.

It could also happen that a vertex $z \in Z'$ is required to resolve a pair of vertices $x \in X$ and $y \in H \setminus X$. If x is at distance 1 from every vertex of Z', z is also required to resolve $x' \in X$ and y. The same argument can be made for vertices at distance 2 from every vertex of Z'. That is why it suffices to know whether there is a vertex in X which is at distance 1 from every vertex of Z'. This is the meaning of the booleans p and q.

Since H has modular-width at most t, it can be constructed from single vertex graphs by the disjoint union and the join operations and decomposing H into at most t modules. In the rest of the computation, $w(H, p, q)$ is described given the modular decomposition of H and the values computed for the child nodes. Since the base case corresponds to a graph of at most size t we may compute the values for leaf nodes by brute force, followed by executing a bottom up dynamic programming algorithm.

The algorithm [2] In the original article the algorithm is split into 3 cases.

- Graph H is a disjoint union of pair of graphs,
- Graph H is a join of a pair of graphs,
- Graph H can be partitioned into at most t graphs, each of modular-width at most t.

The first two cases are subsumed by the third case, but just like the original authors, we shall keep them for clarity of the algorithm. We shall also skip the proof of correctness of the first part of the algorithm and only focus on explaining the proof of the last two theorems, since no other parts of the proofs will need altering in the next chapter.

Disjoint union. Let H be a disjoint union of H_1 and H_2. Then we can assume that $|V(H_1)| \leq |V(H_2)|$. Then there are 3 cases that can occur.

First, if $|V(H_1)| = |V(H_2)| = 1$, then we can easily verify whether the conditions are met and $w(H, false, true) = 1$, $w(W, false, false) = 2$, and $w(H, true, true) = w(true, false) = +\infty$ or $w(\cdot) = +\infty$ if the conditions are not satisfied.

Second, if $|V(H_1)| = 1, |V(H_2)| > 1$ and the values of $w(H_2, p, q)$ are already computed for $p, q \in \{true, false\}$, then the single vertex of H_1 is at distance 2 from any vertex of H_2 in H'. Notice that the vertex of H_1 can, but does not have to, be in the resolving set. By Lemma 7.1

- $w(H, true, true) = w(H_2, true, false)$,
- $w(H, false, true) = \min\{w(H_2, false, false), w(H_2, true, true) + 1, w(H_2, false, true) + 1\}$,
- $w(H, true, false) = +\infty$,
- $w(H, false, false) = \min\{w(H_2, true, false) + 1,\ w(H_2, false, false) + 1\}$.

Third, if $|V(H_1)| > 1, |V(H_2)| > 1$ and the values of $w(H_i, p, q)$ are already computed for $i \in \{1, 2\}$ and $p, q \in \{true, false\}$, then observe that for $x \in V(H_1)$ and $y \in V(H_2)$, $\text{dist}_{H'}(x, y) = 2$ and that any resolving set has at least one vertex in both H_1 and H_2. By Lemma 7.1

- $w(H, true, true) = +\infty$,
- $w(H, true, false) = +\infty$,
- $w(H, false, true) = \min\{w(H_1, p_1, q_1) + w(H_2, p_2, q_2) + 1 \mid p_i, q_i \in \{true, false\}$, $i \in \{1, 2\}$ and $q_1 \neq q_2\}$,
- $w(H, false, false) = \min\{w(H_1, p_1, false) + w(H_1, p_2, false) \mid p_1, p_2 \in \{true, false\}\}$.

Join. H is a join of H_1 and H_2. We can assume that $|V(H_1)| \leq |V(H_2)|$. Again, there are 3 cases that can occur.

First, if $|V(H_1)| = |V(H_2)| = 1$, then we can easily verify whether $w(H, true, false) = 1$, $w(H, false, false) = 2$, and $w(H, true, true) = w(H, false, true) = +\infty$.

Second, if $|V(H_1)| = 1, |V(H_2)| > 1$ and the values of $w(H_2, p, q)$ are already computed for $p, q \in \{true, false\}$, then the single vertex of H_1 is at distance 1 from any vertex of H_2 in H'. Notice that the vertex of H_1 can, but does not have to, be in the resolving set. By Lemma 7.1

- $w(H, true, true) = w(H_2, false, true)$,
- $w(H, false, true) = +\infty$,
- $w(H, true, false) = \min\{w(H_2, false, false), w(H_2, true, true)+1, w(H_2, true, false)+1\}$,
- $w(H, false, false) = \min\{w(H_2, false, true) + 1,\ w(H_2, false, false) + 1\}$.

Third, if $|V(H_1)| > 1, |V(H_2)| > 1$ and the values of $w(H_i, p, q)$ are already computed for $i \in \{1, 2\}$ and $p, q \in \{true, false\}$, then observe that for $x \in V(H_1)$ and $y \in V(H_2)$, $\mathrm{dist}_{H'}(x, y) = 1$ and that any resolving set has at least one vertex in both H_1 and H_2. By Lemma 7.1

- $w(H, true, true) = +\infty$,
- $w(H, false, true) = +\infty$,
- $w(H, true, false) = \min\{w(H_1, p_1, q_1) + w(H_2, p_2, q_2) \mid p_i, q_i \in \{true, false\}$, $i \in \{1, 2\}$ and $p_1 \neq p_2\}$,
- $w(H, false, false) = \min\{w(H_1, false, q_1) + w(H_1, false, q_2) \mid q_1, q_2 \in \{true, false\}\}$.

Partitioning into modules. Let $V(H)$ be partitioned into $s \leq t$ non-empty modules $X_1, ..., X_s,\ s \geq 2$. We assume that $X_1, ..., X_h$ are *trivial*, this means that $|X_i| = 1$ for $i \in \{1, ..., h\}$ where $0 \leq h \leq s$. For distinct $i, j \in \{1, ..., s\}$, either vertex of X_i is adjacent to every vertex of X_j or the vertices of X_i and X_j are not adjacent. Let F be the *prime* graph with a vertex set $\{v_1, ..., v_s\}$ such that v_i is adjacent to v_j if and only if the vertices of X_i are adjacent to the vertices of X_j. Let F' be obtained by addition of a universal vertex to the graph F. Observe that if $x \in X_i$ and $y \in X_j$ for distinct $i, j \in \{1, ..., s\}$, then $\mathrm{dist}_{H'}(x, y) = \mathrm{dist}_{F'}(v_i, v_j)$.

For boolean variables p, q, a set of indices $I \subseteq \{1, ..., h\}$, and boolean variables p_i, q_i where $i \in \{h+1, ..., s\}$ we define

$$\omega(p, q, I, p_{h+1}, q_{h+1}, ..., p_s, q_s) = |I| + \sum_{i=h+1}^{s} w(H[X_i], p_i, q_i)$$

if the following holds:

1. the set $Z = \{v_i \mid i \in I \cup \{h+1, ..., s\}\}$ resolves $V(F)$ in F',

2. if $p_i = true$ for some $i \in \{h+1, ..., s\}$, then for each $j \in \{1, ..., h\} \setminus I$, $\mathrm{dist}_{F'}(v_i, v_j) = 2$ or there is $v_r \in Z$ such that $r \neq i, j$ and $\mathrm{dist}_{F'}(v_r, v_i) \neq \mathrm{dist}_{F'}(v_r, v_j)$,

3. if $q_i = true$ for some $i \in \{h+1, ..., s\}$, then for each $j \in \{1, ..., h\} \setminus I$, $\mathrm{dist}_{F'}(v_i, v_j) = 1$ or there is $v_r \in Z$ such that $r \neq i, j$ and $\mathrm{dist}_{F'}(v_r, v_i) \neq \mathrm{dist}_{F'}(v_r, v_j)$,

