Hardness of Metric Dimension in Graphs of Constant Treewidth

Shaohua Li
Faculty of Mathematics, Informatics and Mechanics, University of Warsaw, Poland

Marcin Pilipczuk
Faculty of Mathematics, Informatics and Mechanics, University of Warsaw, Poland

Abstract

The Metric Dimension problem asks for a minimum-sized *resolving set* in a given (unweighted, undirected) graph G. Here, a set $S \subseteq V(G)$ is *resolving* if no two distinct vertices of G have the same distance vector to S. The complexity of Metric Dimension in graphs of bounded treewidth remained elusive in the past years. Recently, Bonnet and Purohit [IPEC 2019] showed that the problem is W[1]-hard under treewidth parameterization. In this work, we strengthen their lower bound to show that Metric Dimension is NP-hard in graphs of treewidth 24.

2012 ACM Subject Classification Mathematics of computing → Graph algorithms

Keywords and phrases Graph algorithms, parameterized complexity, width parameters, NP-hard

Digital Object Identifier 10.4230/LIPIcs.IPEC.2021.24

Related Version *Full Version*: `https://arxiv.org/abs/2102.09791`

Funding This research is part of a project that has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme, grant agreement 714704.

1 Introduction

Let G be an unweighted and undirected graph and let $S \subseteq V(G)$. For a vertex $v \in V(G)$, the *distance vector* from v to S is the assignment $S \ni w \mapsto \mathrm{dist}_G(v, w)$, where dist_G denotes the distance in the graph G. The set S is *resolving* if a distance vector to S uniquely determines the source vertex; that is, no two vertices of G have the same distance vector to S. The Metric Dimension problem asks for a resolving set of minimum possible size; such a set is sometimes called the *metric basis* of G. The decision version of Metric Dimension asks for a resolving set of size not exceeding a given threshold k.

Metric Dimension has already been introduced in 1970s [7, 12]. Determining its computational complexity turned out to be quite challenging. It is polynomial-time solvable on trees [7, 12, 10], outerplanar graphs [3], and chain graphs [6], but NP-hard for example on planar graphs [3] or split graphs [5]. From the parameterized complexity point of view, the FPT status of the Metric Dimension parameterized by the solution size has been open for a while and finally resolved in negative by Hartung and Nichterlein [8]. In the search of a tractable structural parameterization, FPT algorithms has been shown for parameters: treelength plus maximum degree [1], vertex cover number [8], max leaf number [4], and modular-width [1].

The above list misses probably the most important graph width measure, namely treewidth. Determining the complexity of Metric Dimension, parameterized by treewidth, remained elusive in the last years, and has been asked a few times [1, 3, 4]. Bonnet and Purohit in a paper presented at IPEC 2019 [2] showed that the problem is W[1]-hard, even with pathwidth parameterization. In this work we strengthened their result by proving para-NP-hardness of this parameterization.

▶ **Theorem 1.** *Metric Dimension, restricted to graphs of treewidth at most* 24*, is NP-hard.*

Theorem 1 brings us much closer to closing (unfortunately mostly in negative) the question of the complexity of Metric Dimension in graphs of bounded treewidth. The remaining gap is to determine the exact treewidth value where the problem becomes hard: note that it is open if Metric Dimension is polynomial-time solvable on graphs of treewidth 2.

The proof of Theorem 1 starts with a construction of a graph with a separation of order 9 over which *a lot* of information on a partial solution to Metric Dimension is transfered. More formally, similarly as Bonnet and Purohit [2], we use the Multicolored Resolving Set problem as an auxiliary intermediate problem. In this problem, the input graph is additionally equipped with an integer k, a tuple of k disjoint vertex sets $X_1, X_2, \ldots, X_k$, and a set $\mathcal{P}$ of vertex pairs. The goal is to choose a set S consisting of exactly one vertex from each set X_i so that for every $\{u, v\} \in \mathcal{P}$, the pair $\{u, v\}$ is resolved by S, that is, u and v have different distance vectors to S. In our construction, the sets X_i are on one side of the said separation of order 9, while the pairs $\mathcal{P}$ are on the second side. The crux of the construction is to make every distance from a vertex of the separator to a chosen vertex of S count: despite the fact that the separation has constant size, S is of unbounded size, giving $\Omega(|S|)$ distances to work with. Overall, the above gives a relatively clean reduction giving NP-hardness of Multicolored Resolving Set in graphs of constant treewidth, presented in Section 3. This reduction is the main new insight and technical contribution of this paper.

