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.
May 16, 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$\sim$, 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 May 16, 2024 ....

Abstract

This thesis is directed at FPT algorithms that solve the Truncated Metric Dimension problem. Two already known algorithms solving the Metric Dimension problem are described. For the algorithm bounded by modular-width we present a simple modification for it to solve the Truncated Metric Dimension problem. As for the algorithm bounded by max leaf number, we describe the reason that prevented us from modifying the algorithm. Following that, we explain how we implemented the algorithms bounded by modular-width and show selected performance metrics.

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

Abstrakt

Tato práce se zabývá FPT algoritmy řešící problém Zkrácené Metrické Dimenze. Představíme dva již známé algoritmy řešící problém Metrické Dimenze. Pro algoritmus parametrizovaný šířkou modulu ukážeme jeho jednoduchou modifikaci tak, aby řešil problém Zkrácené Metrické Dimenze. Pro algoritmus parametrizovaný maximálním počtem listů vysvětlíme důvod, který nám zabránil jeho úpravám. Poté nastíníme implementaci algoritmů parametrizovaných šířkou modulu a ukážeme vybrané výkonnostní metriky.

Klíčová slova Metrická Dimenze, Zkrácená Metrická Dimenze, Rozlišující množina, Parametrizované algoritmy, Složitost

List of Acronyms

md	metric dimension
md_k	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 (md_l) problem poses the same question with the simple modification that we only consider vertices from the set, whose distance is at maximum l 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 Metric Dimension with respect to various structural parameters and, if possible alter them in such a way that they then compute the solution of Truncated Metric Dimension. In this thesis we focus on an algorithm parameterized by modular-width that was proposed in an article by Belmonte et al. [2]. We propose a modification to the algorithm and show that the modified algorithms solves Truncated Metric Dimension. The modified algorithm was implemented and tested on appropriate data set. We also explain the difficulties that occur when trying to modify the algorithm parameterized by max leaf number by David Eppstein [3].

Preliminaries

1 Graph Theory

First, we shall start by defining a graph and related concepts.

▶ **Definition 1.1** (Graph, Inspired by [4])**.** 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 sets 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 $N_G^r[v] = \{u \in V \mid \mathrm{dist}_G(u, v) \leq r\}$ 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 the 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. We denote graph *induced* by the set $U \subseteq V$, as $G[U]$. In other words $G[U] = G - (V(G) \setminus U)$.

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

Following the definition of a graph, we define a path.

▶ **Definition 1.2** (Path, Inspired by [4])**.** 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. To do that we also need to define a cycle and a connected graph.

▶ **Definition 1.3** (Cycle [5])**.** 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 [6])**.** 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 [6])**.** A graph $G = (V, E)$ is a *tree* if the graph is connected and does not have a cycle as a subgraph. We call a vertex $v \in V(G)$ a *leaf* if $\deg_G(v) = 1$.

Further we define the distance in a graph and the diameter of a graph.

▶ **Definition 1.6** (Distance, l-truncated distance, Inspired by [7])**.** Distance between two vertices u and v in the graph G, denoted by $\mathrm{dist}_G(u, v)$, is the number of edges in a shortest u-v-path in the graph G.

- Let $\mathrm{dist}_{G,l}(u, v) = \min(\mathrm{dist}_G(u, v), l + 1)$ denote *l-truncated* distance.

- For a vertex $v \in V$ and a set $U \subseteq V$, let $\mathrm{dist}_G(v, U) = \min\{\mathrm{dist}_G(v, u) \mid u \in U\}$ be a minimal distance from a vertex v to any of the vertices from U.

When possible, we shorten $\mathrm{dist}_{G,l}(u, v)$ to $\mathrm{dist}_l(u, v)$.

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

In one of the algorithms we need to make use of a *universal* vertex.

▶ **Definition 1.8** (Universal vertex [2])**.** A vertex $v \in V$ is called *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 vertex set and $E(G_1) \cup E(G_2)$ as its edge set.

The 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 $E(G_1) \cup E(G_2) \cup \{uv \mid \forall u \in V_1, \forall v \in V_2\}$ as its edges.

Since we new know what a modular decomposition is we can define a *prime* graph.

For the next definition we need the definition of a modular decomposition from the section about complexity theory.

▶ **Definition 1.10** (Prime graph [2])**.** Let G be a graph partitioned into $s \leq t$ non-empty modules $X_1, ..., X_s,\ s \geq 2$. Let also F be a graph with the vertex set v_i foreach $i \in \{1, ..., s\}$. Any two distinct vertices $v_i, v_j \in F$ are adjacent if and only if the vertices of the modules $X_i and X_j$ are also adjacent in G for all $i, j \in \{1, ..., s\}$.

A graph constructed as described above is called the *prime* graph F of the graph G.

Modular-width can be computed in linear time by the algorithm of Tedder et al. [8].

2 Complexity Theory

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

▶ **Definition 2.1** (Parameterized problem, Cygan et al. [9]). Parameterized problem is a language $L \subseteq \Sigma \times N$, where Σ is a fixed finite alphabet. For an instance $(x, k) \in \Sigma^* \times \mathbb{N}$, k is called the *parameter*.

▶ **Definition 2.2** (FPT, Cygan et al. [9]). A parameterized problem $L \subseteq \Sigma^* \times N$ is called *fixed-parameter tractable* (*FPT*) if there exists an algorithm A, called *fixed-parameter tractable* algorithm, a computable function $f : N \to N$, 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.

In the following we specify two structural parameters *modular-width* and *max leaf number*, which we use as the parameter for the algorithms.

▶ **Definition 2.3** (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, $mw(G)$, 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 $\mathrm{mw}(G[X_i]) \leq t$ for all $i \in \{1, ..., s\}$.

Sometimes we use the term *trivial* module. That is a module containing only one vertex. We call the tree structure, where each node represents one of the above operations, a *modular decomposition*.

▶ **Definition 2.4** (Max leaf number [3]). The *max leaf number* of a connected graph G is the maximum, over all spanning trees of G, of the number of leaves in the spanning tree.

3 The Metric Dimension and Related 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** ([2]). Let $G = (V, E)$ be a graph. We say that the distinct vertices $u, v \in V$ are *resolved* by $w \in V$ if $\mathrm{dist}(u, w) \neq \mathrm{dist}(v, w)$. Similarly, the vertices $u, v \in V$ are *k-resolved* by $w \in V$ if $\mathrm{dist}_k(u, w) \neq \mathrm{dist}_k(v, w)$.

▶ **Definition 3.2** ([2]). Let $G = (V, E)$ be a graph. We say that the set of vertices $X \subseteq V$ is *resolved* by $w \in V$ if for each two distinct vertices $u, v \in X$ the vertex w resolves u and v. Similarly, the vertices are *k-resolved* by $w \in V$ if for each two distinct vertices $u, v \in X$ the vertex w k-resolves u and v.

▶ **Definition 3.3** (Resolving set [2])**.** Let $G = (V, E)$ be a graph and W and X be sets of vertices of G. We say that the set W is a *resolving set* for X if for all distinct $u, v \in X$ there is $w \in W$ such that w resolves u and v. Similarly, we say that the set W is a k-*resolving set* for X if for all distinct $u, v \in X$ there is $w \in W$ such that w k-resolves u and v.

▶ **Definition 3.4** (Metric Dimension [2])**.** Let $G = (V, E)$ be a graph. The *metric dimension* of G denoted by $\mathrm{md}(G)$ is the minimum cardinality over all resolving sets of G. Metric dimension asks if $\mathrm{md}(G) \leq k$, for a given positive integer k.

▶ **Definition 3.5** (Truncated Metric Dimension [7])**.** Let $G = (V, E)$ be a graph and let l be a positive integer. The *truncated metric dimension* of G denoted by $\mathrm{md}_l(G)$ is the minimum cardinality over all l-resolving sets of G. Truncated Metric Dimension asks, given a graph G and integers l and k, whether $\mathrm{md}_l(G) \leq k$.

Known Results

4 Metric Dimension

The notion of resolving sets was first independently introduced by Slater [10] and Harary with Melter [11] as a way of uniquely identifying all the vertices in a graph. In the year 1996 Khuller et al. [12] have in their paper on the topic shown that it is NP-hard to decide Metric Dimension on general graphs and that it can be solved in linear time on trees. They have also shown that the metric dimension can be approximated in a polynomial time within a factor of $O(\log n)$.

As a result of this finding many other algorithms approximating the metric dimension have been developed. Most notably a genetic algorithm that can approximate the metric dimension relatively quickly [13]. Whilst some of these algorithms find very small resolving sets, there is, as proven by Hauptmann et al. [14] in 2012, no algorithm that can guarantee $(1-\epsilon)\ln(n)$-approximation for any $\epsilon > 0$ unless NP is a subset of DTIME$(n^{\log\log n})$. In the same article they provide $(1+(1+o(1)\cdot\ln(n)))$-approximation, i.e. essentially the best possible approximation, algorithm with running time complexity of $O(|n|^3)$.

In 2009 Daniel Lokshtanov posed a question whether the metric dimension would be a suitable topic in the area of parameterized complexity. Moreover he conjectured that the problem would be W[1]-complete with respect to the solution size. Three years later it was shown, by Hartung et al. [15], that the problem is actually W[2]-complete [15] with respect to the size of the resolving set on graphs of maximum degree three. In the very same article the first parameterized algorithm (parameterized by vertex cover number) for the problem of metric dimension was published. Since then numerous parameterized algorithms have been developed. In 2015 David Eppstein [3] presented a parameterized algorithm bounded by max leaf number. An algorithm bounded by tree-length plus max-degree and an algorithm bounded by modular-width were published by Belmonte et al. [2] in 2016. By the year 2017 Diaz et at. [16] has shown that the problem is NP-complete for a planar graph that has maximum degree 6 and presented a polynomial time algorithm to solve the problem on outerplanar graphs. Bonnet et al. [17] have in the year 2021 shown that the metric dimension problem is W[1]-hard when parameterized by tree-width. Li et al. [18] have later strenghtened this result and shown that the problem is NP-hard for graphs of tree-width 24. Two years later, in 2023, Bousquet et al. [19] have shown an FPT algorithm parameterized by tree-width in chordal graphs. In the same year it was shown that problem is W[1]-hard when parameterized by the combined parameter feedback vertex set number plus path-width and FPT when parameterized either by the distance to cluster of the distance to co-cluster. Most recently, in 2024 Foucaud et al. [20] have published an algorithm parameterized by tree-width, based on the algorithm by Bousquet et al. [19].

Various bounds, exact results, characterizations of graphs and other properties have also been found over the last five decades, see, e.g., [12, 21, 22, 23, 24].