4. if $p_i = p_j = true$ for some distinct $i, j \in \{h+1, ..., s\}$, then $\mathrm{dist}_{F'}(v_i, v_j) = 2$ or there is $v_r \in Z$ such that $r \neq i, j$ and $\mathrm{dist}_{F'}(v_r, v_i) \neq \mathrm{dist}_{F'}(v_r, v_j)$,

5. if $q_i = q_j = true$ for some distinct $i, j \in \{h+1, ..., s\}$, then $\mathrm{dist}_{F'}(v_i, v_j) = 1$ or there is $v_r \in Z$ such that $r \neq i, j$ and $\mathrm{dist}_{F'}(v_r, v_i) \neq \mathrm{dist}_{F'}(v_r, v_j)$,

6. $p = true$ if and only if there is $i \in \{1, ..., h\} \setminus I$ such that $\mathrm{dist}_{F'}(v_i, v_j) = 1$ for $v_j \in Z$ or there is $i \in \{h+1, ..., s\}$ such that $p_i = true$ and $\mathrm{dist}_{F'}(v_i, v_j) = 1$ for $v_j \in Z \setminus \{v_i\}$,

7. $q = true$ if and only if there is $i \in \{1, ..., h\} \setminus I$ such that $\mathrm{dist}_{F'}(v_i, v_j) = 2$ for $v_j \in Z$ or there is $i \in \{h+1, ..., s\}$ such that $q_i = true$ and $\mathrm{dist}_{F'}(v_i, v_j) = 2$ for $v_j \in Z \setminus \{v_i\}$,

and $\omega(p, q, I, p_{h+1}, q_{h+1}, ..., p_s, q_s) = +\infty$ in other cases.

▶ **Lemma 7.3** ([2])**.** The function w is defined as $w(H, p, q) = \min \omega(p, q, I, p_{h+1}, q_{h+1}, ..., p_s, q_s)$, where the minimum is taken over all possible sets $I \subseteq \{1, ..., h\}$ and booleans $p_i, q_i \in \{true, false\}$ for $i \in \{h+1, ..., s\}$.

Now we explain how Belmonte et al.[2] compute the metric dimension using the established function. Since G is a connected graph of modular-width at most t, it is either a single vertex graph, or it is a join of two graph or it can be partitioned into $s \leq t$ modules $X_1, ..., X_s$ such that $\mathrm{mw}(G[X_i]) \leq t$ for $i \in \{1, ..., s\}$.

Single vertex. If $|V(G)| = 1$ it should be obvious, that $\mathrm{md}(G) = 1$.

Join. If G is a join of H_1 and H_2, we can assume that $|V(H_1)| \leq |V(H_2)|$. Three cases can occur.

First, if $|V(H_1)| = |V(H_2)| = 1$, then $\mathrm{md}(G) = 1$.

Second, if $|V(H_1)| = 1$ and $|V(H_2)| > 1$ and the values of $w(H_i, p, q)$ have already been computed for $p, q \in \{true, false\}$, then by the definition of join of a graph, the vertex from H_1 is at distance 1 from all of the vertices of H_2 in G. This vertex can, but does not have to be in the resolving set. By Lemma 7.1

$$\mathrm{md}(G) = \min\{w(H_2, false, true),\ w(H_2, false, false),\ w(H_2, true, true) + 1\}.$$

Third, $|V(H_1)|, |V(H_2)| > 1$ and the values of $w(H_i, p, q)$ have already been computed for $p, q \in \{true, false\}$. By the definition of the join of graph, avertex from H_1 is at distance 1 from all of the vertices of H_2 in G, and every resolving set has at least one vertex in H_1 and one vertex in H_2. By Lemma 7.1

$$\mathrm{md}(G) = \min\{w(H_1, p_1, q_1) + w(H_2, p_2, q_2)\ \mid p_i, q_i \in \{true, false\}, i \in \{1, 2\} \text{ and } p_1 \neq p_2\}.$$

Partitioning into modules. Let $V(H)$ be partitioned into $s \leq t$ non-empty modules $X_1, ..., X_s$, $s \geq 2$. We assume that $X_1, ..., X_h$ are *trivial*, this means that $|X_i| = 1$ for $i \in \{1, ..., h\}$ where $0 \leq h \leq s$. Let F be the *prime* graph with a vertex set $\{v_1, ..., v_s\}$ such that v_i is adjacent to v_j if and only if the vertices of X_i are adjacent to the vertices of X_j. Let F' be obtained by addition of a universal vertex to the graph F. Observe that if $x \in X_i$ and $y \in X_j$ for distinct $i, j \in \{1, ..., s\}$, then $\mathrm{dist}_G(x, y) = \mathrm{dist}_F(v_i, v_j)$.

For a set of indices $I \subseteq \{1, ..., h\}$ and boolean variables p_i, q_i where $i \in \{h+1, ..., s\}$, we define

$$\omega\left(I, p_{h+1}, q_{h+1}, ..., p_s, q_s\right) = |I| + \sum_{i=h+1}^{s} w\left(G\left[X_i\right], p_i, q_i\right)$$

if the following holds:

1. the set $Z = \{v_i \mid i \in I \cup \{h+1, ..., s\}\}$ is a resolving set for F,
2. if $p_i = true$ for some $i \in \{h+1, ..., s\}$, then for each $j \in \{1, ..., h\} \setminus I$, $\mathrm{dist}_F\left(v_i, v_j\right) \geq 2$ or there is $v_r \in Z$ such that $r \neq i, j$ and $\mathrm{dist}_F\left(v_r, v_i\right) \neq \mathrm{dist}_{F'}\left(v_r, v_j\right)$,
3. if $q_i = true$ for some $i \in \{h+1, ..., s\}$, then for each $j \in \{1, ..., h\} \setminus I$, $\mathrm{dist}_F\left(v_i, v_j\right) \neq 2$ or there is $v_r \in Z$ such that $r \neq i, j$ and $\mathrm{dist}_F\left(v_r, v_i\right) \neq \mathrm{dist}_{F'}\left(v_r, v_j\right)$,
4. $p_i = p_j = true$ for some distinct $i, j \in \{h+1, ..., s\}$, then $\mathrm{dist}_F\left(v_i, v_j\right) \geq 2$ or there is some $v_r \in Z$ such that $r \neq i, j$ and $\mathrm{dist}_{F'}\left(v_r, v_i\right) = \mathrm{dist}_F\left(v_r, v_j\right)$,
5. $q_i = q_j = true$ for some distinct $i, j \in \{h+1, ..., s\}$, then $\mathrm{dist}_F\left(v_i, v_j\right) \neq 2$ or there is some $v_r \in Z$ such that $r \neq i, j$ and $\mathrm{dist}_{F'}\left(v_r, v_i\right) = \mathrm{dist}_F\left(v_r, v_j\right)$.,

and $\omega\left(I, p_{h+1}, q_{h+1}, ..., p_s, q_s\right) = +\infty$ in all other cases.
The claim about the metric dimension is expressed by the following theorem.

▶ **Theorem 7.4** ([2])**.** The function md, that expresses the metric dimension of a graph can be expressed as $\mathrm{md}(G) = \min \omega\left(I, p_{h+1}, q_{h+1}, ..., p_s, q_s\right)$, where the minimum is taken over all possible values of $I \subseteq \{1, ..., h\}$ and $p_i, q_i \in \{true, false\}$ for $i \in \{h+1, ..., s\}$.

Proof. We first prove that $\mathrm{md}(G) \geq \min \omega\left(I, p_{h+1}, q_{h+1}, ..., p_s, q_s\right)$:

Let $W \subseteq V(G)$ be a resolving set of minimum size. By definition, $\mathrm{md}(G) = |W|$. Let $W_i = W \cap X_i$. Let $I = \{i \mid i \in \{1, ..., h\}, W_i \neq \emptyset\}$. By Lemma 7.1, $W_i \neq \emptyset$ for $i \in \{h+1, ..., s\}$. For $i \in \{h+1, ..., s\}$, let $p_i = true$ if there is a vertex $x \in X_i$ such that $\mathrm{dist}_G(x, u) = 1$ for some $u \in W_i$, and let $q_i = true$ if there is a vertex $y \in X_i$ such that $\mathrm{dist}_G(y, u) = 2$ for some $u \in W_i$.