Then, again similarly as in the work of Bonnet and Purohit [2], it takes a lot of effort (presented in Section 4) to turn the above reduction to Multicolored Resolving Set into a reduction to Metric Dimension. Here, there are two problems. First, one needs to introduce some gadgets to force the solution to take exactly one vertex from each set X_i. Second, one needs to ensure that the intended solution resolves *all* vertex pairs, not only the ones from $\mathcal{P}$. For both problems, we borrow the tools from Bonnet and Purohit [2]. In particular, the first problem is resolved by *forced set gadgets* in a way very similar to [2]. The second problem is resolved by adding a number of new connections to the graph and *forced vertex gadgets* of [2]. Thus, while the toolbox remains the same as in [2], the application is different as the graph we are working with is significantly different. The construction is presented in Sections 4.1-4.2.

After applying all the modifications to obtain a Metric Dimension instance, one needs to check three aspects. First, one needs to ensure that the forced set gadgets work as intended, forcing the solution to take one vertex from each X_i; this check is rather simple and very similar to the analogous check of [2]. Second, one needs to check that the introduced forced vertex gadgets, that contain extra vertices from the intended resolving set (apart from the ones in X_is), do not accidentally resolve any pair from $\mathcal{P}$. This check is not trivial, but still relatively simple. Note that the mentioned two properties already ensure one of the implications in the proof of the correctness of the reduction: if the final Metric Dimension instance is a yes-instance, then the input instance of the source problem is a yes-instance. These two checks are presented in Section 4.3.

Then one needs to check that every pair of vertices is resolved by an intended solution. Due to the complexity of the construction and the properties of this problem, this turned out to be long and arduous (the full proof is deferred to the full version of the paper).

Besides, we show that the treewidth of the constructed graph is bounded by a constant in Section 4.4.

2 Preliminaries

In this paper, all graphs are undirected. In a graph G, let $V(G)$ be the set of vertices of G. For a vertex $v \in V(G)$, we denote the open neighborhood and closed neighborhood of v by $N_G(v)$ and $N_G[v]$ respectively (or just $N(v)$ and $N[v]$ if the graph is clear in the context). For two vertices $u, v \in V(G)$, let $P(u, v)$ be a path from u to v. Since the graph is undirected, $P(u, v)$ and $P(v, u)$ denote the same path. We denote the neighbor of u on $P(u, v)$ by $N_u(u, v)$ (also the neighbor of v on $P(u, v)$ by $N_v(u, v)$). Similarly, if there is a path which is named as, for example, $P^h(i, j, x)$ such that u is one endpoint of $P^h(i, j, x)$, we denote the neighbor of u on $P^h(i, j, x)$ by $N_u^h(i, j, x)$. For simplicity, we abuse the notation in the sense that for a path P, we use P to denote the path or the vertex set of the path. The meaning should be clear in the context. We define the length of a path P to be the number of edges on the path and denote it by $|P|$. For two vertices $u, v \in V(G)$, we define the distance between u and v to be the length of any shortest path from u to v, denoted by $\text{dist}_G(u, v)$. Note that in this paper we use $\text{dist}(u, v)$ to denote the distance between u and v mostly if the graph is clear in the context. For a path P of even length with two endpoints u and v, let w be the vertex on P such that the length of the subpath of P from u to w equals the length of the subpath of P from w to v. Then we call w the *middle vertex* of P and denote it by $\text{mid}(P)$. We say that two distinct vertices u, u' are *false twins* if $N[u] = N[u']$. Since a path decomposition is also a tree decomposition, the treewidth of a graph G is at most the pathwidth of G. In this paper, for convenience of the proof, we use the alternative characterization of pathwidth, i.e. the pathwidth of a graph G equals the *node search number* of G minus one [11]. The definition of the node search number comes from the node search game. We give an informal definition of the node search game as follows. Imagine that the edges of an undirected graph G are tunnels and they are contaminated by some gas. We need to put searchers on vertices of G to clean the gas. The rule is that when the two endpoints of an edge are occupied by two searchers, this edge becomes clean. However, if we remove some searchers from the graph, a cleaned edge can be recontaminated immediately through an unoccupied endpoint to which a contaminated edge is incident. The node search number of G is the minimum number of searchers required to clean all edges of G.