5 Truncated Metric Dimension

The notion of Truncated Metric Dimension has been established by Geneson and Yi [25] under the name broadcast dimension and was later expanded upon by Frongillo et al. [7]. The motivation for such 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 [7, 26, 27].

In comparison to Metric Dimension, not as much is known about the Truncated Metric Dimension.

Just like Metric Dimension, it has been shown that Truncated Metric Dimension is NP-hard by reduction from 3-SAT [12]. This should be obvious since we can set the parameter l 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. In this sense Truncated Metric Dimenion can be seen as a generalization of the Metric Dimension problem.

In the article by Frongillo et al. [7] characterizations and bounds for various types of graphs have been shown. It was also later claimed that computing Truncated Metric Dimension on trees is NP-hard for general k, but it can be done in polynomial time on trees for constant k [28].

No other algorithms were published about the topic, at least to our knowledge.

Algorithms Parameterized By Modular-width

6 Metric Dimension

In this section, we present an algorithm for the METRIC DIMENSION *problem that runs in linear time with respect to the modular-width.*

6.1 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 all vertices of X are the same. This is expressed by the next lemma.

▶ **Lemma 6.1** ([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 $x, y \in X$ in G 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.

The result is summarized in the following theorem.

▶ **Theorem 6.2** ([2])**.** The metric dimension of a connected graph G of modular-width at most t can be computed in time $O(t^3 4^t n + m)$.

To compute $\mathrm{md}(G)$, auxiliary values $w(H, p, q)$ for each of the sub-modules of the root module of the graph H are used. First, we define this function and right after that we explain it in a more intuitive manor.

▶ **Definition 6.3.** Let H be a graph 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 $w(H, p, q)$ is, the minimum size of set $W \subseteq V(H)$ such that

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

We assume that $w(H, p, q) = +\infty$ if such a 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 6.1. 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 in 1 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 $y \in G \setminus X$. If x is part of Z, then x resolves x and y. If x is at distance 1 from some $z \in Z'$ and there is $z' \in Z'$ such that z' is at distance 2 from x, then either z or z' resolves x and y, because $\text{dist}_G(z, y) = \text{dist}_G(z', y)$. Now let x be at distance 1 in G from every vertex in Z'. If $x' \in X$ is also at distance 1 from every vertex of Z', then $z \in Z$ resolves x and y if and only if z resolves x' in y. This is why its is sufficient to know why whether X has a vertex at distance 1 from every vertex of W. This is captured by the boolean variable p and set W in 2. The same argument is used for vertices at distance 2 from every vertex of Z' which is captured by 3.

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 size at most t we may compute the values for leaf nodes by brute force, followed by executing a bottom up dynamic programming algorithm.

In the original article [2] the algorithm to compute $w(H, p, q)$ is split into 3 cases.

- Graph H is a disjoint union of a 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.

While the first two cases are subsumed by the third case, we keep them in the text just like the original authors for clarity of the algorithm.

- **Disjoint union.** Let H be a disjoint union of H_1 and H_2. We 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 see that
  - $w(H, false, true) = 1$,
  - $w(W, false, false) = 2$,
  - $w(H, true, true) = w(true, false) = +\infty$.

  For the disjoint union we try to give an idea as to why are these equivalences true. We do not do so in other cases as the reasoning is very analogous.

  The first equivalence is true since we can choose either one of the vertices. Let us, without loss of generality, choose the vertex $x \in H_1$ and the vertex in H_2 be y. Then x resolves $V(H)$ in H' satisfying 1, and the vertex x is at the distance 2 from y satisfying 2–3 and $p = false$, $q = true$. Similarly, to fulfill all of the conditions in the second case, we have to choose both of the vertices. It is easy to see that p cannot be $true$, since the only two vertices in H are not adjacent.

  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 6.1

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

The idea in the first case is that because $q = true$ and $q_2 = false$, the vertex $x \in H_1$ has to be the only one for which $\text{dist}_{H'}(v, x) = 2$ for all $v \in W$, and, thus $V(H)$ will be resolved in H' and for every $y \in V(H_2)$ the distance between x and y will always be 2 in H', satisfying 3 when $q = true$ for H. As for the second case, if $p = false$ and $q = true$ we have 3 sub-cases that can occur. Either $p_2 = false$, $q_2 = false$ and the same logic as in the previous case applies. Or in the other two sub-cases $p_2 = true$ and we cannot identify the vertex in $V(H_1)$, so we have to add it to the resolving set. The fact that the conditions 2 and 3 hold is easy to verify. In the third case, where $p = false$ and $q = false$, we need to add the vertex from $V(H_1)$ to the resolving set since we have no way to distinguish it. The value q_2 has to be set to $false$, because if instead $q_2$was $true$ and we added $V(H_1)$ to the resolving set, it would imply that $q = true$. That would be a contradiction. And no other cases satisfy the conditions.

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 6.1

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

In this case we basically just try all the possible options. Only two things we need to be careful about is first if $q = false$ it cannot happen that $q_1 = q_2$, because that would be a contradiction and second that again p cannot be $true$ since there has to be at least one vertex in W from both H_1 and H_2 and for every $x \in H_1$ $\text{dist}_{H'}(x, V(H_2)) = 2$.

- **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 that

  - $w(H, true, false) = 1$,
  - $w(H, false, false) = 2$, 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 6.1

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

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 $y \in V(H_2)$, $\text{dist}_{H'}(x, y) = 1$ and that any resolving set has at least one vertex in both H_1 and H_2. By Lemma 6.1

- $w(H, true, true) = +\infty$,
- $w(H, false, true) = +\infty$,
- $w(H, true, false) = \min\{w(H_1, p_1, q_1) + w(H_2, p_2, q_2) \mid p_i, q_i \in \{true, false\},$ $i \in \{1, 2\}$ 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. We use the graph F' instead of F, when asking in whether Z resolves $V(F)$ or when calculating the distances, to emulate the fact that the distance between any two vertices can be at most 2. 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.

  The idea behind the conditions is that if, while computing the function ω for a module X_i, there might be a vertex x, which we track by the conditions 6 and 7, in X_i (or a whole another module) that we currently can uniquely identify, but in the next step of the computation (when computing ω for the parent of X_i), we might introduce another a module, say X_j, after which we would not be able to distinguish between some of the vertices of X_i and X_j. That is why we initially track these vertices and modules and later (during the computation of ω for the parent of X_i) check in the conditions 2 and 3 (4 and 5) if we are still able to uniquely identify each of these vertices (modules).

▶ **Claim 6.4** ([2]). The function w can be computed 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 $i \in \{h+1,...,s\}$.

Proof. First, we show that $w(H,p,q) \geq \min \omega(p,q,I,p_{h+1},q_{h+2},\ldots,p_s,q_s)$:

If $w(H,p,q) = +\infty$, then the inequality trivially holds. Let $w(H,p,q) < +\infty$. Then there is a set $W \subseteq V(H)$ of minimum size such that

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

By the definition, $w(H,p,q) = |W|$. Let $W_i = W \cap X_i$ for $i \in \{1,\ldots,s\}$. Let $I = \{i \mid i \in \{1,\ldots,h\}, W_i \neq \emptyset\}$. Notice that $W_i \neq \emptyset$ for $i \in \{h+1,\ldots,s\}$ by Lemma 6.1. For $i \in \{h+1,\ldots,s\}$, let $p_i = true$ if there is a vertex $x \in X_i$ such that $\text{dist}_{H'}(x,u) = 1$ for $u \in W_i$, and let $q_i = true$ if there is a vertex $x \in X_i$ such that $\text{dist}_{H'}(x,u) = 2$ every for $u \in W_i$.

By Lemma 6.1, W_i resolves X_i in the graph obtained from $H[X_i]$ by the addition of a universal vertex for $i \in \{h+1,\ldots,s\}$. Hence, $|W_i| \geq w(H[X_i],p_i,q_i)$ for $i \in \{h+1,\ldots,s\}$ and, therefore, $|W| \geq |I| + \sum_{i=h+1}^{s} w(H[X_i],p_i,q_i)$.

We show that all the conditions are fulfilled for I and the defined values of p_i, q_i.

1. Consider distinct vertices v_i, v_j of F. If $v_i \in Z$ or $v_j \in Z$, then it is straightforward to see that Z resolves v_i and v_j. Suppose that $i,j \in \{1,\ldots,h\} \setminus I$. Then X_i, X_j are trivial modules with the unique vertices x and y, respectively. Because W resolves $V(H)$, there is $u \in W$ such that $\text{dist}_{H'}(u,x) \neq \text{dist}_{H'}(u,y)$. Consider the set W_r containing u. It remains to observe that v_r resolves v_i and v_j, because $\text{dist}_{F'}(v_r,v_i) = \text{dist}_{H'}(u,x) \neq \text{dist}_{H'}(u,y) = \text{dist}_{F'}(v_r,v_j)$.

2. Assume that $p_i = true$ for some $i \in \{h+1,\ldots,s\}$ and consider $j \in \{1,\ldots,h\} \setminus I$. Suppose that $\text{dist}_{F'}(v_i,v_j) = 1$. Then X_i has 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 and, therefore, there is $u \in W$ such that $\text{dist}_{H'}(u,x) \neq \text{dist}_{H'}(u,y)$. If $u \in X_i$, then we have that $\text{dist}_{H'}(u,x) = 1 = \text{dist}_{F'}(v_i,v_j) = \text{dist}_{H'}(u,y)$ which is a contradiction. Hence, $u \notin X_i$. Let X_r be the module containing u. Then we have that $\text{dist}_{F'}(v_r,v_i) = \text{dist}_{H'}(u,x) \neq \text{dist}_{H'}(u,y) = \text{dist}_{F'}(v_r,v_i)$.

3. Assume that $q_i = true$ for some $i \in \{h+1,\ldots,s\}$ and consider $j \in \{1,\ldots,h\} \setminus I$. Suppose that $\text{dist}_{F'}(v_i,v_j) = 2$. Then X_i has 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 and, therefore, there is $u \in W$ such that $\text{dist}_{H'}(u,x) \neq \text{dist}_{H'}(u,y)$. If $u \in X_i$, then we have that $\text{dist}_{H'}(u,x) = 2 = \text{dist}_{F'}(v_i,v_j) = \text{dist}_{H'}(u,y)$ which is a contradiction. Hence, $u \notin X_i$. Let X_r be the module containing u. Then we have that $\text{dist}_{F'}(v_r,v_i) = \text{dist}_{H'}(u,x) \neq \text{dist}_{H'}(u,y) = \text{dist}_{F'}(v_r,v_i)$.