By Lemma 7.1, W_i resolves X_i in $G'\left[X_i\right]$ for $i \in \{h+1, ..., s\}$. This implies that $|W_i| \geq w\left(G\left[X_i\right], p_i, q_i\right)$ for $i \in \{h+1, ..., s\}$ and therefore $|W| \geq |i| + \sum_{i=h+1}^{s} w\left(G\left[X_i\right], p_i, q_i\right)$.

Now we prove that each of the conditions 1-5 is met for the chosen values of I, p_i and q_i.

1. Let v_i, v_j be distinct vertices in F. If $v_i \in Z$ or $v_j \in Z$, Z obviously resolves v_i and v_j. Let $i, j \in \{1, ..., h\} \setminus I$. Then X_i, X_j are trivial modules with vertices x, y respectively. Since W is a resolving set of G, there has to be $u \in W$ such that $\mathrm{dist}_G(u, x) \neq \mathrm{dist}_G(u, y)$. Consider the set W_r containing u. Vertices v_i, v_j are resolved by v_r, because $\mathrm{dist}_F\left(v_r, v_i\right) = \mathrm{dist}_G(u, x) \neq \mathrm{dist}_G(u, y) = \mathrm{dist}_G\left(v_r, v_j\right)$.
2. Assume that $p_i = true$ for some $i \in \{h+1, ..., s\}$ and consider some $j \in \{1, ..., h\} \setminus I$. Let us also assume that $\mathrm{dist}_F\left(v_i, v_j\right) = 1$. Then X_i has to have a vertex x adjacent to all the vertices of W_i. Let y be the unique vertex of X_j. The set W resolves x, y, which means there is $u \in W$ such that $\mathrm{dist}_G(u, x) \neq \mathrm{dist}_G(u, y)$. If $u \in X_i$ then we have that $\mathrm{dist}_G(u, x) = 1 = \mathrm{dist}_F\left(v_i, v_j\right) = \mathrm{dist}_G(u, y)$. That is a contradiction. Thus u cannot belong to X_i. Let then X_r be the module containing u. Then we have that $\mathrm{dist}_F\left(v_r, v_i\right) = \mathrm{dist}_G(u, x) \neq \mathrm{dist}_G(u, y) = \mathrm{dist}_F\left(v_r, v_j\right)$.
3. Assume that $q_i = true$ for some $i \in \{h+1, ..., s\}$ and consider some $j \in \{1, ..., h\} \setminus I$. Let us also assume that $\mathrm{dist}_F\left(v_i, v_j\right) = 2$. Then X_i has to have a vertex x at distance 2 from all the vertices of W_i. Let y be the unique vertex of X_j. The set W resolves x, y, which means there is $u \in W$ such that $\mathrm{dist}_G(u, x) \neq \mathrm{dist}_G(u, y)$. If $u \in X_i$ then we have that $\mathrm{dist}_G(u, x) = 2 = \mathrm{dist}_F\left(v_i, v_j\right) = \mathrm{dist}_G(u, y)$. That is a contradiction. This means that u cannot belong to X_i. Let then X_r be the module containing u. Then we have that $\mathrm{dist}_F\left(v_r, v_i\right) = \mathrm{dist}_G(u, x) \neq \mathrm{dist}_G(u, y) = \mathrm{dist}_F\left(v_r, v_j\right)$.

4. Suppose that $p_i = p_j = true$ for some $i \in \{h+1, ..., s\}$ and assume that $\mathrm{dist}_F(v_i, v_j) = 1$. Then X_i has a vertex x adjacent to all the vertices of W_i and X_j has a vertex y that is adjacent to all the vertices of W_j. The set W resolves x, y and therefore there has to be $u \in W$ such that $\mathrm{dist}_G(u, x) \neq \mathrm{dist}_G(u, y)$. If $u \in X_i$ then we get that $\mathrm{dist}_G(u, x) = \mathrm{dist}_F(v_i, v_j) = \mathrm{dist}_G(u, y)$. That is a contradiction. This means that u cannot belong to X_i. We get that $u \notin X_j$ by the same argument. Let then X_r be the module containing u. Then we have that $\mathrm{dist}_F(v_r, v_i) = \mathrm{dist}_G(u, x) \neq \mathrm{dist}_G(u, y) = \mathrm{dist}_F(v_r, v_j)$.

5. Suppose that $q_i = q_j = true$ for some $i \in \{h+1, ..., s\}$ and assume that $\mathrm{dist}_F(v_i, v_j) = 2$. Then X_i has a vertex x at distance 2 to all the vertices of W_i and X_j has a vertex y that is at distance 2 to all the vertices of W_j. The set W resolves x, y and therefore there has to be $u \in W$ such that $\mathrm{dist}_G(u, x) \neq \mathrm{dist}_G(u, y)$. If $u \in X_i$ then we get that $\mathrm{dist}_G(u, x) = \mathrm{dist}_F(v_i, v_j) = \mathrm{dist}_G(u, y)$. That is a contradiction. This means that u cannot belong to X_i. We get that $u \notin X_j$ by the same argument. Let then X_r be the module containing u. Then we have that $\mathrm{dist}_F(v_r, v_i) = \mathrm{dist}_G(u, x) \neq \mathrm{dist}_G(u, y) = \mathrm{dist}_F(v_r, v_j)$.

All five of the conditions are fulfilled. By that the inequality
$\mathrm{md}(G) \geq \min \omega(I, p_{h+1}, q_{h+1}, ..., p_s, q_s)$ has been proven.

Now we prove that $\mathrm{md}(G) \leq \min \omega(I, p_{h+1}, q_{h+1}, ..., p_s, q_s)$:

Assume that the values of p_i, q_i for $i \in \{h+1, ..., s\}$ are chosen in such a way, that $\omega(I, p_{h+1}, q_{h+1}, ..., p_s, q_s)$ has the minimum possible value. If $\omega(I, p_{h+1}, q_{h+1}, ..., p_s, q_s) = +\infty$, the inequality holds trivially. Suppose that $\omega(I, p_{h+1}, q_{h+1}, ..., p_s, q_s)$ is finite. Then $\omega(I, p_{h+1}, q_{h+1}, ..., p_s, q_s) = |I| + \sum_{i=h+1}^{s} w(H[X_i], p_i, q_i)$ and 1-5 hold.

For $i \in \{h+1, .., s\}$, let $W_i \subseteq X_i$ be a set of minimum size such that:

1. W_i resolves X_i in the graph H_i',

2. X_i has a vertex x such that $\mathrm{dist}_{H_i'}(x, v) = 1$ for every $v \in W_i$ if and only if $p_i = true$,

3. X_i has a vertex x such that $\mathrm{dist}_{H_i'}(x, v) = 2$ for every $v \in W_i$ if and only if $q_i = true$.

By the definition, $w(G[X_i], p_i, q_i) = |W_i|$ for $i \in \{h+1, ..., s\}$. Let

$$W = (\cup_{i \in I} X_i) \cup \left(\cup_{i=h+1}^{s} W_i\right).$$

◀

We have that $|W| = \omega(p, q, I, p_{h+1}, q_{h+1}, ..., p_s, q_s)$.

▶ **Theorem 7.5** ([2]). *W is a resolving set for G.*

Proof. Let x, y be distinct vertices of G. Let us show that a vertex $u \in W$, that resolves x and y in G, exists. It is obvious, that is suffices to prove this for $x, y \in G \setminus W$. Let X_i, X_j be the modules that contain x, y, respectively. If $i = j$, then a vertex $u \in W_i$ resolves x and y in H_i' and, therefore, u resolves x and y in G. Assume that $i \neq j$.