3 Reduction from 3-Dimensional Matching to Multicolored Resolving Set

Bonnet and Purohit introduced k-Multicolored Resolving Set as an intermediate problem in order to show the W[1]-hardness of Metric Dimension parameterized by treewidth [2].

k-Multicolored Resolving Set

Input: An undirected graph $G = (V, E)$, an integer k, a set $\chi = \{X_1, ..., X_k\}$ where $X_1, ..., X_k$ are disjoint subsets of $V(G)$ and a set $\mathcal{P} = \{\{x_1, y_1\}, ..., \{x_h, y_h\}\}$ where $\{x_1, y_1\}, ..., \{x_h, y_h\}$ are vertex pairs of G.

Question: Is there a set of k vertices S such that
(i) $|S \cap X_i| = 1$ for every $i = 1, ..., k$, and
(ii) for every $\ell \in \{1, ..., h\}$, there exists a vertex $v \in S$ such that $\text{dist}(v,x_\ell) \neq \text{dist}(v,y_\ell)$.

We show that this problem is NP-hard on graphs of constant treewidth. We make a reduction from 3-dimensional matching, which is well-known to be NP-hard [9].

3-DIMENSIONAL MATCHING

Input: the universe $U = \{1,2,3\} \times [n]$ and a set $\mathcal{F} = \{A_1, ..., A_m\}$ of tuples such that for every $j \in [m]$, the tuple $A_j = \{(1,x),(2,y),(3,z)\}$ where $(1,x),(2,y),(3,z) \in U$.

Question: are there n tuples $A_{j_1}, ..., A_{j_n}$ such that $\bigcup\limits_{h=1}^{n} A_{j_h} = U$.

Given an instance $(U, \mathcal{F})$ of 3-DIMENSIONAL MATCHING with the universe $U = \{1,2,3\} \times [n]$ and a set $\mathcal{F}$ of m tuples $A_1, ..., A_m \subseteq U$, we construct an instance $(G, n, \chi, \mathcal{P})$ of n-MULTICOLORED RESOLVING SET as follows. First, we create m vertices $s_i^1, ..., s_i^m$ as X_i for each $i \in [n]$. Let $\chi = \{X_1, ..., X_n\}$ and $X = \bigcup\limits_{i=1}^{n} X_i$. Then we create n vertex pairs $\{u_r^1, v_r^1\}, ..., \{u_r^n, v_r^n\}$ for each $r \in \{1,2,3\}$ and let $\mathcal{P}_r = \{\{u_r^i, v_r^i\} | i = 1, ..., n\}$. We create 3 vertices a_r, b_r, c_r and let $W_r = \{a_r, b_r, c_r\}$ for each $r \in \{1,2,3\}$. Let $\mathcal{P} = \mathcal{P}_1 \cup \mathcal{P}_2 \cup \mathcal{P}_3$ and $W = W_1 \cup W_2 \cup W_3$. Finally, let $M = 40(n+1)$. For each tuple $A_j = \{(1,x),(2,y),(3,z)\}$ ($j \in [m], x, y, z \in [n]$) of the given instance and each integer $i \in [n]$, we link s_i^j to a_1, b_1, c_1 with three paths $P(s_i^j, a_1), P(s_i^j, b_1), P(s_i^j, c_1)$ of lengths $\frac{M}{2} + 10x$, $\frac{M}{2} + 5x + 1$ and $\frac{M}{2} - 10x$ respectively, link s_i^j to a_2, b_2, c_2 with three paths $P(s_i^j, a_2), P(s_i^j, b_2), P(s_i^j, c_2)$ of lengths $\frac{M}{2} + 10y$, $\frac{M}{2} + 5y + 1$ and $\frac{M}{2} - 10y$ respectively, and link s_i^j to a_3, b_3, c_3 with three paths $P(s_i^j, a_3), P(s_i^j, b_3), P(s_i^j, c_3)$ of lengths $\frac{M}{2} + 10z$, $\frac{M}{2} + 5z + 1$ and $\frac{M}{2} - 10z$ respectively. For every vertex pair $\{u_r^p, v_r^p\}$ ($p \in [n], r \in \{1,2,3\}$), we link u_r^p to a_r, b_r, c_r with three paths $P(u_r^p, a_r), P(u_r^p, b_r), P(u_r^p, c_r)$ of lengths $\frac{M}{2} - 10p$, $\frac{M}{2} - 5p - 1$ and $\frac{M}{2} + 10p$ respectively, and link v_r^p to a_r, b_r, c_r with three paths $P(v_r^p, a_r), P(v_r^p, b_r), P(v_r^p, c_r)$ of lengths $\frac{M}{2} - 10p$, $\frac{M}{2} - 5p - 2$ and $\frac{M}{2} + 10p$ respectively. This finishes the construction. See Figure 1 for an example.