4. Suppose that $p_i = p_j = true$ for some distinct $i,j \in \{h+1,\ldots,s\}$ and assume that $\text{dist}_{F'}(v_i,v_j) = 1$. Then X_i has a vertex x adjacent to all the vertices of W_i and X_j has a vertex y adjacent to all the vertices of W_j. The set W resolves x,y and, therefore, there is $u \in W$ such that $\text{dist}_{H'}(u,x) \neq \text{dist}_{H'}(u,y)$. If $u \in X_i$, then we have that $\text{dist}_{H'}(u,x) = \text{dist}_{F'}(v_i,v_j) = \text{dist}_{H'}(u,y)$ which is a contradiction. Hence, $u \notin X_i$. By the same arguments, $u \notin X_j$. Let X_r be the module containing u. Then we have that $\text{dist}_{F'}(v_r,v_i) = \text{dist}_{H'}(u,x) \neq \text{dist}_{H'}(u,y) = \text{dist}_{F'}(v_r,v_i)$.

5. Suppose that $q_i = q_j = true$ for some distinct $i,j \in \{h+1,\dots,s\}$ 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 at distance 2 to all the vertices of W_j. The set W resolves x, y and, therefore, there is $u \in W$ such that $\mathrm{dist}_{H'}(u,x) \neq \mathrm{dist}_{H'}(u,y)$. If $u \in X_i$, then we have that $\mathrm{dist}_{H'}(u,x) = \mathrm{dist}_{F'}(v_i,v_j) = \mathrm{dist}_{H'}(u,y)$ which is a contradiction. Hence, $u \notin X_i$. By the same arguments, $u \notin X_j$. Let X_r be the module containing u. Then we have that $\mathrm{dist}_{F'}(v_r,v_i) = \mathrm{dist}_{H'}(u,x) \neq \mathrm{dist}_{H'}(u,y) = \mathrm{dist}_{F'}(v_r,v_i)$.

6. Recall that $p = true$ if and only if $V(H)$ has a vertex x that is adjacent to all the vertices of W. Suppose that $V(H)$ has a vertex x that is adjacent to all the vertices of W. If $x \in X_i$ for some $i \in \{1,\dots,h\} \setminus I$, then $\mathrm{dist}_{F'}(v_i,v_j) = 1$ for all $v_j \in Z$. If $x \in X_i$ for some $i \in \{h+1,\dots,s\}$, then $p_i = true$ and $\mathrm{dist}_{F'}(v_i,v_j) = 1$ for every $v_j \in Z \setminus \{v_i\}$. Suppose that there is $i \in \{1,\dots,h\} \setminus I$ such that $\mathrm{dist}_{F'}(v_i,v_j) = 1$ for every $v_j \in Z$ or there is $i \in \{h+1,\dots,s\}$ such that $p_i = true$ and $\mathrm{dist}_{F'}(v_i,v_j) = 1$ for every $v_j \in Z \setminus \{v_i\}$. If there is $i \in \{1,\dots,h\} \setminus I$ such that $\mathrm{dist}_{F'}(v_i,v_j) = 1$ for every $v_j \in Z$, then the unique vertex x of X_i is at distance 1 from all the vertices of W and $p = true$. If there is $i \in \{h+1,\dots,s\}$ such that $p_i = true$, then there is $x \in X_i$ at distance 1 from each vertex of W_i. If $\mathrm{dist}_{F'}(v_i,v_j) = 1$ for every $v_j \in Z \setminus \{v_i\}$, then x is at distance 1 from all the vertices $W \setminus W_i$ and, therefore, $p = true$.

7. Recall that $q = true$ if and only if $V(H)$ has a vertex x that is at distance 2 from every vertex of W. Suppose that $V(H)$ has a vertex x that is at distance 2 from all the vertices of W. If $x \in X_i$ for some $i \in \{1,\dots,h\} \setminus I$, then $\mathrm{dist}_{F'}(v_i,v_j) = 2$ for every $v_j \in Z$. If $x \in X_i$ for some $i \in \{h+1,\dots,s\}$, then $q_i = true$ and $\mathrm{dist}_{F'}(v_i,v_j) = 2$ for every $v_j \in Z \setminus \{v_i\}$. Suppose that there is $i \in \{1,\dots,h\} \setminus I$ such that $\mathrm{dist}_{F'}(v_i,v_j) = 2$ for $v_j \in Z$ or there is $i \in \{h+1,\dots,s\}$ such that $q_i = true$ and $\mathrm{dist}_{F'}(v_i,v_j) = 2$ for $v_j \in Z \setminus \{v_i\}$. If there is $i \in \{1,\dots,h\} \setminus I$ such that $\mathrm{dist}_{F'}(v_i,v_j) = 2$ for every $v_j \in Z$, then the unique vertex x of X_i is at distance 2 from all the vertices of W and $q = true$. If there is $i \in \{h+1,\dots,s\}$ such that $q_i = true$, then there is $x \in X_i$ at distance 2 from each vertex of W_i. If $\mathrm{dist}_{F'}(v_i,v_j) = 2$ for every $v_j \in Z \setminus \{v_i\}$, then x is at distance 2 from the vertices $W \setminus W_i$ and, therefore, $q = true$.

Because all of the seven conditions are fulfilled

$$w(H,p,q) = |W| \geq |I| + \sum_{i=h+1}^{s} w(H[X_i],p_i,q_i) = \omega(p,q,I,p_{h+1},q_{h+2},\dots,p_s,q_s).$$

Now we show that $w(H,p,q) \leq \min \omega(p,q,I,p_{h+1},q_{h+2},\dots,p_s,q_s)$. Assume that I and the values of $p_{h+1},q_{h+1},\dots,p_s,q_s$ are chosen in such a way that $\omega(p,q,I,p_{h+1},q_{h+2},\dots,p_s,q_s)$ has the minimum possible value. If $\omega(p,q,I,p_{h+1},q_{h+1},\dots,p_s,q_s) = +\infty$, then, trivially, we have that $w(H,p,q) \leq \omega(p,q,I,p_{h+1},q_{h+1},\dots,p_s,q_s)$. Suppose that $\omega(p,q,I,p_{h+1},q_{h+1},\dots,p_s,q_s)$ is finite. Then $\omega(p,q,I,p_{h+1},q_{h+1},\dots,p_s,q_s) = |I| + \sum_{i=h+1}^{s} w(H[X_i],p_i,q_i)$ and all the conditions are fulfilled for p, q, I and the values of $p_{h+1},q_{h+1},\dots,p_s,q_s$.

Notice that $w(H[X_i],p_i,q_i) < +\infty$ for $i \in \{h+1,\dots,s\}$. For $i \in \{h+1,\dots,s\}$, let $W_i \subseteq X_i$ be a set of minimum size such that

- W_i resolves X_i in the graph H'_i obtained from $H[X_i]$ by the addition of a universal vertex,
- 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 $p_i = true$, and
- 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, $w(H[X_i],p_i,q_i) = |W_i|$ for $i \in \{h+1,\dots,s\}$. Let

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

We have that $|W| = \omega(p,q,I,p_{h+1},q_{h+2},\dots,p_s,q_s)$.

▶ **Claim 6.5.** *W is a resolving set for $V(H)$ in H'.*

Proof. Let x, y be distinct vertices of H. We show that there is a vertex u in W that resolves x and y in H'. Clearly, it is sufficient to prove it for $x, y \in V(H) \setminus W$. Let X_i and X_j be the modules that contain x and 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 H'. Suppose that $i \neq j$.

Assume first that $i, j \in \{1, \ldots, h\}$. Then $i, j \in \{1, \ldots, h\} \setminus I$, because $X_1, \ldots, X_h$ are trivial. By 1, Z resolves $V(F)$ in F'. Hence, there is $v_r \in Z$ such that $\mathrm{dist}_{F'}(v_r, v_i) \neq \mathrm{dist}_{F'}(v_r, v_j)$. Notice that $W_r \neq \emptyset$ by the definition of W_r and Z. Let $u \in W_r$. We have that $\mathrm{dist}_{H'}(u, x) = \mathrm{dist}_{F'}(v_r, v_i) \neq \mathrm{dist}_{F'}(v_r, v_j) = \mathrm{dist}_{H'}(u, y)$.

Let now $i \in \{h+1, \ldots, s\}$ and $j \in \{1, \ldots, h\}$. If there are $u_1, u_2 \in W_i$ such that $\mathrm{dist}_{H'_i}(u_1, x) \neq \mathrm{dist}_{H'_i}(u_2, x)$, then u_1 or u_2 resolves x and y, because $\mathrm{dist}_{H'}(u_1, y) = \mathrm{dist}_{H'}(u_2, y)$. Assume that all the vertices of W_i are at the same distance from x in H'_i. Let $u \in W_i$. If $\mathrm{dist}_{H'_i}(u, x) = 1$, then $p_i = \mathit{true}$ and, by 2, $\mathrm{dist}_{F'}(v_i, v_j) = 2$ or there is $v_r \in Z$ such that $r \neq i, j$ and $\mathrm{dist}_{F'}(v_r, v_i) \neq \mathrm{dist}_{F'}(v_r, v_j)$. If $\mathrm{dist}_{F'}(v_i, v_j) = 2$, then u resolves x and y, as $\mathrm{dist}_{H'}(u, y) = 2$. Otherwise, any vertex $u' \in W_r$ resolves x and y. Similarly, if $\mathrm{dist}_{H'_i}(u, x) = 2$, then $q_i = \mathit{true}$ and, by 3, $\mathrm{dist}_{F'}(v_i, v_j) = 1$ or there is $v_r \in Z$ such that $r \neq i, j$ and $\mathrm{dist}_{F'}(v_r, v_i) \neq \mathrm{dist}_{F'}(v_r, v_j)$. If $\mathrm{dist}_{F'}(v_i, v_j) = 1$, then u resolves x and y, as $\mathrm{dist}_{H'}(u, y) = 1$. Otherwise, any vertex $u' \in W_r$ resolves x and y.