First, assume $i, j \in \{1, ..., h\}$. Then $i, j \neq I$, because $X_1, ..., X_h$ are trivial. By 1, since Z is a resolving set for F, there is $v_r \in Z$ such that $\mathrm{dist}_F(v_r, v_i) \neq \mathrm{dist}_F(v_r, v_j)$. Set W_r is not empty, by the definition of W_r and Z. Let $u \in W_r$. Then $\mathrm{dist}_G(u, x) = \mathrm{dist}_F(v_r, v_i) \neq \mathrm{dist}_F(v_r, v_i) = \mathrm{dist}_G(u, y)$.

Now assume that $i \in \{h+1, ..., s\}$ and $j \in \{1, ..., h\}$. If there are $u_1, u_2 \in X_i$ such that $\mathrm{dist}_{H_i'}(u_1, x) \neq \mathrm{dist}_{H_i'}(u_2, x)$, then either u_1 or u_2 resolves x, y, because $\mathrm{dist}_G(u_1, y) = \mathrm{dist}_G(u_2, y)$. Assume that all the vertices of W_i are at the same distance from x in H_i'. Let $u \in W_i$. If $\mathrm{dist}_{H_i'}(u_1, x) \neq \mathrm{dist}_{H_i'}(u_2, x)$, then u_1 or u_2 must resolve x and y, because $\mathrm{dist}_G(u_1, x) \neq \mathrm{dist}_G(u_2, x)$. Suppose that all the vertices of W_i are at the same distance from x in H_i'. Let $u \in W_i$. If $\mathrm{dist}_{H_i'}(u, x) = 1$, then $p_i = true$ and by the second condition, $\mathrm{dist}_{F'}(v_i, v_j) \geq 2$ or there is $v_r \in Z$ such that $r \neq i, j$ and $\mathrm{dist}_F(v_r, v_i) \neq \mathrm{dist}_{F'}(v_r, v_j)$. If

$\text{dist}_{F'}(v_i, v_j) \geq 2$, x and y must be resolved by u, because $\text{dist}_G(u, y) = 2$. Otherwise x and y are resolved by any vertex $u' \in W_r$.

In the same way if $\text{dist}_G(u, x) = 2$, then $q_i = true$ and by the third condition $\text{dist}_F(v_r, v_j) \neq \text{dist}_F(v_r, v_j)$. If $\text{dist}_G(u, x) \neq 2$ then u resolves x and y. Now let $i, j \in \{h+1, ...s\}$. If $u_1, u_2 \in X_i$ such that $\text{dist}_{H'_i}(u_1, x) \neq \text{dist}_{H'_i}(u_2, x)$ exist, then x and y are resolved by either u_1 or u_2, since $\text{dist}_G(u_1, x) = \text{dist}_{F'}(u_2, x)$. The same argument can be used if there are $u_1, u_2 \in X_j$ such that $\text{dist}_{H'_i}(u_1, y) \neq \text{dist}_{H'_i}(u_2, y)$, then u_1 or u_2 resolves x, y. Assume all the vertices of W_i are at the same distance from y in H'_j. Let $u_1 \in W_i$ and $u_2 \in W_j$. If $\text{dist}_{H'_i}(u_1, x) \neq \text{dist}_{H'_i}(u_2, x)$, then u_1 or u_2 resolves x and y, since $\text{dist}_G(u_1, y) = \text{dist}_G(u_2, x)$. Suppose that $\text{dist}_{H'_i}(u_1, x) = \text{dist}_{H'_j}(u_2, y) = 1$. Then $p_i = p_j = true$ and by the fourth condition, $\text{dist}_{F'}(v_i, v_j) \geq 2$, or there is $v_r \in Z$ such that $r \neq i, j$ and $\text{dist}_F(v_r, v_i) \neq \text{dist}_F(v_r, v_j)$. If $\text{dist}_F(v_i, v_j) \neq 2$, then u_1 resolves x and y. Otherwise any vertex $u' \in W_r$ resolves x and y. If $\text{dist}_{H'_i}(u_1, x) = \text{dist}_{H'_j}(u_2, y) = 2$, then $q_i = q_j = true$ and by the fourth condition $\text{dist}_F(v_i, v_j) \neq 2$ or there is $v_r \in Z$ such that $r \neq i, j$ and $\text{dist}_F(v_r, v_i) \neq \text{dist}_F(v_r, v_j)$. If $\text{dist}_F(v_i, v_j) \neq 2$, then u_1 resolves x and y. Otherwise any vertex of W_r resolves x and y. ◀

We have shown that W is a resolving set for G and, therefore, $\text{md}(G) = |W|$ and $|W| = \omega(I, p_{h+1}, q_{h+1}, ..., p_s, q_s)$.

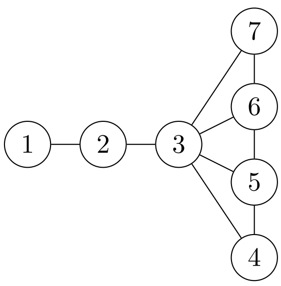
For evaluation of the running time of the function $w(H, p, q)$ we only need to consider the case when $V(H)$ can be partitioned into $s \leq t$ modules. We consider at most 4^t posibilities to choose I and p_i, q_i for $i \in \{h+1, ...s\}$. Then all the conditions can be verified in $O(t^3)$ time. Hence, the total time complexity is $O(4^t \cdot t^3)$. In the same way the computation of the function of $\text{md}(G)$ can be performed in $O(4^t \cdot t^3)$. The conclusion is that since the algorithm by Tedder et al. [7] is linear, we can solve the metric dimension problem in $O(4^t \cdot t^3 \cdot n + m)$ time.

7.1 Counterexample

Now we will present a counterexample to the algorithm bounded by metric-dimension.

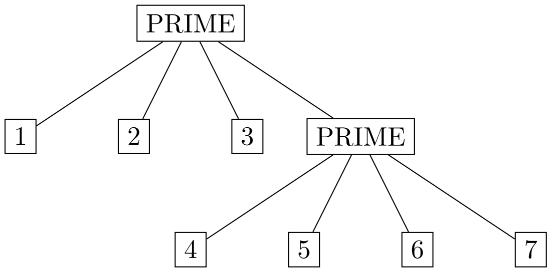
Let us have the graph G as shown in Figure 1.

Figure 1 Counterexample graph G.



This graph has the modular decomposition of Figure 2, where a PRIME module means the fourth operation of a modular decomposition as described in preliminaries.

Figure 2 Modular decomposition of the graph seen in Figure 1.



We can easily verify that the smallest resolving sets of G are the sets $\{4,7\}$ and $\{5,6\}$. These two are the only two resolving sets of size 2. Since both of the sets have only vertices from one module, we can do the calculation with either of them. Knowing that, we can compute what the boolean values p and q of the function $w\left(G\left[\{4,5,6,7\}\right],p,q\right)$. The function w is defined as $\min \omega\left(p,q,I,p_{h+1},q_{h+1},...,p_s,q_s\right)$. Let us then set $I=\{4,7\}$ and since we have no non-trivial sub-modules in this step of the computation, we do not have any p_i and q_i booleans to set. We compute the values like this (in reverse) since we already know the final resolving sets. One by one we check the conditions 1-7. The graph F' is constructed as in the description of the algorithm.

The graphs F and F' are below for clarity.

Figure 3 The prime graph F constructed from the subgraph of the graph seen in Figure 1, that is induced by the set of vertices $\{4,5,6,7\}$.

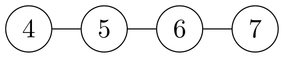
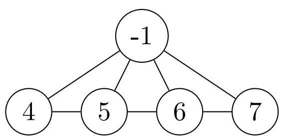


Figure 4 The graph F' with the universal vertex -1, constructed from the graph seen in Figure 3.