▶ **Lemma 2.** *For an arbitrary vertex pair $\{u_r^x, v_r^x\} \in \mathcal{P}$ ($r \in \{1,2,3\}, x \in [n]$),$\{u_r^x, v_r^x\}$ is resolved by s_i^j ($i \in [n], j \in [m]$) if and only if $(r,x) \in A_j$.*

Proof. On one hand, suppose that $(r,x) \in A_j$. For an arbitrary $i \in [n]$, the three paths from s_i^j to u_r^x via a_r, b_r and c_r have lengths M, M and M respectively. The three paths from s_i^j to v_r^x via a_r, b_r and c_r have lengths $M, M-1$ and M respectively. Note that there could be other paths from s_i^j to v_r^x or u_r^x that go repeatedly between vertices in X and vertices in W. However, the lengths of such paths are at least $M - 20n + M - 10n > M$. As a result, the shortest paths from s_i^j to u_r^x and v_r^x are of lengths M and $M-1$ respectively. Thus $\{u_r^x, v_r^x\}$ is resolved by s_i^j.

On the other hand, for an arbitrary tuple $A_i = \{(1,p_1),(2,p_2),(3,p_3)\}$, the paths from the vertex s_i^j ($i \in [n]$) to u_r^x ($r \in \{1,2,3\}$) via a_r, b_r and c_r have lengths $M + 10(p_r - x), M + 5(p_r - x)$ and $M - 10(p_r - x)$ respectively. The paths from the vertex s_i^j ($i \in [n]$) to v_r^x ($r \in \{1,2,3\}$) via a_r, b_r and c_r have lengths $M + 10(p_r - x), M + 5(p_r - x) - 1$ and $M - 10(p_r - x)$ respectively. Note that the paths from s_i^j to u_r^x (or v_r^x) that go repeatedly between the vertices in X and the vertices in W have lengths at least $M - 20n + M - 10n > M + 10n$. They are not the shortest paths from s_i^j to u_r^x (or v_r^x). If $p_r < x$, the shortest paths from s_i^j to u_r^x and v_r^x both have lengths $M + 10(p_r - x)$. If $p_r > x$, the shortest paths from s_i^j to u_r^x and v_r^x both have lengths $M - 10(p_r - x)$. If $p_r = x$, the shortest paths from s_i^j to u_r^x and v_r^x have lengths M and $M-1$ respectively. As a result, if $\{u_r^x, v_r^x\}$ is resolved by s_i^j, then $p_r = x$. According to the construction, $(r,x) \in A_j$. ◀

▶ **Lemma 3.** *The constructed instance $(G, n, \chi, \mathcal{P})$ of n-MULTICOLORED RESOLVING SET is a yes-instance if and only if the given instance $(U, \mathcal{F})$ of 3-DIMENSIONAL MATCHING is a yes-instance.*