Finally, let $i, j \in \{h+1, \ldots, s\}$. If there are $u_1, u_2 \in W_i$ such that $\mathrm{dist}_{H'_i}(u_1, x) \neq \mathrm{dist}_{H'_i}(u_2, x)$, then u_1 or u_2 resolves x and y, because $\mathrm{dist}_{H'}(u_1, y) = \mathrm{dist}_{H'}(u_2, y)$. By the same arguments, if there are $u_1, u_2 \in X_j$ such that $\mathrm{dist}_{H'_j}(u_1, y) \neq \mathrm{dist}_{H'_i}(u_2, y)$, then u_1 or u_2 resolves x and y. Assume that all the vertices of W_i are at the same distance from x in H'_i and all the vertices of W_j are at the same distance from y in H'_j. Let $u_1 \in W_i$ and $u_2 \in W_j$. If $\mathrm{dist}_{H'_i}(u_1, x) \neq \mathrm{dist}_{H'_j}(u_2, y)$, then u_1 or u_2 resolves x and y, because $\mathrm{dist}_{H'}(u_1, y) = \mathrm{dist}_{H'}(u_2, x)$. Suppose that $\mathrm{dist}_{H'_i}(u_1, x) = \mathrm{dist}_{H'_j}(u_2, y) = 1$. Then $p_i = p_j = \mathit{true}$ and, by 4, $\mathrm{dist}_{F'}(v_i, v_j) = 2$ or there is $v_r \in Z$ such that $r \neq i, j$ and $\mathrm{dist}_{F'}(v_r, v_i) \neq \mathrm{dist}_{F'}(v_r, v_j)$. If $\mathrm{dist}_{F'}(v_i, v_j) = 2$, then u_1 resolves x and y. Otherwise, any vertex $u' \in W_r$ resolves x and y. If $\mathrm{dist}_{H'_i}(u_1, x) = \mathrm{dist}_{H'_j}(u_2, y) = 2$, then $q_i = q_j = \mathit{true}$ and, by 5, $\mathrm{dist}_{F'}(v_i, v_j) = 1$ or there is $v_r \in Z$ such that $r \neq i, j$ and $\mathrm{dist}_{F'}(v_r, v_i) \neq \mathrm{dist}_{F'}(v_r, v_j)$. If $\mathrm{dist}_{F'}(v_i, v_j) = 1$, then u_1 resolves x and y. Otherwise, any vertex $u' \in W_r$ resolves x and y. ◀

By 6, $p = \mathit{true}$ if and only if there is $i \in \{1, \ldots, h\} \setminus I$ such that $\mathrm{dist}_{F'}(v_i, v_j) = 1$ for every $v_j \in Z$ or there is $i \in \{h+1, \ldots, s\}$ such that $p_i = \mathit{true}$ and $\mathrm{dist}_{F'}(v_i, v_j) = 1$ for every $v_j \in Z \setminus \{v_i\}$. If there is $i \in \{1, \ldots, h\} \setminus I$ such that $\mathrm{dist}_{F'}(v_i, v_j) = 1$, then the unique vertex $x \in X_i$ is at distance 1 from any vertex of W. If there is $i \in \{h+1, \ldots, s\}$ such that $p_i = \mathit{true}$ and $\mathrm{dist}_{F'}(v_i, v_j) = 1$ for every $v_j \in Z \setminus \{v_i\}$, then there is a vertex $x \in X_i$ at distance 1 from each vertex of W_i, because $p_i = \mathit{true}$, and as $\mathrm{dist}_{F'}(v_i, v_j) = 1$ for $v_j \in Z \setminus \{v_i\}$, x is at distance 1 from any vertex of $W \setminus W_i$. Suppose that there is a vertex $x \in V(H)$ at distance 1 from each vertex of W. Let X_i be the module containing x. If $i \in \{1, \ldots, h\}$, then $i \in \{1, \ldots, h\} \setminus I$ and $\mathrm{dist}_{F'}(v_i, v_j) = 1$ for every $v_j \in Z$. Hence, $p = \mathit{true}$. If $i \in \{h+1, \ldots, s\}$, then $p_i = \mathit{true}$, because x is at distance 1 from all the vertices of W_i. Because x is at distance 1 from all the vertices of $W \setminus W_i$, $\mathrm{dist}_{F'}(v_i, v_j) = 1$ for every $v_j \in Z \setminus \{v_i\}$. Therefore, $p = \mathit{true}$.

Similarly, by 7, $q = \mathit{true}$ if and only if there is $i \in \{1, \ldots, h\} \setminus I$ such that $\mathrm{dist}_{F'}(v_i, v_j) = 2$ for $v_j \in Z$ or there is $i \in \{h+1, \ldots, s\}$ such that $q_i = \mathit{true}$ and $\mathrm{dist}_{F'}(v_i, v_j) = 2$ for every $v_j \in Z \setminus \{v_i\}$. If there is $i \in \{1, \ldots, h\} \setminus I$ such that $\mathrm{dist}_{F'}(v_i, v_j) = 2$ for every $v_j \in Z$, then the unique vertex $x \in X_i$ is at distance 2 from any vertex of W. If there is $i \in \{h+1, \ldots, s\}$ such that $q_i = \mathit{true}$ and $\mathrm{dist}_{F'}(v_i, v_j) = 2$ for $v_j \in Z \setminus \{v_i\}$, then there is a vertex $x \in X_i$ at distance 2 from each vertex of W_i, because $q_i = \mathit{true}$, and, as $\mathrm{dist}_{F'}(v_i, v_j) = 2$ for every $v_j \in Z \setminus \{v_i\}$, x is at distance 2 from any vertex of $W \setminus W_i$. Suppose that there is a vertex $x \in V(H)$ at distance 2 from each vertex of W. Let X_i be the module containing x. If $i \in \{1, \ldots, h\}$, then $i \in \{1, \ldots, h\} \setminus I$ and $\mathrm{dist}_{F'}(v_i, v_j) = 2$ for $v_j \in Z$. Hence, $q = \mathit{true}$. If $i \in \{h+1, \ldots, s\}$, then

$q_i = true$, because x is at distance 2 from the vertices of W_i. Because x is at distance 2 from the vertices of $W \setminus W_i$, $\text{dist}_{F'}(v_i, v_j) = 2$ fo everyr $v_j \in Z \setminus \{v_i\}$. Therefore, $q = true$.

We conclude that

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

Therefore, $w(H, p, q) \leq |W| = \omega(p, q, I, p_{h+1}, q_{h+2}, \ldots, p_s, q_s)$.

This finishes the proof of claim 6.4. ◀

Now we explain how to compute the metric dimension of G using the function $w(H, p, q)$. 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 $\text{mw}(G[X_i]) \leq t$ for $i \in \{1, ..., s\}$.

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

- **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.

  We again explain why the following equivalences hold. We do so only for the join of the graphs as the reasoning is analogous in other cases.

  First, if $|V(H_1)| = |V(H_2)| = 1$, then since the partitions are joined and there are only two vertices in the graph we only need any one of the vertices to resolve both of them, and, thus $\text{md}(G) = 1$.

  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 6.1

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

  Third, $|V(H_1)|, |V(H_2)| > 1$ and the values of $w(H_2, p, q)$ have already been computed for $p, q \in \{true, false\}$. By the definition of the join of graph, a vertex 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 6.1

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

  When computing $w(H, p, q)$ we construct a graph H' by adding a universal vertex to H. Doing so emulates the property that for each of the modules except for the root module in the modular decomposition of G, the distance between any two vertices inside a single module is at most 2. This is the case only for the proper modules of G, as for the set of vertices of G it can happen that $\text{diam}(G) > 2$. Therefore we have to modify some of the conditions in the third case.

- **Partitioning into modules.** Let $V(H)$ be partitioned into $s \leq t$ non-empty modules $X_1, ..., X_s$, $s \geq 2$. We again 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 again 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. Observe again that if $x \in X_i$ and $y \in X_j$ for distinct $i, j \in \{1, ..., s\}$, then $\text{dist}_G(x, y) = \text{dist}_F(v_i, v_j)$. Note that we do not consider F' and the distances within F can be more that 2.

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 $p_i = \mathit{true}$ 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 $q_i = \mathit{true}$ 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. $p_i = p_j = \mathit{true}$ 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 $\text{dist}_F(v_r, v_i) \neq \text{dist}_F(v_r, v_j)$,
5. $q_i = q_j = \mathit{true}$ 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 $\text{dist}_F(v_r, v_i) \neq \text{dist}_F(v_r, v_j)$.,

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

All of the conditions 1–5 are analogous to the conditions in computation of $w(H, p, q)$ with the exception that now, the distances can be larger than 2.

The claim about the metric dimension is expressed by the following theorem.

▶ **Theorem 6.6** ([2])**.** The function md, that expresses the metric dimension of a graph can be expressed as $\text{md}(G) = \min \omega(I, p_{h+1}, q_{h+1}, ..., p_s, q_s)$, where the minimum is taken over all possible values of $I \subseteq \{1, ..., h\}$ and $p_i, q_i \in \{\mathit{true}, \mathit{false}\}$ for $i \in \{h+1, ..., s\}$.

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

Let $W \subseteq V(G)$ be a resolving set of minimum size. By definition, $\text{md}(G) = |W|$. Let $W_i = W \cap X_i$. Let $I = \{i \mid i \in \{1, ..., h\}, W_i \neq \emptyset\}$. By Lemma 6.1, $W_i \neq \emptyset$ for $i \in \{h+1, ..., s\}$. For $i \in \{h+1, ..., s\}$, let $p_i = \mathit{true}$ if there is a vertex $x \in X_i$ such that $\text{dist}_G(x, u) = 1$ for all $u \in W_i$, and let $q_i = \mathit{true}$ if there is a vertex $y \in X_i$ such that $\text{dist}_G(y, u) = 2$ for all $u \in W_i$.

By Lemma 6.1, W_i resolves X_i in $G'[V(X_i)]$ for $i \in \{h+1, ..., s\}$, where G' is the graph obtained by the addition of a universal vertex to the graph G. This implies that $|W_i| \geq w(G[X_i], p_i, q_i)$ for $i \in \{h+1, ..., s\}$ and therefore $|W| \geq |i| + \sum_{i=h+1}^{s} w(G[X_i], p_i, q_i)$.

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 $\text{dist}_G(u, x) \neq \text{dist}_G(u, y)$. Consider the set W_r containing u. Vertices v_i, v_j are resolved by v_r, because $\text{dist}_F(v_r, v_i) = \text{dist}_G(u, x) \neq \text{dist}_G(u, y) = \text{dist}_F(v_r, v_j)$.

2. Assume that $p_i = \mathit{true}$ for some $i \in \{h+1, ..., s\}$ and consider some $j \in \{1, ..., h\} \setminus I$. Let us also assume that $\text{dist}_F(v_i, v_j) = 1$. 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 $\text{dist}_G(u, x) \neq \text{dist}_G(u, y)$. If $u \in X_i$ then we have that $\text{dist}_G(u, x) = 1 = \text{dist}_F(v_i, v_j) = \text{dist}_G(u, y)$. That is a contradiction. Thus u cannot belong to X_i. Let then X_r be the module containing u. Then we have that $\text{dist}_F(v_r, v_i) = \text{dist}_G(u, x) \neq \text{dist}_G(u, y) = \text{dist}_F(v_r, v_j)$.

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 $\mathrm{dist}_F(v_i, v_j) = 2$. 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 $\mathrm{dist}_G(u, x) = 2 = \mathrm{dist}_F(v_i, v_j) = \mathrm{dist}_G(u, y)$. 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 $\mathrm{dist}_F(v_r, v_i) = \mathrm{dist}_G(u, x) \neq \mathrm{dist}_G(u, y) = \mathrm{dist}_F(v_r, v_j)$.