The conditions have following results:

1. The set $Z=\{4,7\}$ does resolve $V(F)$ in F' by Lemma 7.1, since it also resolves G.
2. There is no other non-trivial sub-module of the currently computed module, so the condition is implicitly satisfied.
3. There is no other non-trivial sub-module of the currently computed module, so the condition is implicitly satisfied.
4. There is no other non-trivial sub-module of the currently computed module, so the condition is implicitly satisfied.
5. There is no other non-trivial sub-module of the currently computed module, so the condition is implicitly satisfied.
6. For $i\in\{5,6\}$ there is a vertex $v_j\in Z$ for which $\text{dist}_{F'}\left(v_i,v_j\right)=1$. The indices $i=5$ and $j=4$ satisfy this condition.

7. For $i \in \{5,6\}$ there is a vertex $v_j \in Z$ for which $\text{dist}_{F'}(v_i, v_j) = 2$. The indices $i = 6$ and $j = 4$ satisfy this condition.

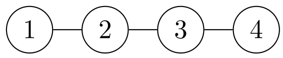
This means that it is required that $p = true$ and $q = true$ for the algorithm to evaluate $\{4,7\}$ as the smallest resolving set. Note that we get exactly the same values for the resolving set $\{5,6\}$. Since both of the sets are the same size and the vertices in the sets are from the same module, we can continue this computation with either of the resolving sets. We choose to continue with the set $\{4,7\}$. Until this point everything works as expected. This is about to change in the next step of the computation.

Knowing the values of p and q for the resolving set $\{4,7\}$ and the fact that this set resolves the whole graph G, we can check the conditions in the final step of computation of the function $\text{md}(G)$.

This means we are checking the function $\omega(\emptyset, true, true)$. The set I is an empty set as we do not need any more vertices to resolve any vertex.

The graph F can seen in Figure 5.

Figure 5 The prime graph F constructed from the graph seen in Figure 1. The vertex 4 represents the set of vertices $\{4,5,6,7\}$ from G. Other modules are trivial.



We proceed with the verification of the conditions.

1. We have that $Z = \{5\}$, and, thus the condition is satisfied, as the vertex in F' representing the sub-module is a vertex on the end of path. This means that all the distance vectors are unique.

2. Now we know that $p_4 = true$. This means that there has to be $j \in \{1, ..., h\} \setminus I$, such that $\text{dist}_F(v_i, v_j) \geq 2$, however, that is not true for $j = 3$. Additionally we cannot select $v_r \in Z$ such that $r \neq i, j$, because $Z = \{4\}$. This means that the condition is not satisfied.

3. Similar observation as in 2. can be made.

4. There is only one element to chose i, j from, so the condition is satisfied.

5. Same as above.

This means that $\omega(\emptyset, true, true) = +\infty$, because some of the conditions are not satisfied. This is a contradiction with the Theorem 7.4.

There is one other thing we considered. It comes down to inexactness in the conditions of the algorithm. Until now we suspected that when the authors say for $v_r \in Z$, the the condition needs to be met for only one of the vertices of Z. It could also be interpreted as for all the vertices in Z. However, even for this case we managed to find a counterexample which we shall present. We do not write the computation of the function w for the sub-modules, since there are many of them and are all done in the exact same way as in the previous case.

Consider a graph H as displayed in Figure 6 with the modular decomposition as shown in Figure 7. It can be verified that the size of any resolving set of this graph has size at least size 3. Such a set of the minimum size would have any 2 vertices from the set $\{1,2,3,4\}$ and any single vertex from the set $\{5,6\}$.

Figure 6 Counterexample graph H

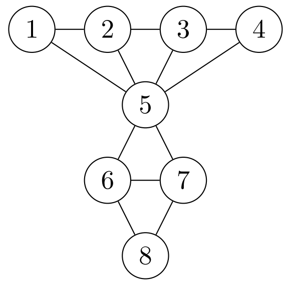
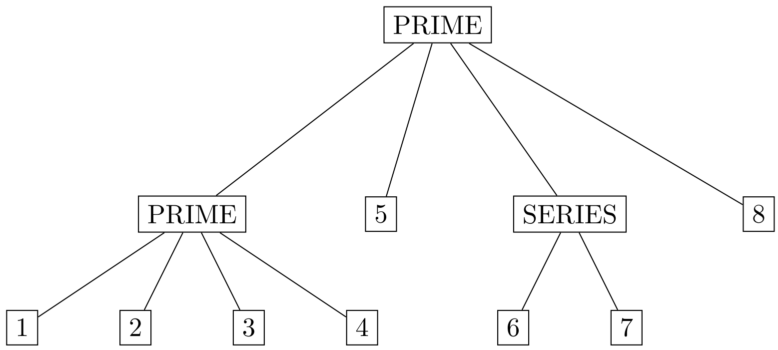


Figure 7 Modular decomposition of the graph seen in Figure 6.



Let us then do the computation of $w\,(H[\{1,2,3,4\}],p,q)$.Let us now assume that the values of p and q are computed correctly for each of the pairs of vertices. For the computation of $w\,(H[\{6,7\}],p,q)$ we can verify that for any set of indices of size one, the boolean values have only one valid configuration that fulfills the conditions, that is $p = true$ and $q = false$.

Now we construct the prime graphs F and F', the graph with the universal vertex -1, with the vertex 1 representing the sub-module of size 4 and the vertex 6 representing the sub-module of size 2. These graphs can be seen on Figure 8 and 9, respectively.

Figure 8 The prime graph F constructed from the graph seen in Figure 6.

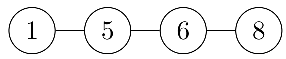
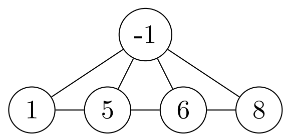


Figure 9 The graph F' with the universal vertex -1 constructed from the graph seen in Figure 6.



At this point we perform the final computation of the $\mathrm{md}(F)$ function. We set the value of I to the empty set as we do not need any other vertices to form a resolving set. Now we again check the conditions one by one.

1. We have that $Z = \{1,6\}$, and, thus Z does resolve the graph F.
2. We know that the $p_i = true$ for $i = 6$. This means that for each $j \in \{5,8\}$ we need the vertices v_i and v_j to be at least at distance 2 from each other in the graph F (which is fulfilled

for neither of the vertices v_j) or for the following condition to be satisfied $\text{dist}_F(v_r, v_i) \neq \text{dist}_{F'}(v_r, v_j)$. And we can see that this is only fulfilled for $j = 5$ and not for $j = 8$.

We do not need to check the rest of the conditions as the second condition is not met, which gives us contradiction in the values of p_i and therefore $\omega(\emptyset, p_1, q_1, true, false) = +\infty$ for any $p_1, q_1 \in \{true, false\}$ and the algorithm again yields a wrong result.

We do have a proposition how to fix this issue, however we do not provide any proof, and therefore any guarantee that the algorithms works, since proving this proposition would be far outside the scope of this thesis.

The only statement we can confidently say is that the following propositions fix the described issues for graphs with modular decomposition tree of maximum depth 2. However, we cannot say so in general.

First we need to solve the issue of the ambiguity of computation of the function w. We do that by specifying the correct quantifier in the following proposition.

▶ **Proposition 7.6.** Let $V(H)$ be partitioned into $s \leq t$ non-empty modules $X_1, ..., X_s$, $s \geq 2$. We assume that $X_1, ..., X_h$ are *trivial*, this means that $|X_i| = 1$ for $i \in \{1, ..., h\}$ where $0 \leq h \leq s$. For distinct $i, j \in \{1, ..., s\}$, either vertex of X_i is adjacent to every vertex of X_j or the vertices of X_i and X_j are not adjacent. Let F be the *prime* graph with a vertex set $\{v_1, ..., v_s\}$ such that v_i is adjacent to v_j if and only if the vertices of X_i are adjacent to the vertices of X_j. Let F' be obtained by addition of a universal vertex to the graph F. Observe that if $x \in X_i$ and $y \in X_j$ for distinct $i, j \in \{1, ..., s\}$, then $\text{dist}_{H'}(x, y) = \text{dist}_{F'}(v_i, v_j)$.