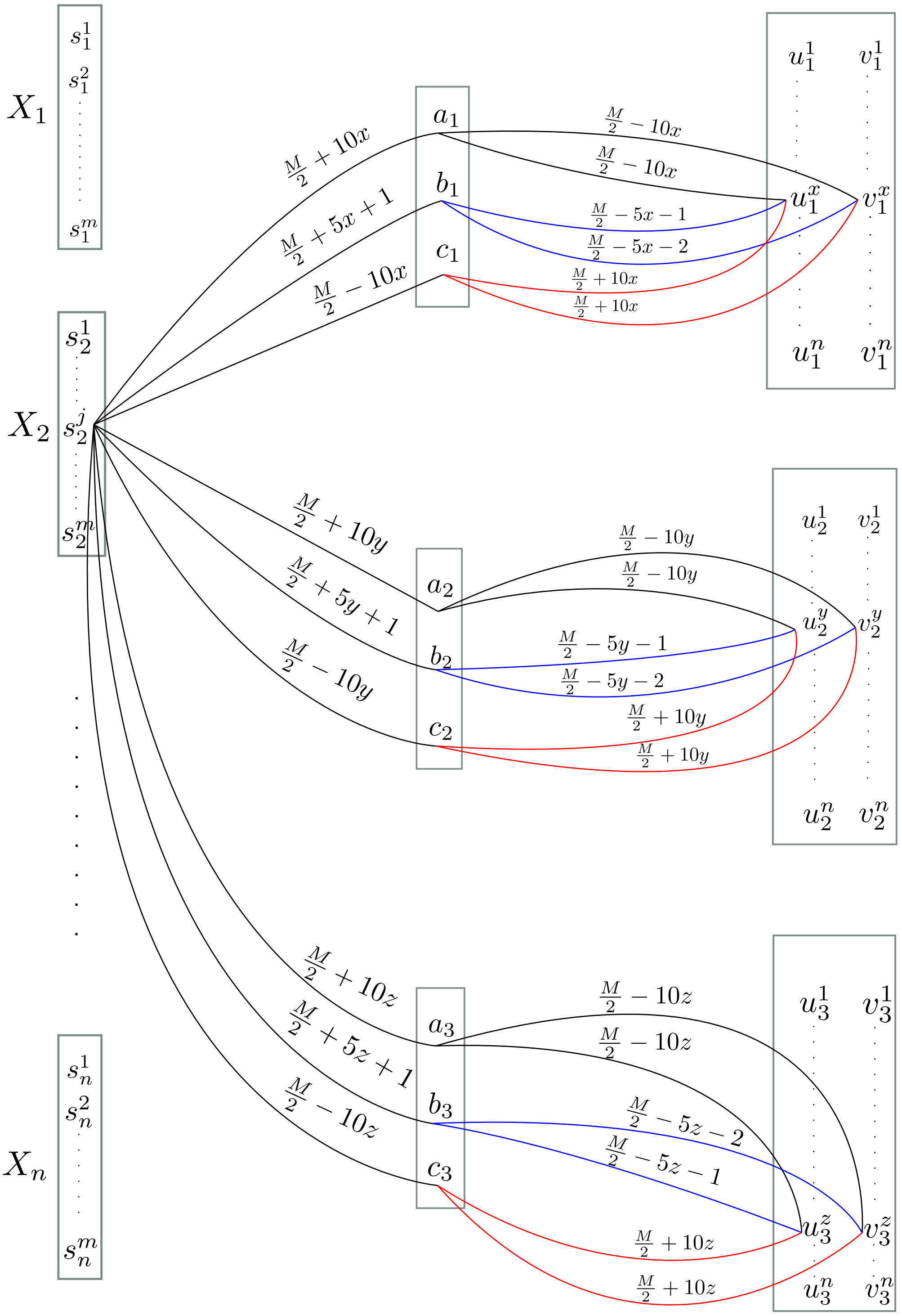

Figure 1 An example of the reduction from 3-Dimensional Matching to n-Multicolored Resolving Set in which $U = \{1,2,3\} \times [n]$ and $\mathcal{F} = \{A_1, ..., A_m\}$. Here we only draw the corresponding paths and resolved pairs of the tuple $A_j = \{(1,x),(2,y),(3,z)\}$.

Proof. ($\Leftarrow$) For an arbitrary tuple $A_i = \{(1,x),(2,y),(3,z)\}$, according to Lemma 2, pairs $\{u_1^x, v_1^x\}$,$\{u_2^y, v_2^y\}$ and $\{u_3^z, v_3^z\}$ are all resolved by s_i^j for every $i \in [n]$. Suppose that the given instance of 3-DIMENSIONAL MATCHING is a yes-instance, that is, there exists $A_{j_1}, ..., A_{j_n}$ satisfying that $\bigcup_{h=1}^{n} A_{j_h} = U$. It follows that $S = \{s_h^{j_h} : h \in [n]\}$ is a solution for the constructed instance of n-MULTICOLORED RESOLVING SET.

($\Rightarrow$) Let $S = \{s_h^{j_h} : h \in [n]\}$ be a solution for the constructed instance of n-MULTICOLORED RESOLVING SET. For an arbitrary pair $\{u_r^x, v_r^x\}$, since it is resolved by some $s_{h'}^{j_{h'}} \in S$, according to Lemma 2, $(r,x) \in A_{j_{h'}}$. As a result, $\{A_{j_h} : h \in [n]\}$ is a solution for the instance of 3-DIMENSIONAL MATCHING. ◀

It is well-known that the treewidth of a graph is bounded by the size of a minimum feedback vertex set of the graph. We can easily observe that W is a feedback vertex set of size 9 for G. It follows that the treewidth of G is at most 10. Then we have the following lemma.

▶ **Lemma 4.** *k-Multicolored Resolving Set is NP-hard even on graphs of treewidth at most* 10.