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 $I \subseteq \{1, ..., h\}$ and 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

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

◀

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

▶ **Claim 6.7** ([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 $i, j \notin I$, 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. 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 $u_1, u_2 \in W_i$ such that $\text{dist}_{H_i'}(u_1, x) \neq \text{dist}_{H_i'}(u_2, x)$, then either u_1 or u_2 resolves x, y, because $\text{dist}_G(u_1, y) = \text{dist}_G(u_2, y)$. Assume that all the vertices of W_i are at the same distance from x in H_i'. Let $u \in W_i$. Let $u \in W_i$. If $\text{dist}_{H_i'}(u, x) = 1$, then $p_i = true$ and by the second 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 $\text{dist}_F(v_i, v_j) \geq 2$, x and y must be resolved by u, because $\text{dist}_G(u, y) \geq 2$. 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 $u_1, u_2 \in W_i$ 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 $\text{dist}_G(u_1, x) = \text{dist}_G(u_2, x)$. 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 $\text{dist}_F(v_i, v_j) \neq 1$, 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, $\text{md}(G) \leq |W|$ and $|W| = \omega(I, p_{h+1}, q_{h+1}, ..., p_s, q_s)$.

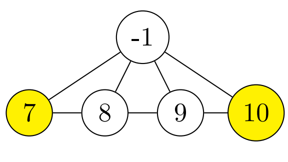
For evaluation of the running time of $w(H, p, q)$ we only need to consider the case when $V(H)$ can be partitioned into $s \leq t$ modules. We consider at most 4^t possibilities to choose I and p_i, q_i for $i \in \{h+1, ...s\}$. Then all the conditions can be verified in $O(t^3)$ time. Hence, the total 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. [8] is linear, we can solve the metric dimension problem in $O(4^t \cdot t^3 \cdot n + m)$ time.

6.2 An Example

We proceed to show the working of the algorithm on a simple example. The computation will be shown only for the branch that is going to give us the optimal result as going through the whole computation would be too lenghty. To do this we need to already know what nodes is the resolving set composed of and make some assumptions along the way. However the idea of the algorithm should still be very clear from the example.

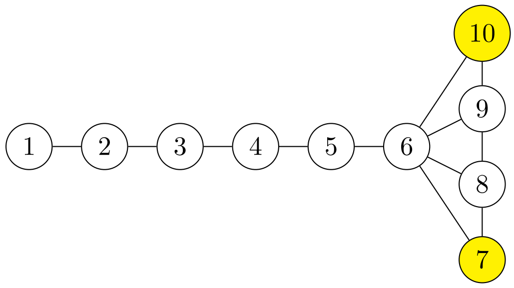
Let us have the graph G as shown in Figure 1. We state that for this graph the optimal resolving set $W = \{7, 10\}$. This graph has the modular decomposition depicted in Figure 2, where a PRIME module means the fourth operation of a modular decomposition as described in preliminaries.

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



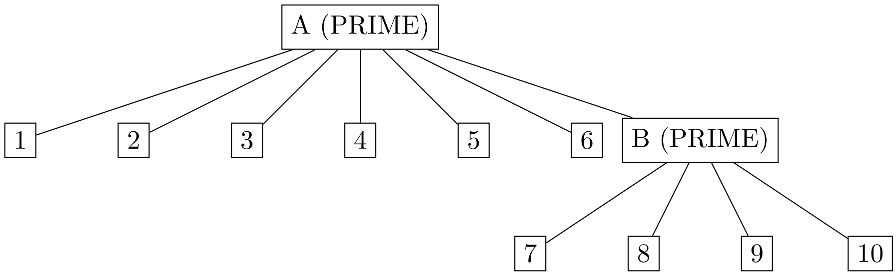
Computing from bottom-up, we first need to compute the value of $\omega(p, q, I, p_{h+1}, q_{h+1}, ..., p_s, q_s)$ for the prime node B. We can see that B has no non-trivial children nodes and four trivial nodes. Let us then choose $I = \{7, 10\}$, because we know that those are in the optimal resolving set.

Figure 1 Example graph G.



Note: Nodes in the resolving set for G are marked in yellow.

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



Note: "A (PRIME)" means the name of the node is A and it is a PRIME node.

Now it is also the time to construct the graphs F and F', shown in Figure 3 and Figure 4, as described in the algorithm. We proceed by verifying the conditions.

1. The set $Z = \{7, 10\}$ does resolve $V(F)$ in F', 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 $i \in \{8, 9\}$ it is not true that $\forall v_j \in Z : \text{dist}_{F'}(v_i, v_j) = 1$. Which means that $p = false$.
7. For $i \in \{8, 9\}$ it is not true that $\forall v_j \in Z : \text{dist}_{F'}(v_i, v_j) = 2$. Which means that $q = false$.

We get $\omega(false, false, \{7, 10\}) = |I| = 2$.

Finally, to calculate $md(G)$, we need to compute the value of $\omega(I, p_{h+1}, q_{h+1}, ..., p_s, q_s)$ for the prime node A. Since we know that no other vertices need to be added to the resolving set we set $I = \emptyset$. We also know the we have exactly one non-trivial node for which we set $p = false$ and $q = false$. This means we need to compute the function $\omega(I, p_B, q_B) = \omega(\emptyset, false, false)$. We again construct the graph F as described in the algorithm. Those can be seen in Figure 5. Next step is the verification of the five conditions.

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 $\{7, 8, 9, 10\}$.

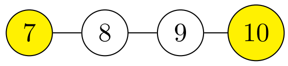
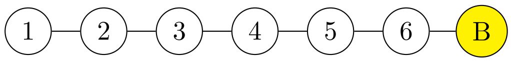


Figure 5 The prime graph F.



1. We have that $Z = \{B\}$, 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. Since $p_B = false$ the condition is satisfied.
3. Similar observation as in 2 can be made.
4. There is only one element to choose i, j from, so the condition is satisfied.
5. Same as above.

The conditions are satisfied and from the algorithm we get $\omega(\emptyset, false, false) = |I| + w(G[V(B)], false, false) = 0 + 2 = 2$. And since we stated that this would be the optimal computational branch we also get $md(G) = \omega(\emptyset, false, false) = 2$.

We will return to this example in the next section.

7 Truncated Metric Dimension

In the first part of this section, we present a simple modification to the algorithm bounded by modular-width to compute TRUNCATED METRIC DIMENSION*. In the latter part we show an example of computation using the algorithm and compare it to the example shown above.*

7.1 The Algorithm

Because, as stated before, the case where the graph is composed of a disjoint union of a pair of graphs or a join of a pair of graphs are subsumed by case where the graph is partitioned into at most t graphs, each of module-width at most t, we are only going to show how to alter the last case in this section. We start by explaining how to compute the $\omega(I, p_{h+1}, q_{h+1}, ..., p_s, q_s)$ function and later show an example of the computation. The $w(H, p, q)$ function is defined and computed in the same way as in the previous section, because of the fact that the distance between any two vertices of a single module (except for the root module) of a modular decomposition of a graph can be at most 2. This is also the reason why we only need to change the latter part of the computation. We feel the need to be point out that the only change in the computation is the usage of the truncated distance function instead of the non-truncated one. In practice, that means iterating through the elements of the distance matrix and clamping each of the values.

Partitioning into modules. Let $V(H)$ be partitioned into $s \leq t$ non-empty modules $X_1, ..., X_s$, $s \geq 2$. We again assume that $X_1, ..., X_h$ are *trivial*. And let again 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. Observe again that if $x \in X_i$ and $y \in X_j$ for distinct $i, j \in \{1, ..., s\}$, then $\mathrm{dist}_{G,k}(x, y) = \mathrm{dist}_{F,k}(v_i, v_j)$ for any k.

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

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

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

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

▶ **Theorem 7.1** ([2])**.** The function md_k, that expresses the truncated metric dimension of a graph can be computed as $\mathrm{md}_k(G) = \min \omega_k(I, p_{h+1}, q_{h+1}, ..., p_s, q_s)$, 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 $\mathrm{md}_k(G) \geq \min \omega_k(I, p_{h+1}, q_{h+1}, ..., p_s, q_s)$:

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 6.1, $W_i \neq \emptyset$ for $i \in \{h+1, ..., s\}$. For $i \in \{h+1, ..., s\}$, let $p_i = \mathit{true}$ if there is a vertex $x \in X_i$ such that $\mathrm{dist}_{G,k}(x, u) = 1$ for all $u \in W_i$, and let $q_i = \mathit{true}$ if there is a vertex $y \in X_i$ such that $\mathrm{dist}_{G,k}(y, u) = 2$ for all $u \in W_i$.

By Lemma 6.1, W_i resolves X_i in $G'[X_i]$ for $i \in \{h+1, ..., s\}$. This implies that $|W_i| \geq w(G[X_i], p_i, q_i)$ for $i \in \{h+1, ..., s\}$ and therefore $|W| \geq |i| + \sum_{i=h+1}^{s} w(G[X_i], p_i, q_i)$.

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 $\mathrm{dist}_{G,k}(u, x) \neq \mathrm{dist}_{G,k}(u, y)$. Consider the set W_r containing u. Vertices v_i, v_j are resolved by v_r, because $\mathrm{dist}_{F,k}(v_r, v_i) = \mathrm{dist}_{G,k}(u, x) \neq \mathrm{dist}_{G,k}(u, y) = \mathrm{dist}_{F,k}(v_r, v_j)$.

2. Assume that $p_i = \mathit{true}$ for some $i \in \{h+1, ..., s\}$ and consider some $j \in \{1, ..., h\} \setminus I$. Let us also assume that $\mathrm{dist}_{F,k}(v_i, v_j) = 1$. 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 $\mathrm{dist}_{G,k}(u, x) \neq \mathrm{dist}_{G,k}(u, y)$. If $u \in X_i$ then we have that $\mathrm{dist}_{G,k}(u, x) = 1 = \mathrm{dist}_{F,k}(v_i, v_j) = \mathrm{dist}_{G,k}(u, y)$. That is a contradiction. Thus u cannot belong to X_i. Let then X_r be the module containing u. Then we have that $\mathrm{dist}_{F,k}(v_r, v_i) = \mathrm{dist}_{G,k}(u, x) \neq \mathrm{dist}_{G,k}(u, y) = \mathrm{dist}_{F,k}(v_r, v_j)$.