For boolean variables p, q, a set of indices $I \subseteq \{1, ..., h\}$, and boolean variables p_i, q_i where $i \in \{h+1, ..., s\}$ we define

$$\omega(p, q, I, p_{h+1}, q_{h+1}, ..., p_s, q_s) = |I| + \sum_{i=h+1}^{s} w(H[X_i], p_i, q_i)$$

if the following holds:

1. the set $Z = \{v_i \mid i \in I \cup \{h+1, ..., s\}\}$ resolves $V(F)$ in F',
2. if $p_i = true$ for some $i \in \{h+1, ..., s\}$, then for each $j \in \{1, ..., h\} \setminus I$, $\text{dist}_{F'}(v_i, v_j) = 2$ or there is $v_r \in Z$ such that $r \neq i, j$ and $\text{dist}_{F'}(v_r, v_i) \neq \text{dist}_{F'}(v_r, v_j)$,
3. if $q_i = true$ for some $i \in \{h+1, ..., s\}$, then for each $j \in \{1, ..., h\} \setminus I$, $\text{dist}_{F'}(v_i, v_j) = 1$ or there is $v_r \in Z$ such that $r \neq i, j$ and $\text{dist}_{F'}(v_r, v_i) \neq \text{dist}_{F'}(v_r, v_j)$,
4. if $p_i = p_j = true$ for some distinct $i, j \in \{h+1, ..., s\}$, then $\text{dist}_{F'}(v_i, v_j) = 2$ or there is $v_r \in Z$ such that $r \neq i, j$ and $\text{dist}_{F'}(v_r, v_i) \neq \text{dist}_{F'}(v_r, v_j)$,
5. if $q_i = q_j = true$ for some distinct $i, j \in \{h+1, ..., s\}$, then $\text{dist}_{F'}(v_i, v_j) = 1$ or there is $v_r \in Z$ such that $r \neq i, j$ and $\text{dist}_{F'}(v_r, v_i) \neq \text{dist}_{F'}(v_r, v_j)$,
6. $p = true$ if and only if there is $i \in \{1, ..., h\} \setminus I$ such that $\text{dist}_{F'}(v_i, v_j) = 1$ **for all** $v_j \in Z$ or there is $i \in \{h+1, ..., s\}$ such that $p_i = true$ and $\text{dist}_{F'}(v_i, v_j) = 1$ **for all** $v_j \in Z \setminus \{v_i\}$,
7. $q = true$ if and only if there is $i \in \{1, ..., h\} \setminus I$ such that $\text{dist}_{F'}(v_i, v_j) = 2$ **for all** $v_j \in Z$ or there is $i \in \{h+1, ..., s\}$ such that $q_i = true$ and $\text{dist}_{F'}(v_i, v_j) = 2$ **for all** $v_j \in Z \setminus \{v_i\}$,

and $\omega(p, q, I, p_{h+1}, q_{h+1}, ..., p_s, q_s) = +\infty$ in other cases.

Now we propose a solution to the problem of contradicting values p_i, q_i in the final computation of $\text{md}(G)$ for a graph G.

▶ **Proposition 7.7.** Let $V(H)$ be partitioned into $s \leq t$ non-empty modules $X_1, ..., X_s,\ s \geq 2$. We assume that $X_1, ..., X_h$ are *trivial*, this means that $|X_i| = 1$ for $i \in \{1, ..., h\}$ where $0 \leq h \leq s$. Let F be the *prime* graph with a vertex set $\{v_1, ..., v_s\}$ such that v_i is adjacent to v_j if and only if the vertices of X_i are adjacent to the vertices of X_j. Let F' be obtained by addition of a universal vertex to the graph F. Observe that if $x \in X_i$ and $y \in X_j$ for distinct $i, j \in \{1, ..., s\}$, then $\text{dist}_G(x, y) = \text{dist}_F(v_i, v_j)$.

For a set of indices $I \subseteq \{1, ..., h\}$ and boolean variables p_i, q_i where $i \in \{h+1, ..., s\}$, we define

$$\omega(I, p_{h+1}, q_{h+1}, ..., p_s, q_s) = |I| + \sum_{i=h+1}^{s} w(G[X_i], p_i, q_i)$$

if the following holds:

1. The set $Z = \{v_i \mid i \in I \cup \{h+1, ..., s\}\}$ is a resolving set for F,
2. If $p_i = true$ for some $i \in \{h+1, ..., s\}$, then for each $j \in \{1, ..., h\} \setminus I$, $\text{dist}_F(v_i, v_j) \geq 2$ or there is $v_r \in Z$ such that $r \neq i, j$ and $\text{dist}_F(v_r, v_i) \neq \text{dist}_F(v_r, v_j)$,
3. If $q_i = true$ for some $i \in \{h+1, ..., s\}$, then for each $j \in \{1, ..., h\} \setminus I$, $\text{dist}_F(v_i, v_j) \neq 2$ or there is $v_r \in Z$ such that $r \neq i, j$ and $\text{dist}_F(v_r, v_i) \neq \text{dist}_F(v_r, v_j)$,
4. $p_i = p_j = true$ for some distinct $i, j \in \{h+1, ..., s\}$, then $\text{dist}_F(v_i, v_j) \geq 2$ or there is some $v_r \in Z$ such that $r \neq i, j$ and $\text{dist}_F(v_r, v_i) = \text{dist}_F(v_r, v_j)$,
5. $q_i = q_j = true$ for some distinct $i, j \in \{h+1, ..., s\}$, then $\text{dist}_F(v_i, v_j) \neq 2$ or there is some $v_r \in Z$ such that $r \neq i, j$ and $\text{dist}_F(v_r, v_i) = \text{dist}_F(v_r, v_j)$.

and $\omega(I, p_{h+1}, q_{h+1}, ..., p_s, q_s) = +\infty$ in all other cases.

Algorithms For Truncated Metric Dimension

In this chapter, we elaborate as to why the algorithm bounded by max-degree and tree-length is not suitable to be converted to solve Truncated Metric Dimension*. Since we found the algorithm bounded by modular-width not correct, it does not make sense to consider any modifications to it.*

8 Max-degree and Tree-length

In this section we provide an argument as to why the algorithm bounded by max-degree and tree-length is not suitable for conversion to solve the truncated metric dimension problem. The issue lies within the Lemma 6.4.

More specifically, if we restate the Lemma 6.3 using the k-truncated distance definition, we get that:

▶ **Proposition 8.1.** Let G be a connected graph with max-degree $\Delta(G) = \Delta$ and let (X, T), where $X = \{X_i \mid i \in V(T)\}$, be a nice tree decomposition of G of length at most l and k a positive integer. Then for every $i, j \in V(T)$ and any $x \in X_i, y \in X_j$ the following holds:

$$\operatorname{dist}_T(i, j) \leq \alpha(\Delta, l)(\operatorname{dist}_{G,k}(x, y) + 1) - 1.$$

Counterexample. Let $x \in X_i$ and $y \in X_j$, for $i, j \in V(T)$. Let R be the shortest (x, y)-path in G, and let P be the unique (i, j)-path in T. For any vertex $h \in V(P)$, X_h contains at least one vertex of R. And since any vertex z from R is included in at most $\alpha(\Delta, l)$ bags X_h for $h \in V(P)$, $|V(P)| \leq \alpha(\Delta, l)|V(R)|$ (By Lemma 6.2) and by rearranging the equation, we get $\frac{|V(P)|}{(\Delta,l)} \leq \alpha|V(R)|$. But since we are using the truncated distance function, it is not true, that the distance of x and y is $|V(R)| - 1$. Therefore we cannot rely on the following inequality $\frac{\operatorname{dist}_T(i,j)}{\alpha(\Delta,l)} \leq \min(\operatorname{dist}_G(x, y), k+1)+1$, since if we choose k to be less than $\frac{\operatorname{dist}_T(i,j)}{\alpha(\Delta,l)} - 2$, inequality does not hold and therefore the lemma does not hold. ◀

And now after restating the Locality Lemma 6.4 and its proof, we can see multiple points of failure.