4 Reduction from Multicolored Resolving Set to Metric Dimension

In this section, we create in polynomial time an instance (G', k) of METRIC DIMENSION, which is equivalent to the instance $(G, n, \chi, \mathcal{P})$ of n-MULTICOLORED RESOLVING SET we created in last section. Roughly speaking, the reduction consists in adding gadgets on base of the constructed instance $(G, n, \chi, \mathcal{P})$ to solve the following two issues: (1) the solution for METRIC DIMENSION could contain vertices not in any set of χ or more than one vertex from some set of χ, which would spoil the desired reduction; (2) we did not make sure that every pair of distinct vertices are resolved by the solution in an instance of n-MULTICOLORED RESOLVING SET. We find that similar strategies to those in [2] can be used to solve these two issues. More specifically, we solve the first issue by adding *forced set gadgets*. One such gadget contains two pairs of vertices such that they are only resolved simultaneously by a vertex of X_i (where it is attached). We solve the second issue by adding *forced vertex gadgets*. One such gadget contains a pair of pendant neighboring vertices (*false twins*), both of which are also adjacent to an identical vertex. Such construction forces at least one vertex of the false twins to be chosen in the solution. The chosen vertices (*forced vertices*) are designed to resolve the remaining unresolved vertex pairs. Besides, we need to add a number of extra paths and set appropriate budget k to make sure that the reduction works as described above.

4.1 Construction of the forced set gadgets

Let $(G, n, \chi, \mathcal{P})$ be an instance of n-MULTICOLORED RESOLVING SET that we created in last section. For every $X_i \in \chi$ ($i \in [n]$), we add two pairs of isolated vertices $\{p_i^1, q_i^1\}$ and $\{p_i^2, q_i^2\}$. Then we add two vertices π_i^1 and π_i^2 such that p_i^1, q_i^1 are adjacent to π_i^1, p_i^2, q_i^2 are adjacent to π_i^2. The vertex triples p_i^1, q_i^1, π_i^1 and p_i^2, q_i^2, π_i^2 ($i \in [n]$) form a forced set gadget. Then we create a path $P(s_i^j, p_i^1)$ of length $20(n+1)$ from s_i^j to p_i^1 and create a path $P(s_i^j, p_i^2)$ of length $20(n+1)$ from s_i^j to p_i^2 for each $i \in [n], j \in [m]$. In order to make sure that a vertex can resolve p_i^1, q_i^1 and p_i^2, q_i^2 simultaneously if and only if it belongs to X_i, we need to create 4 paths of length $20(n+1)$ from π_i^1 to $N_{s_i^j}(s_i^j, a_r)$, from π_i^1 to $N_{s_i^j}(s_i^j, b_r)$, from π_i^1 to $N_{s_i^j}(s_i^j, c_r)$ and from π_i^1 to $N_{s_i^j}(s_i^j, p_i^2)$ respectively for each $i \in [n]$, $j \in [m]$ and $r \in \{1, 2, 3\}$. For simplicity, we name the four paths as $P^1(i, j, a_r)$, $P^1(i, j, b_r)$, $P^1(i, j, c_r)$ and $P^1(i, j, p_i^2)$ respectively.

Symmetrically, we need to create 4 paths of length $20(n+1)$ from π_i^2 to $N_{s_i^j}(s_i^j, a_r)$, from π_i^2 to $N_{s_i^j}(s_i^j, b_r)$, from π_i^2 to $N_{s_i^j}(s_i^j, c_r)$ and from π_i^2 to $N_{s_i^j}(s_i^j, p_i^1)$ respectively for each $i \in [n]$ and $r \in \{1, 2, 3\}$. For simplicity, we name the four paths as $P^2(i, j, a_r)$, $P^2(i, j, b_r)$, $P^2(i, j, c_r)$ and $P^2(i, j, p_i^1)$ respectively. Let $\Pi^h(i, j, r) = \{P^h(i, j, a_r), P^h(i, j, b_r), P^h(i, j, c_r), P^h(i, j, p_i^{3-h})\}$ for $i \in [n], j \in [m], r \in \{1, 2, 3\}, h \in \{1, 2\}$.

This completes the construction of the first phase.