3. Assume that $q_i = \mathit{true}$ for some $i \in \{h+1, ..., s\}$ and consider some $j \in \{1, ..., h\} \setminus I$. Let us also assume that $\mathrm{dist}_{F,k}(v_i, v_j) = 2$. 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 $\mathrm{dist}_{G,k}(u, x) \neq \mathrm{dist}_{G,k}(u, y)$. If $u \in X_i$ then we have that $\mathrm{dist}_{G,k}(u, x) = 2 = \mathrm{dist}_{F,k}(v_i, v_j) = \mathrm{dist}_{G,k}(u, y)$. 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 $\mathrm{dist}_{F,k}(v_r, v_i) = \mathrm{dist}_{G,k}(u, x) \neq \mathrm{dist}_{G,k}(u, y) = \mathrm{dist}_{F,k}(v_r, v_j)$.

4. Suppose that $p_i = p_j = \mathit{true}$ 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 $\mathrm{dist}_{G,k}(u, x) \neq \mathrm{dist}_{G,k}(u, y)$. If $u \in X_i$ then we get that $\mathrm{dist}_{G,k}(u, x) = \mathrm{dist}_{F,k}(v_i, v_j) = \mathrm{dist}_{G,k}(u, y)$. 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 $\mathrm{dist}_{F,k}(v_r, v_i) = \mathrm{dist}_{G,k}(u, x) \neq \mathrm{dist}_{G,k}(u, y) = \mathrm{dist}_{F,k}(v_r, v_j)$.

5. Suppose that $q_i = q_j = \mathit{true}$ for some $i \in \{h+1, ..., s\}$ and assume that $\mathrm{dist}_{F,k}(v_i, v_j) = 2$. 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 $\mathrm{dist}_{G,k}(u, x) \neq \mathrm{dist}_{G,k}(u, y)$. If $u \in X_i$ then we get that $\mathrm{dist}_{G,k}(u, x) = \mathrm{dist}_{F,k}(v_i, v_j) = \mathrm{dist}_{G,k}(u, y)$. 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 $\mathrm{dist}_{F,k}(v_r, v_i) = \mathrm{dist}_{G,k}(u, x) \neq \mathrm{dist}_{G,k}(u, y) = \mathrm{dist}_{F,k}(v_r, v_j)$.

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

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

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

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

◀

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

▶ **Claim 7.2** ([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 $i, j \notin I$, 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 $\mathrm{dist}_{F,k}(v_r, v_i) \neq \mathrm{dist}_{F,k}(v_r, v_j)$. Set W_r is not empty, by the definition of W_r and Z. Let $u \in W_r$. Then $\mathrm{dist}_{G,k}(u, x) = \mathrm{dist}_{F,k}(v_r, v_i) \neq \mathrm{dist}_{F,k}(v_r, v_i) = \mathrm{dist}_{G,k}(u, y)$.

Now assume that $i \in \{h + 1, ..., s\}$ and $j \in \{1, ..., h\}$. If there are $u_1, u_2 \in W_i$ 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 $\mathrm{dist}_{G,k}(u_1, y) = \mathrm{dist}_{G,k}(u_2, y)$. Assume that all the vertices of W_i are at the same distance from x in H'_i. Let $u \in W_i$. If $\mathrm{dist}_{H'_i}(u, x) = 1$, then $p_i = true$ and by the second condition, $\mathrm{dist}_{F,k}(v_i, v_j) \geq 2$ or there is $v_r \in Z$ such that $r \neq i, j$ and $\mathrm{dist}_{F,k}(v_r, v_i) \neq \mathrm{dist}_{F,k}(v_r, v_j)$. If $\mathrm{dist}_{F,k}(v_i, v_j) \geq 2$, x and y must be resolved by u, because $\mathrm{dist}_{G,k}(u, y) \geq 2$. Otherwise x and y are resolved by any vertex $u' \in W_r$.

In the same way if $\mathrm{dist}_G(u, x) = 2$, then $q_i = true$ and by the third condition $\mathrm{dist}_F(v_r, v_j) \neq \mathrm{dist}_F(v_r, v_j)$. If $\mathrm{dist}_G(u, x) \neq 2$, then u resolves x and y.

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

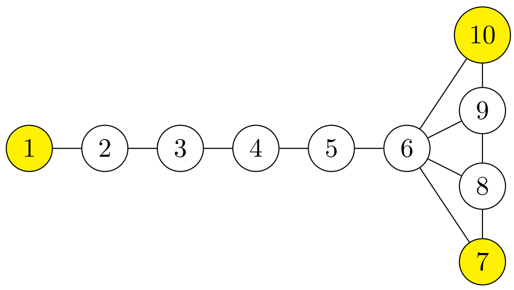
We have shown that W is a resolving set for G and, therefore, $\mathrm{md}(G) \leq |W|$ and $|W| = \omega(I, p_{h+1}, q_{h+1}, ..., p_s, q_s)$.

The evaluation of the running time of is very pretty much the same as for the previous algorithm. We already know that we can compute $w(H, p, q)$ in $O(4^t \cdot t^3)$ time. We only need to consider the case when G can be partitioned into $s \leq t$ modules. We consider at most 4^t possibilities 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 along with the clamping of the distance values which can be done in $O(t^2)$ time. Hence, the total time complexity is $O(4^t \cdot t^3)$. The conclusion is that, since the algorithm by Tedder et al. [8] is linear, we can solve the metric dimension problem in $O(4^t \cdot t^3 \cdot n + m)$ time.

7.2 An Example

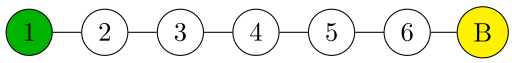
We are going to show an example computation on the graph we used in the section about METRIC DIMENSION to show the difference between the two algorithms. That is the graph depicted in Figure 6 with the modular decomposition shown in Figure 2. As mentioned above, there is no need to compute the function $w(\cdot)$ again, since we already did that in the previous section. Then the only part left to compute is $\omega(I, p_B, q_B)$. Just like in the previous section we are only showing the computation of the optimal computational branch. From the previous section we also know that for this case $p_B = false$ and $q_B = false$.

Figure 6 Example graph G.



Note: Nodes in the resolving set for G are marked in yellow.

Figure 7 The prime graph F.



Lets set the truncation parameter $k = 3$. We choose the paramter in this way so that the first condition of the computation is not met. In other words $Z = \{B\}$ is not resolving set for F (constructed as in the description of the algorithm and depicted in Figure 7), since for example $\text{dist}_{F,3}(B, 1) = \text{dist}_{F,3}(B, 2)$. To solve this we choose $I = \{1\}$, which means $Z = \{1, B\}$. Now can we verify the conditions again for the function $\omega(\{1\}, false, false)$.

1. We have that $Z = \{1, B\}$, and, thus the condition is satisfied.
2. Since $p_B = false$ the condition is 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.

All the conditions are satisfied and since we computed the function with the parameters that gives us the minimum number $\text{md}_k(G) = \omega(\{1\}, false, false) = 1 + 2 = 3$, for $k = 3$.

Algorithms Parameterized by Max Leaf Number

In this chapter we first describe the algorithm parameterized by Max Leaf Number and later describe the difficulties that prevented us from modifying the algorithm to solve TRUNCATED METRIC DIMENSION.

8 Metric Dimension

The algorithm is not parameterized directly by max leaf number, but rather by a functionally equivalent parameter, that is the number of *branches* in a graph. The fact that these parameters are equivalent was proven by the author [3] of the article and is captured by the following lemmas.

▶ **Definition 8.1** ([3])**.** A *branch* of a graph G is a maximal path or a cycle in which every internal vertex of the path has degree 2 in G. A vertex v belongs to a branch if v is incident to an edge of the branch and it is not incident to edges of any other branches.

▶ **Lemma 8.2** ([3])**.** In any connected graph with max leaf number l, there can be at most $O(l^2)$ branches.

▶ **Lemma 8.3** ([3])**.** Every connected graph $b > 0$ branches has max leaf number at most $2b$.

▶ **Lemma 8.4** ([3])**.** Let G be a graph, B be a branch of G, and s be any vertex of G. Then B may be partitioned into at most three contiguous paths within which the distance from s is monotonic.

▶ **Definition 8.5** ([3])**.** Let G be a graph, A, B two of its branches and let s be a vertex in a resolving set for G. Then the *indistinct* set for s, A, and B is defined to be the set of pairs of vertices (a, b) with $a \in A$ and $b \in B$ with $\text{dist}(s, a) = \text{dist}(s, b)$.

See Figure 8 for an example.

▶ **Lemma 8.6** ([3])**.** Let G be a graph, with A and B being two of its branches, and let s be a vertex in a resolving set for G. Then the indistinct set for s, A and B has the size $O(\min\{||A|, |B|\})$.

Proof. By Lemma 8.4 the vertices in A and B may be divided into at most three paths per branch, within which the distance from s is monotonic. Therefore, there are $O(1)$ points in both A and B that have a given distance d from s, and only $O(1)$ pairs of one point from A and one point from B that both have this distance. The total number of pairs that are not distinguished is the sum of this $O(1)$ bound over the at most $\min\{|A|, |B|\}$ different distances that need to be distinguished. ◀

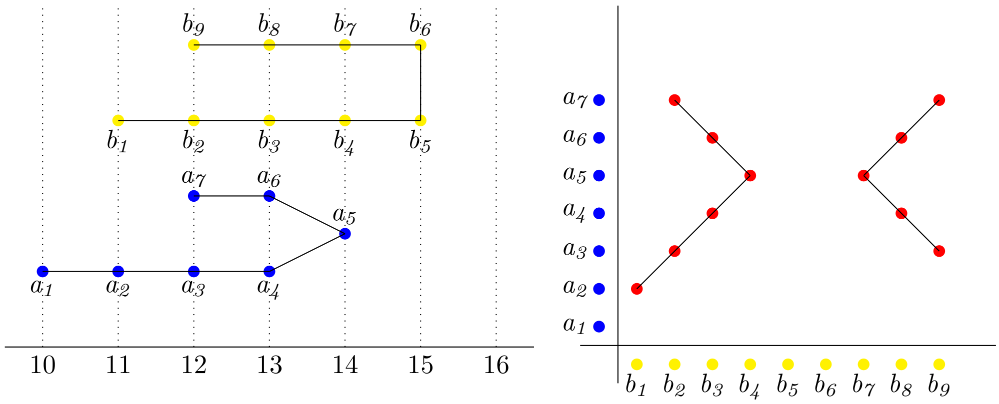


Figure 8 Two branches A and B arranged by their distances from a locating point s (left), and their indistinct set (of pairs not distinguished by s) plotted using the positions in the branches as Cartesian coordinates (right).

When plotted in two dimension, with the positions of a in A as one Cartesian coordinate and the position of b in B as the other, an indistinct set has the structure of $O(1)$ line segments whith slopes ± 1. By rotating this coordinate system by 45° we may use a more convenient coordinate system in which these segments are all horizontal or vertical. However, we must be careful when using this rotated system, as only the integer points with even sums of coordinates correspond to the integer points in the un-rotated system, which are the only points that can be members of an indistinct set.