▶ **Proposition 8.2.** Let (X, T), where $X = \{X_i \mid i \in V(T)\}$, be a nice tree decomposition of G of length at most l such that T is rooted in $r, X_r = \{u\}$. Let $\Delta = \Delta(G)$ be the max-degree of G and let $s = \alpha(\Delta, l)(2l + 1)$. Then the following holds:

1. If $i \in V(G)$ is an introduce node with the child i' and v is the unique vertex of $X_i \setminus X_{i'}$ then for any $x \in V(G_j)$ for a node $j \in V(T_i)$ such that $\operatorname{dist}_T(i, j) \geq s$, u resolves v and x.

2. If $i \in V(G)$ is a join node with the children i', i'' and $x \in V(G_j) \setminus X_j$ for $j \in T_{i'}$ such that $dist_T(i', j) \geq s-1$ and $y \in V(G_{i''}) \setminus X_{i''}$ then u or an arbitrary vertex $v \in (V(G_j) \setminus X_j)$ resolves x and y.

Disproof. As for the first claim, let us consider $x \in V(G_j)$ for some $j \in V(T_{i'})$ such that $\mathrm{dist}_T(i', j') \geq s$. Now it is stated that the either $u \in X_i$ or u is separated from x by X_i,

$$\mathrm{dist}_{G,k}(u, x) = \min\left\{\mathrm{dist}_{G,k}(u, y) + \mathrm{dist}_{G,k}(y, z) + \mathrm{dist}_{G,k}(z, x) \mid y \in X_i, z \in X_j\right\}.$$

This is obviously not a true statement, since the triangle inequality does not hold when using the truncated distance function.

Next, there is an observation, that when the Lemma 6.3 is used, another inequality holds. However since we disproved the Lemma 6.3 for usage with truncated distance function, the inequality also does also not hold.

Similar observations can be made for the second statement of the lemma. ◀

As seen in the propositions above, the fact that the k-truncated version of the Lemma 6.3 is not true means that the proof for the main structural lemma [2], as it is called by the authors, Lemma 6.4 does not hold either. And as the correctness of the algorithm, for each type of node, relies on the Lemma 6.4 we claim that this algorithm is unfit for modification to solve the truncated version of the metric dimension problem.

9 Modular-width

Since we found that the algorithm bounded by modular-width is not working correctly and we did not prove the proposed change to be correct, we have no algorithm bounded by modular-widtrh to alter.

Implementation and Testing

10 Implementation

In this chapter we describe the implementation of the algorithms for the generation of the data set and the metric dimension bounded by modular-width. We will also present the measured results.

The language Python with the SageMath framework was chosen based on many factors. Primarily it was the built-in algorithms for modular decomposition and other operations with graphs, while being very easy to use. The fact that SageMath provides reasonable performance was also a factor. Additionally Python and SageMath are popular tools among the scientific community, which means the interpretation of our implementation should be less of a problem than with less common languages.

10.1 Data Generator

The data generator is a simple random modular decomposition generator. The generating function accepts two parameters. The modular-width t and the maximal depth of the modular decomposition tree d. Built-in SageMath function is then used to generate a graph from the modular decomposition.

The reasoning behind generating a modular-width decomposition as opposed the generating a graph and then calculating its modular decomposition is that the we can better test the running time of the algorithm as the dominating determining factor of the running time is the maximal and average width of a module. It is important to mention that we chose the root module so it can always be partitioned into exactly t modules, where the modular-width of each of the sub-modules is less or equal to t. This decision was made to ensure that the modular-width is t and the the graph is connected. A leaf node always has to be Normal node. All the other nodes are chosen randomly using uniform distribution of four choices

1. Normal node, meaning a single vertex,
2. Prime node, meaning a module that has at minimum four and at maximum t sub-modules,
3. Parallel node, meaning a disjoint union of modules,
4. Series node, meaning a complete join of modules.

One might argue that generating multiple date sets with slightly different probabilities of each of the nodes might be useful. For example with the probability of Prime node set higher and compensate for it with making the probability of the Normal node smaller, however because of the performance of the algorithm, we are generating graphs so small, that we don't find that this change would have any interesting results.

10.2 Metric Dimension Algorithm

The generated graph and its modular decomposition are passed to functions `md_original`, `md_modified`, `md_naive`. The functions then compute the metric dimension according to

- the algorithm by Belmonte et al.
- the algorithm by Belmonte et al with our proposed changes,
- an algorithm, that tests all the possible subsets of vertices of the graph,

respectively. Each of the results is then printed out.

11 Measured Results

Table 1 Performance of the algorithm

t	# non-trivial modules	z	$\mid V(G) \mid$	md(G)	time [s]
4	2	4	7	2	0.0
4	4	3.7	11	7	0.1
4	4	3.7	11	5	0.2
4	4	4.2	13	8	0.3
4	5	4.0	16	10	0.8
4	8	1.3	17	7	8.9
4	11	3.6	29	16	66.6
4	13	3.0	27	15	111.8
4	13	3.3	30	19	167.1
4	14	3.7	39	21	200.3
6	5	5.4	22	13	14.3
6	6	3.7	21	11	12.2
6	6	4.1	20	10	20.5
6	7	5.0	31	20	55.2
6	7	5.1	30	15	77.0
6	10	4.3	32	21	2128.9
6	10	5.3	43	24	2427.1
6	15	4.5	53	35	18023.0
6	17	4.4	58	42	17069.7
6	24	4.7	89	68	13920.9
8	5	5.4	22	10	90.2
8	6	6.1	31	16	455.7
8	6	6.6	34	26	814.8
8	7	5.2	30	14	1273.0
8	7	5.8	34	26	1091.4

t modular-width

z average size of a non-trivial module

Note: Each graph was tested 5 times and the running times were averaged.

11.1 Performance

First we shall focus on performance of the algorithm. Since our proposition does not change the algorithm in any significant way, from the running time point of view, it does not matter that we chose to measure the running time of the unmodified algorithm. We chose to measure the performance of the `md_original` function.

The test was done on a computer with an Intel i7-8700 CPU, with 32 GB of RAM. All the input data can be found in the `data` folder, where the the data are sorted into folders. Primarily by the modular-width parameter and secondarily by the depth of the tree.

Since the complexity of the algorithm depends on the maximal size of any module in the modular decomposition and number of vertices, we have generated data with relatively small modular-width and maximal depth of modular decomposition tree, limiting the maximal number of vertices, otherwise the computation would take unreasonable amount of time to finish. After some experimentation we decided 4, 6 and 8 are reasonable values of the modular-width for testing.

While the worst case complexity has the upper bound of $O(4^t \cdot t^3 \cdot n + m)$, where t is the module size, n the number of vertices, and m the number of edges, that does not tell much about the average time complexity of the algorithm. While we will not prove such bound, we shall present Table 1, where we have chosen three important metrics, from which we can approximate the running time much better. Those are the modular-width, the number of non-trivial modules and the average size of non-trivial module of the graph.

This is the case, because in the computation of each of the modules, there are three main components that add to the running time:

1. 2^k values are generated and tested (2 values for each of the non-trivial modules), where k is the number of non-trivial modules,
2. all the subsets of the trivial modules are tested,
3. a table of distances in the sub-graph for the module is computed.

From this simple observation one should be able to see why we chose these metrics. We can also see that the number of edges is not very important, so we chose to omit it.

We emphasize that these are just approximations, as it can happen that the final set is found early in the computation leading to cutting some of the computation branches, or that some of the graphs may favour a better computation branch cutting due to the conditions not being satisfied. We can observe this for example for the very bottom of the computations of modular-width 6 and 8.