4.2 Construction of the forced vertex gadgets

A forced vertex gadget consists of a triangle, namely three vertices such that each vertex is adjacent to the other two vertices. Two vertices of the triangle are false twins whose degrees are exactly 2 and we call the other vertex in the triangle the *connecting vertex* of the gadget. When we say that we add a forced vertex gadget F to a vertex v, we mean that we create a forced vertex gadget F such that v is identified with the connecting vertex of F. For each $i \in [n], j \in [m], r \in \{1, 2, 3\}, h \in \{1, 2\}$, we add a forced vertex gadget $F^h(i, j, a_r)$ to $N^h_{\pi_h}(i, j, a_r)$, $F^h(i, j, b_r)$ to $N^h_{\pi_h}(i, j, b_r)$, $F^h(i, j, c_r)$ to $N^h_{\pi_h}(i, j, c_r)$ and $F^h(i, j, p_i^{3-h})$ to $N^h_{\pi_h}(i, j, , p_i^{3-h})$ respectively.

In order to make sure that the false twins of $F^h(i, j, b_r)$ for $i \in [n], j \in [m], r \in \{1, 2, 3\}, h \in \{1, 2\}$ do not resolve any vertex pair of $\mathcal{P}$, we create a path $P(\pi_i^h, a_r)$ and a path $P(\pi_i^h, c_r)$ both of length $10(n+1)$ for $i \in [n]$, $h \in \{1, 2\}$ and $r \in \{1, 2, 3\}$.

For each $i \in [n], j \in [m], r \in \{1, 2, 3\}, h \in \{1, 2\}$, we add a forced vertex gadget $F(\pi_i^h, a_r)$ to $N_{a_r}(\pi_i^h, a_r)$ and a forced vertex gadget $F(\pi_i^h, c_r)$ to $N_{c_r}(\pi_i^h, c_r)$. For each $i \in [n], j \in [m], r \in \{1, 2, 3\}$, we add a forced vertex gadget $F(s_i^j, a_r)$ to $N_{a_r}(s_i^j, a_r)$ and a forced vertex gadget $F(s_i^j, c_r)$ to $N_{c_r}(s_i^j, c_r)$.

Let $\text{mid}(P^h(i, j, p_i^{3-h}))$ be the middle vertex of $P^h(i, j, p_i^{3-h})$ for $i \in [n], j \in [m], h \in \{1, 2\}$. In order to make sure that the false twins of $F^h(i, j, p_i^{3-h})$ do not resolve the vertex pair $\{p_i^{3-h}, q_i^{3-h}\}$, create a path $P(q_i^h, \text{mid}(P^{3-h}(i, j, p_i^h)))$ from q_i^h to $\text{mid}(P^{3-h}(i, j, p_i^h))$ of length $|P^{3-h}(i, j, p_i^h)|/2 + |P(s_i^j, p_i^h)| - 1$. Then add a forced vertex gadget $F^{mid}(i, j, h)$ to $\text{mid}(P^h(i, j, p_i^{3-h}))$.

For $i \in [n], j \in [m], r \in \{1, 2, 3\}, h \in \{1, 2\}$, add a forced vertex gadget $F^{ecc}(i, j, h, r)$ to the vertex $x \in P^h(i, j, a_r)$ such that $\text{dist}(\pi_i^h, x) = 10(n+1) + 1$.

For each $i \in [n], r \in \{1, 2, 3\}$, create two forced vertex gadgets $F^1(u_r^i, v_r^i)$ and $F^2(u_r^i, v_r^i)$ for the vertex pair $\{u_r^i, v_r^i\} \in \mathcal{P}_r$. Then create an edge from the connecting vertex of $F^1(u_r^i, v_r^i)$ to u_r^i, to v_r^i, to $N_{u_r^i}(a_r, u_r^i)$ and to $N_{u_r^i}(c_r, u_r^i)$ respectively for $i \in [n], r \in \{1, 2, 3\}$. Create an edge from the connecting vertex of $F^2(u_r^i, v_r^i)$ to u_r^i, to v_r^i, to the vertex x such that $x \in P(a_r, u_r^i)$ and $\text{dist}(x, u_r^i) = 2$, and to the vertex y such that $y \in P(c_r, u_r^i)$ and $\text{dist}(y, u_r^i) = 2$. This completes the construction of the second phase.

Finally, let G' be the graph constructed by above two phases and set $k = 34nm + 19n$. This finishes constructing the instance (G', k) of Metric Dimension. Figure 2 shows a part of G'.