Next, we consider how the indistinct set of s, A and B changes with the change of position of s along a third branch C.

▶ **Definition 8.7** ([3])**.** We say that two indistinct sets are combinatorially equivalent if there is a one-to-one correspondence between the diagonal segments of the two sets with the following properties:

- If s is a diagonal of one indistinct set, the corresponding diagonal in the other set has the same slope as s.
- If s and t are two diagonals of one indistinct set that intersect each other, then the corresponding diagonals in the other set also intersect each other.
- If s, t and u are three diagonals of one indistinct set, with t and u both intersecting s, then the corresponding two intersections of diagonals in the other intersecting set have the ordering.

Combinatorial equivalence is an equivalence relation and we define the combinatorial structure of an indistinct set to be its equivalence class in this equivalence relation.

▶ **Definition 8.8** ([3])**.** A *stem* is defined to be a maximal contiguous subset od a branch C of a given graph G within which the indistinct set of all points s in C and all pairs (A, B) of branches have the same combinatorial structure, as specified by Definition 8.7.

▶ **Lemma 8.9** ([3])**.** For a given pair of branches (A, B) and a third branch C, there are $O(1)$ positions along C such that the indistinct set of a vertex s of a branch C and the pair (A, B) changes structure at that position.

At all the points of C other than these, the indistinct set maintains the same combinatorial structure for s, A, and B. The position of its segments either remain fixed as s varies along the path, or they shift linearly with the position of s along C.

▶ **Lemma 8.10** ([3])**.** Every graph G with b branches has $O(b^3)$ stems.

▶ **Lemma 8.11** ([3])**.** The metric dimension of every graph with b branches is $O(b)$.

▶ **Theorem 8.12** ([3])**.** The metric dimension of any graph with n vertices and b branches may be determined in the time $O(n) + 2^{O(b^3 \log b)} \log n$.

Proof. We may assume without loss of generality that the graph is connected, for otherwise we could partition it into connected components and process each component separately. Partitioning the graph into branches may be performed in time $O(n)$. After this step all shortest path computations in the given graph can be performed by instead using a weighted graph with $O(b)$ vertices and edges, in which each edge represents a branch of the original graph and is weighted by that branch's length. In particular, after partitioning the graph into branches, we may partition the branches into stems in total time $b^{O(1)}$.

We search for locating sets of size $O(b)$, according to Theorem 8.11, by nondeterministically choosing the number of vertices in the locating set S, and the stem containing each vertex (but not the location of the vertex within the stem). There are $2^{O(b \log b)}$ possible choices of this type. This choice determines the combinatorial structure of each indistinct set.

Next, for each pair (A, B) of branches (allowing $A = B$) and each member s of the locating set (now associated with a specific stem but not placed at a particular vertex within that stem), we consider the line segments forming the indistinct sets for s, A and B, in the rotated coordinate system for which these line segments are horizontal and vertical. For a given pair (A, B) there are $O(b)$ line segments ($O(1)$ for each member of the locating set) and each line segment may be specified by the two Cartesian coordinate pairs for its endpoints. Rather than choosing these coordinate values numerically, we nondeterministically choose the sorted order of the x-coordinates and similarly the sorted order of the y-coordinates, allowing ties in our nondeterministic choices. In other words, separately for the x and y coordinates, we select a weak ordering of the segment endpoints, specifying for any two segment endpoints whether they have equal coordinate values or, if not, which one has a smaller coordinate value than the other. We also choose nondeterministically the parity of each Cartesian coordinate. Each of the $O(b^2)$ pairs of branches has $2^{O(b \log b)}$ choices for these orderings and parities, so there are $2^{O(b^3 \log b)}$ possible nondeterministic choices overall. For each such choice and each pair (A, B) we verify that, if we can find a placement of the vertices of the locating set that gives rise to the chosen sorted orderings, then the intersection of the indistinct sets for A and B will not contain any integer points (in the un-rotated coordinate system).

To test whether two indistinct sets have a non-empty intersection, we test each pair of a line segment from one set and a line segment from the other set for an intersection. Two horizontal line segments intersect each other if and only if they have the same y-coordinate and overlapping intervals of x-coordinates; a symmetric calculation is valid for two vertical line segments. A horizontal line segment intersects a vertical line segment if and only if the y-coordinate of the horizontal segment is within the range of y-coordinates of the vertical segment, the x-coordinate of the vertical segment is within the range of x-coordinates of the horizontal segment, and the parities of the coordinates of the two segments cause their crossing point to land on an integer point rather than on a half-integer point. In this way, the existence of an intersection point can be determined in time polynomial in b, using only the information about the sorted order and parities of coordinates that we have chosen nondeterministically.

When these nondeterministic choices find a collection of indistinct sets, and a sorted ordering of the features of those sets, for which every pair of branches has an empty intersection of indistinct sets, it remains to determine whether there exists a placement of each locating set vertex within its stem, in order to cause the indistinct set features to have the sorted orders that we have already chosen. Each ordering constraint between two features that are consecutive in one of the sorted orders translates directly to a linear constraint between the positions of two locating set vertices s and s' within their stems; therefore, the problem of finding positions that satisfy all of these constraints can be formulated and solved as an integer linear programming feasibility problem, with $O(b)$ variables (the positions of the locating vertices on their stems) and $O(b^3)$ constraints (sorted orderings of $O(b)$ items for each of $O(b^2)$ pairs of branches, specified with numbers of $O(\log n)$ bits (the lengths of the stems). By standard algorithms for low-dimensional integer linear programming problems, this problem can be solved in time $2^{O(b \log b)} \log n$. [29, 30, 31, 32].

The product of the numbers of nondeterministic choices made by the algorithm with the time for integer linear programming for each choice gives the stated time bound. ◀

This concludes the algorithm parameterized by max leaf number for METRIC DIMENSION.

9 Truncated Metric Dimension

There are several issues that do not allow us to modify the algorithm for the TRUNCATED METRIC DIMENSION problem in a reasonable way.

The lemmas 8.2, 8.3 and 8.4 hold for TRUNCATED METRIC DIMENSION using the same reasoning that was used in the original article, however in the Lemma 8.6 we can easily see, that the lemma does not hold for TRUNCATED METRIC DIMENSION. That is because, unlike for METRIC DIMENSION, for the paths A and B and a point s there are $O(|A| + |B|)$ points in both A and B that have given distance d from s and $O(|A| \cdot |B|)$ pairs of one point from A and one point from B that both have this distance.

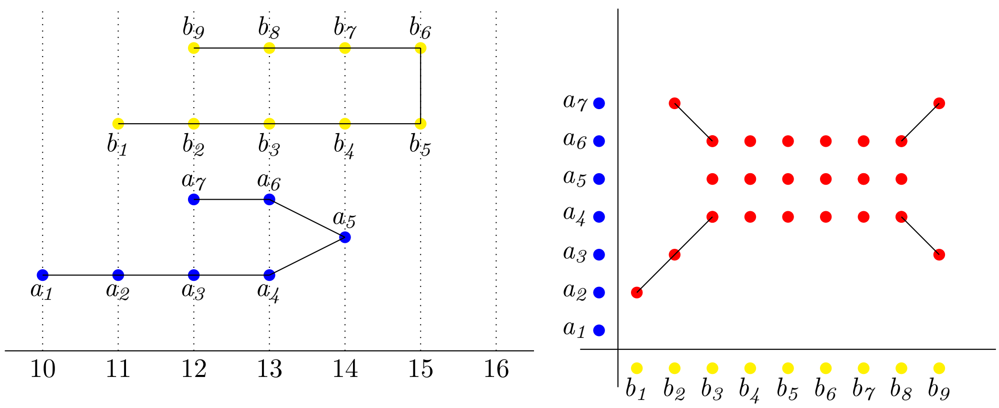


Figure 9 An example of Definition 8.5 when computing TRUNCATED METRIC DIMENSION

We also give a visual example how the truncated distance function changes the behavior of the Definition 8.5 in the Figure 9.

The main issue lies within the Lemma 8.11, where it is obvious to see, that the lemma does not hold for TRUNCATED METRIC DIMENSION. A simple counterexample to the lemma is a path of n vertices and the truncation parameter $k < n$. That is because the path is a single branch and we need $\Theta(\frac{n}{3k+2})$ [7] vertices in the resolving set for every vertex of the path to be resolved.

In the first step of the algorithm we partition the graph into branches and construct a weighted graph of $O(b)$ vertices and edges where each edge represents a branch of the original graph. Next we partition the branches into stems in $b^{O(1)}$ time and after that we would search for the resolving set of size $O(b)$ by non-deterministically choosing a number of vertices in the resolving set, and the stem containing each vertex. There would be $2^{O(b \log b)}$ such possible choices, but since, as stated above, the Lemma 8.11 does not hold, we would be searching for the resolving set of size $O(n)$ making it $2^{O(n \log n)}$ choices. This then means that the algorithm would not be parameterized by the max leaf number.

Implementation and Testing

10 Implementation

In this chapter we briefly describe the implementation of the algorithms for the generation of the data set and the metric dimension parameterized by modular-width. In the last part of this chapter we present results of the testing we did.

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 can be found in the `module_generator.py` file. There are two ways by which we can get a modular decomposition. One is by generating a random modular decomposition with the method `generate_md_tree` of the `modular_decomposition_generator` class. The function accepts two parameter. The modular-width t and the maximal depth of the modular decomposition tree d. The second way to get a modular decomposition is to read it from a file using the method `read_from_file`. This method only accepts one parameter and that is the path to the file from which the modular decomposition is to be read. This function was created mainly for repeated testing on the same data set. The data is expected to be formated from the second line (the first line is ignored) in the same way as printed out by `print_md_tree`.

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 to generating a graph and then calculating its modular decomposition is that 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 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 data 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 do not find that this change would yield any interesting results.

10.2 (Truncated) Metric Dimension Algorithms

There are two ways to compute the metric dimension. Either by using the function `md_naive`, which is a naive algorithm that was mainly implemented for testing purposes, or by the function `md`, which is an implementation of the algorithm by Belmonte et al. [2]. The function `md_naive` accepts graph as the only parameter, whereas the function `md` accepts the graph along with a modular decomposition of the graph.

Very similarly, there are two ways to compute the truncated metric dimension. Either by using the function `k_md_naive`, which is an implementation of a naive algorithm, or by the function `k_md`. The function `k_md_naive` accepts two parameters, a graph and an integer value to which the distances are truncated to. The function `k_md` accepts three parameters, a graph, a modular decomposition of the graph and an integer value to which the distances are truncated.

11 Measured Results

11.1 Metric Dimension

In this subsection we focus on performance of the `md` 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 data are sorted into folders. The first level of folders divides the data by modular-width parameter and the second level divides the data by the depth of the modular decomposition 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 that, for this test, reasonable values for the modular-width are 4, 6 and 8 and modular decomposition tree depth 2 or 3.