We conclude that the algorithm performs within our expectations.

Conclusion

The goals of this thesis were to research the metric dimension problem and already known FPT algorithms, find out whether it is possible for some of these algorithms to also solve the truncated version of the metric dimension problem with minimal modifications to the algorithm itself and to implement such algorithm.

We got familiar with the concept of the metric dimension problem and its truncated variant. We also got familiar with concepts of parameterized complexity and various structural parameters. We have shown that for one of the algorithms by Belmonte et al.[2] it is not possible to convert it in a suitable way to solve the truncated metric dimension problem and for the other one we presented counterexamples to its correctness.

Then the algorithm bounded by modular-width was implemented in Python with the support of SageMath libraries and its performance was evaluated.

12 Possible Improvements

There are multiple possible ways to iterate on this thesis. Other already known algorithms bounded by other structural parameters could be considered. For example, an algorithm solving the metric dimension problem with linear running time with respect to tree-width is known. Then, of course, the algorithm bounded by modular-width could be fixed, if possible. Both of the algorithms could then be re-implemented in a more performant language and added into SageMath or packaged separately.

Bibliography

1. YERO, Ismael González; ESTRADA-MORENO, Alejandro; RODRÍGUEZ-VELÁZQUEZ, Juan A. Computing the k-metric dimension of graphs. *Appl. Math. Comput.* 2017, vol. 300, pp. 60–69. Available from DOI: 10.1016/j.amc.2016.12.005.
2. BELMONTE, Rémy; FOMIN, Fedor V.; GOLOVACH, Petr A.; RAMANUJAN, M. S. Metric Dimension of Bounded Tree-length Graphs. *SIAM J. Discret. Math.* 2017, vol. 31, no. 2, pp. 1217–1243. Available from DOI: 10.1137/16M1057383.
3. GROSS, Jonathan L.; YELLEN, Jay (eds.). *Handbook of Graph Theory*. Chapman & Hall / Taylor & Francis, 2003. Discrete Mathematics and Its Applications. ISBN 978-1-58488-090-5. Available from DOI: 10.1201/9780203490204.
4. KNOP, Dušan; MALÍK, Josef; SUCHÝ, Ondřej; TVRDÍK, Pavel; VALLA, Tomáš. *Základy grafů* [online]. FIT CTU, 2022 [visited on 2023-05-08]. Available from: https://courses.fit.cvut.cz/BI-AG1/lectures/media/bi-ag1-p1-handout.pdf.
5. KNOP, Dušan; MALÍK, Josef; SUCHÝ, Ondřej; TVRDÍK, Pavel; VALLA, Tomáš. *Souvislost, složitost, stromy* [online]. FIT CTU, 2022 [visited on 2023-05-08]. Available from: https://courses.fit.cvut.cz/BI-AG1/lectures/media/bi-ag1-p2-handout.pdf.
6. FRONGILLO, Rafael M.; GENESON, Jesse; LLADSER, Manuel E.; TILLQUIST, Richard C.; YI, Eunjeong. Truncated metric dimension for finite graphs. *Discret. Appl. Math.* 2022, vol. 320, pp. 150–169. Available from DOI: 10.1016/j.dam.2022.04.021.
7. TEDDER, Marc; CORNEIL, Derek G.; HABIB, Michel; PAUL, Christophe. Simpler Linear-Time Modular Decomposition Via Recursive Factorizing Permutations. In: ACETO, Luca; DAMGÅRD, Ivan; GOLDBERG, Leslie Ann; HALLDÓRSSON, Magnús M.; INGÓLFSDÓTTIR, Anna; WALUKIEWICZ, Igor (eds.). *Automata, Languages and Programming, 35th International Colloquium, ICALP 2008, Reykjavik, Iceland, July 7-11, 2008, Proceedings, Part I: Tack A: Algorithms, Automata, Complexity, and Games*. Springer, 2008, vol. 5125, pp. 634–645. Lecture Notes in Computer Science. Available from DOI: 10.1007/978-3-540-70575-8_52.
8. DOURISBOURE, Yon; GAVOILLE, Cyril. Tree-decompositions with bags of small diameter. *Discret. Math.* 2007, vol. 307, no. 16, pp. 2008–2029. Available from DOI: 10.1016/j.disc.2005.12.060.
9. KLOKS, Ton. *Treewidth, Computations and Approximations*. Vol. 842. Springer, 1994. Lecture Notes in Computer Science. ISBN 3-540-58356-4. Available from DOI: 10.1007/BFb0045375.

10. CYGAN, Marek; FOMIN, Fedor V.; KOWALIK, Lukasz; LOKSHTANOV, Daniel; MARX, Dániel; PILIPCZUK, Marcin; PILIPCZUK, Michal; SAURABH, Saket. *Parameterized Algorithms*. Springer, 2015. ISBN 978-3-319-21274-6. Available from DOI: 10.1007/978-3-319-21275-3.
11. SLATER, Peter. *Leaves of Trees*. Utilitas Mathematica Pub, 1975. Proceedings of the Southeastern Conference on Combinatorics, Graph Theory, and Computing, Congressus numerantium, no. 14.
12. HARARY, Frank; MELTER, Robert. *On the Metric Dimension of a Graph*. The Charles Babbage Research Centre, 1976. Ars Combinatoria, no. 2.
13. KHULLER, Samir; RAGHAVACHARI, Balaji; ROSENFELD, Azriel. Landmarks in Graphs. *Discret. Appl. Math.* 1996, vol. 70, no. 3, pp. 217–229. Available from DOI: 10.1016/0166-218X(95)00106-2.
14. KRATICA, Jozef; KOVACEVIC-VUJCIC, Vera; CANGALOVIC, Mirjana. Computing the metric dimension of graphs by genetic algorithms. *Comput. Optim. Appl.* 2009, vol. 44, no. 2, pp. 343–361. Available from DOI: 10.1007/s10589-007-9154-5.
15. MLADENOVIC, Nenad; KRATICA, Jozef; KOVACEVIC-VUJCIC, Vera; CANGALOVIC, Mirjana. Variable neighborhood search for metric dimension and minimal doubly resolving set problems. *Eur. J. Oper. Res.* 2012, vol. 220, no. 2, pp. 328–337. Available from DOI: 10.1016/j.ejor.2012.02.019.
16. HAUPTMANN, Mathias; SCHMIED, Richard; VIEHMANN, Claus. Approximation complexity of Metric Dimension problem. *J. Discrete Algorithms*. 2012, vol. 14, pp. 214–222. Available from DOI: 10.1016/j.jda.2011.12.010.
17. SPINELLI, Brunella; CELIS, L. Elisa; THIRAN, Patrick. The effect of transmission variance on observer placement for source-localization. *Appl. Netw. Sci.* 2017, vol. 2, p. 20. Available from DOI: 10.1007/s41109-017-0040-5.
18. PINTO, Pedro C.; THIRAN, Patrick; VETTERLI, Martin. Locating the Source of Diffusion in Large-Scale Networks. *CoRR*. 2012, vol. abs/1208.2534. Available from arXiv: 1208.2534.

Contents of the supplied medium

```
readme.txt ................ tutorial for running the program
src
├── impl .................. source code of the implementation
│   └── main.py ........... implementation of the algorithm
├── test_data ............. data set
│   ├── t4 ................ data set with module-width 4
│   │   ├── d2 ............ data set with max tree depth 2
│   │   └── d2 ............ data set with max tree depth 3
│   ├── t6 ................ data set with module-width 6
│   │   ├── d2 ............ data set with max tree depth 2
│   │   └── d2 ............ data set with max tree depth 3
│   └── t8 ................ data set with module-width 8
│       └── d2 ............ data set with max tree depth 2
└── thesis ................ source code of the thesis LaTeX
text ...................... the thesis
└── thesis.pdf ............ the thesis in PDF format
```