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. W 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. all 4^k possible permutations of values are generated and tested, where k is the number of non-trivial modules (2 boolean values for each non-trivial module),
2. all the combinations of the trivial modules are tested,
3. a table of distances in the sub-graph for the module is computed.

Table 1 Performance of the algorithm

t modular-width

z average size of a non-trivial module

Note: Each graph was tested 3 times and the running times were averaged.

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	0.2
4	4	3.7	11	5	0.2
4	4	4.2	13	8	0.3
4	5	4.0	15	10	0.8
4	8	1.3	17	7	8.9
4	11	3.6	29	19	59.3
4	13	3.0	27	16	113.2
4	13	3.3	30	18	128.6
4	14	3.7	39	22	116.5
6	5	5.4	22	13	15.9
6	6	3.7	21	11	14.0
6	6	4.1	20	10	27.3
6	7	5.0	31	20	65.5
6	7	5.1	30	15	80.9
6	10	4.3	33	18	2499.0
6	10	5.3	43	21	3550.6
6	15	4.5	53	31	17030.4
6	17	4.4	58	37	18019.1
6	24	4.7	89	61	13147.6
8	5	5.4	22	10	90.2
8	6	6.1	31	16	690.5
8	6	6.6	34	26	1293.5
8	7	5.2	30	14	1773.0
8	7	5.8	34	26	1292.2

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 of course happen that for some graphs more of the conditions can be satisfied, and, thus not yielding an early return and resulting in more demanding calculation and vice versa. We can observe this for example for the very bottom of the computations of modular-width 4 and 6.

We conclude that the algorithm performs within our expectations.

11.2 Truncated Metric Dimension

The performance of computing `k_md` function depends on the very same factors as the computation of the `md` function, that were described in the previous subsection. From the Table 2 we can see that the time does not depend on the truncation parameter k in any significant way. Notice that for the most part the times are very similar to the times of the `md` function. This should be obvious from the description of the algorithm.

Note that in the Figure 2 the truncated metric dimension does not change for different k for this particular data set as these graphs are highly connected.

We again conclude that the performance of the algorithm is within our expectation.

Table 2 Performance of the algorithm

t modular-width

z average size of a non-trivial module

Note: Each graph was tested 3 times and the running times were averaged.

t	# non-trivial modules	z	$\mid V(G) \mid$	$\mathrm{md}_k(G)$	time [s] ($k = 3$)	time [s] ($k = 6$)
4	2	4	7	2	0.0	0.0
4	4	3.7	11	7	0.2	0.2
4	4	3.7	11	5	0.2	0.2
4	4	4.2	13	8	0.8	0.4
4	5	4.0	15	10	0.9	0.8
4	8	1.3	17	7	9.0	9.1
4	11	3.6	29	19	56.7	55.9
4	13	3.0	27	16	108.3	105.5
4	13	3.3	30	18	121.0	117.6
4	14	3.7	39	22	116.6	110.8
6	5	5.4	22	13	14.6	14.0
6	6	3.7	21	11	10.7	10.4
6	6	4.1	20	10	23.2	22.7
6	7	5.0	31	20	57.9	55.7
6	7	5.1	30	15	70.0	67.7
6	10	4.3	33	18	1877.2	2020.7
6	10	5.3	43	21	2526.5	2678.6
6	15	4.5	53	31	17023.0	16592.6
6	17	4.4	58	37	18069.7	17415.2
6	24	4.7	89	61	13258.7	11178.1
8	5	5.4	22	10	167.2	136.5
8	6	6.1	31	16	479.6	452.4
8	6	6.6	34	26	866.9	875.5
8	7	5.2	30	14	1998.1	1515.3
8	7	5.8	34	26	1646.5	1437.8

Conclusion

The goals of this thesis were to research Metric Dimension and already known FPT algorithms for the problem. Then find out whether it is possible for some of these algorithms to also solve the truncated version of Metric Dimension with minimal modifications to the algorithm itself and lastly to implement one of the chosen algorithms.

We got familiar with the concept of Metric Dimension and its truncated variant. We also got familiar with concepts of parameterized complexity and various structural parameters. We have shown that for the algorithm parameterized by modular-width, published by Belmonte et al. [2], it is easy to convert it in a suitable way to solve Truncated Metric Dimension. We have implemented the algorithm for both the truncated and non-truncated versions of the problem and tested it on suitable dataset.

We have also presented the algorithm parameterized by max leaf number for Metric Dimension and shown that it is not suitable to be converted for Truncated Metric Dimension.

12 Possible Improvements

There are two ways that immediately come to mind when thinking about how to iterate on this thesis. First way would be to consider other structural parameters, mainly the ones mentioned in the beginning of this thesis. For example, an algorithm solving the Metric Dimension problem with linear running time with respect to tree-width is known. Second, both of the algorithms could then be re-implemented in a more performant language, for example C++, 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. EPPSTEIN, David. Metric Dimension Parameterized by Max Leaf Number. *Journal of Graph Algorithms and Applications*. 2015, vol. 19, no. 1, pp. 313–323. ISSN 1526-1719. Available from DOI: `10.7155/jgaa.00360`.
4. 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`.
5. 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`.
6. 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`.
7. 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`.
8. 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`.
9. 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`.

10. SLATER, Peter. *Leaves of Trees*. Utilitas Mathematica Pub, 1975. Proceedings of the South-eastern Conference on Combinatorics, Graph Theory, and Computing, Congressus numerantium, no. 14.
11. HARARY, Frank; MELTER, Robert. *On the Metric Dimension of a Graph*. The Charles Babbage Research Centre, 1976. Ars Combinatoria, no. 2.
12. 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.
13. 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.
14. 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.
15. HARTUNG, Sepp; NICHTERLEIN, André. On the Parameterized and Approximation Hardness of Metric Dimension. In: *Proceedings of the 28th Conference on Computational Complexity, CCC 2013, K.lo Alto, California, USA, 5-7 June, 2013.* IEEE Computer Society, 2013, pp. 266–276. Available from DOI: 10.1109/CCC.2013.36.
16. DIAZ, Josep; POTTONEN, Olli; SERNA, Maria; VAN LEEUWEN, Erik Jan. Complexity of metric dimension on planar graphs. *Journal of Computer and System Sciences.* 2017, vol. 83, no. 1, pp. 132–158. ISSN 0022-0000. Available from DOI: https://doi.org/10.1016/j.jcss.2016.06.006.
17. BONNET, Édouard; PUROHIT, Nidhi. Metric Dimension Parameterized By Treewidth. *Algorithmica.* 2021, vol. 83, no. 8, pp. 2606–2633. Available from DOI: 10.1007/S00453-021-00808-9.
18. LI, Shaohua; PILIPCZUK, Marcin. Hardness of Metric Dimension in Graphs of Constant Treewidth. *Algorithmica.* 2022, vol. 84, no. 11, pp. 3110–3155. Available from DOI: 10.1007/S00453-022-01005-Y.
19. BOUSQUET, Nicolas; DESCHAMPS, Quentin; PARREAU, Aline. Metric Dimension Parameterized by Treewidth in Chordal Graphs. In: PAULUSMA, Daniël; RIES, Bernard (eds.). *Graph-Theoretic Concepts in Computer Science - 49th International Workshop, WG 2023, Fribourg, Switzerland, June 28-30, 2023, Revised Selected Papers.* Springer, 2023, vol. 14093, pp. 130–142. Lecture Notes in Computer Science. Available from DOI: 10.1007/978-3-031-43380-1_10.
20. FOUCAUD, Florent; GALBY, Esther; KHAZALIYA, Liana; LI, Shaohua; INERNEY, Fionn Mc; SHARMA, Roohani; TALE, Prafullkumar. *Problems in NP can Admit Double-Exponential Lower Bounds when Parameterized by Treewidth or Vertex Cover.* 2024. Available from arXiv: 2307.08149 [cs.CC].
21. HERNANDO, Carmen; MORA, Mercè; PELAYO, Ignacio M.; SEARA, Carlos; WOOD, David R. Extremal Graph Theory for Metric Dimension and Diameter. *Electron. J. Comb.* 2007, vol. 17. Available also from: https://doi.org/10.37236/302.
22. CHARTRAND, Gary; EROH, Linda; JOHNSON, Mark A.; OELLERMANN, Ortrud R. Resolvability in graphs and the metric dimension of a graph. *Discret. Appl. Math.* 2000, vol. 105, pp. 99–113. Available also from: https://doi.org/10.1016/S0166-218X(00)00198-0.
23. MELTER, Robert A.; TOMESCU, Ioan. Metric bases in digital geometry. *Comput. Vis. Graph. Image Process.* 1984, vol. 25, pp. 113–121. Available also from: https://doi.org/10.1016/0734-189X(84)90051-3.

24. RAJAN, Bharati; RAJASINGH, Indra; CYNTHIA, Jude Annie; MANUEL, Paul D. Metric dimension of directed graphs. *International Journal of Computer Mathematics*. 2014, vol. 91, pp. 1397–1406. Available also from: `https://doi.org/10.1080/00207160.2013.844335`.

25. GENESON, Jesse; YI, Eunjeong. Broadcast Dimension of Graphs. *AUSTRALASIAN JOURNAL OF COMBINATORICS*. 2022, vol. 83(2). Available also from: `https://ajc.maths.uq.edu.au/pdf/83/ajc_v83_p243.pdf`.

26. 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`.

27. 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`.

28. GUTKOVICH, Paul; YEOH, Zi Song. Computing Truncated Metric Dimension of Trees. *ArXiv*. 2023, vol. abs/2302.05960. Available also from: `https://api.semanticscholar.org/CorpusID:254594288`.

29. LENSTRA, H. W. Integer Programming with a Fixed Number of Variables. *Mathematics of Operations Research*. 1983, vol. 8, no. 4, pp. 538–548. Available from DOI: `10.1287/moor.8.4.538`.

30. KANNAN, Ravi. Minkowski's Convex Body Theorem and Integer Programming. *Mathematics of Operations Research*. 1987, vol. 12, no. 3, pp. 415–440. Available from DOI: `10.1287/moor.12.3.415`.

31. FRANK, András; TARDOS, Éva. An application of simultaneous diophantine approximation in combinatorial optimization. *Combinatorica*. 1987, vol. 7, pp. 49–65. Available also from: `https://doi.org/10.1007/BF02579200`.

32. CLARKSON, Kenneth L. Las Vegas algorithms for linear and integer programming when the dimension is small. *J. ACM*. 1995, vol. 42, no. 2, pp. 488–499. ISSN 0004-5411. Available from DOI: `10.1145/201019.201036`.

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
```