4.3 Soundness of the reduction

First, we define the vertex sets to be used in the following parts. Recall that $X_i = \{s_i^1, ..., s_i^m\}$. For every $i \in [n], r \in \{1, 2, 3\}, h \in \{1, 2\}$, let

$$U_i^h = \bigcup_{j \in [m]} P(s_i^j, p_i^h),$$

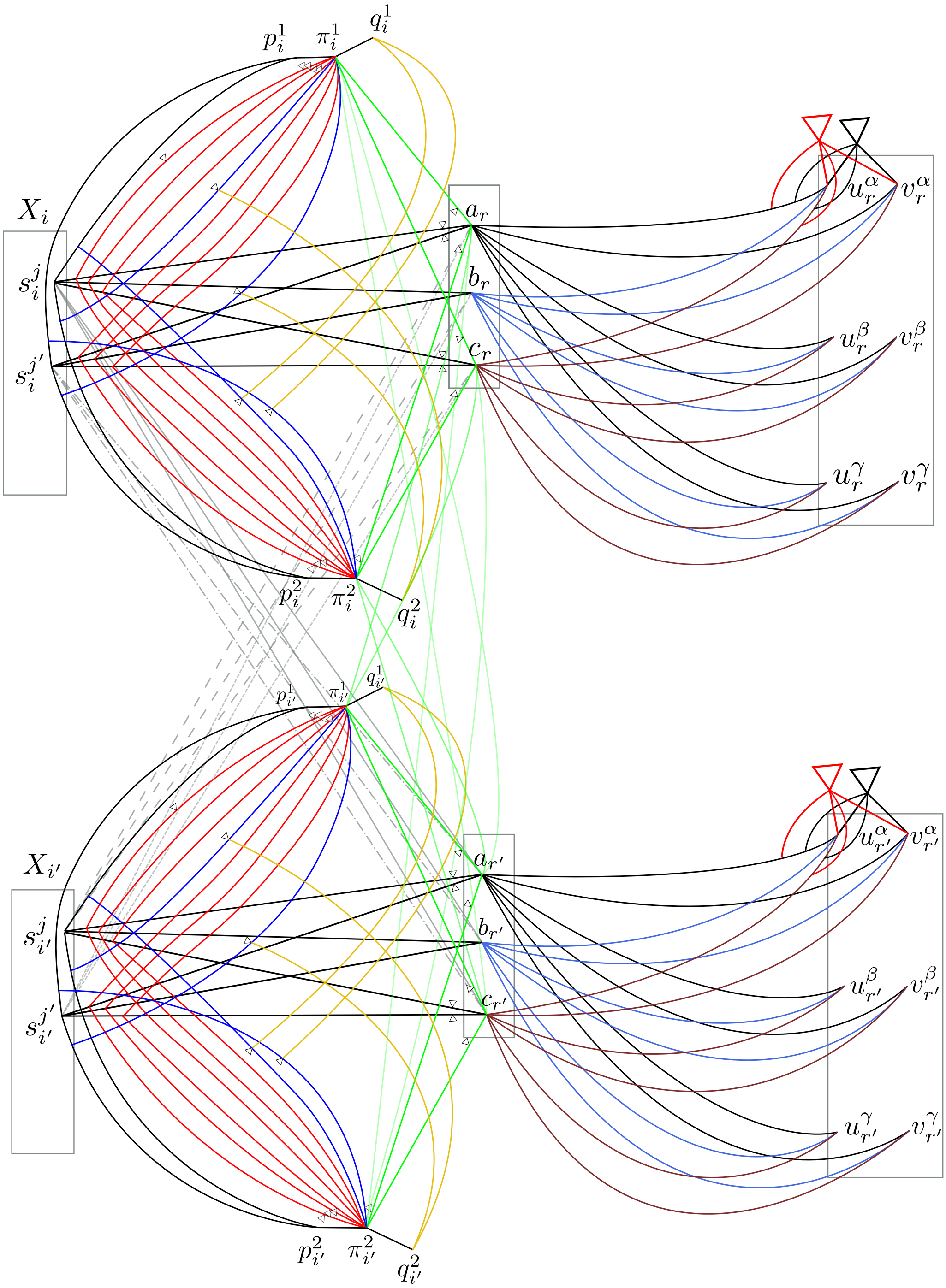

Figure 2 An example showing a part of G'. Triangles represent corresponding forced vertex gadgets. For clarity, some forced vertex gadgets do not appear on the figure. Dotted or dashed lines are used in order for cleanness of the figure.

$$H_{i,r} = \bigcup_{j\in[m]} P(s_i^j, a_r) \cup P(s_i^j, b_r) \cup P(s_i^j, c_r),$$

$$S_i^h = \bigcup_{r\in\{1,2,3\}} P(\pi_i^h, a_r) \cup P(\pi_i^h, c_r),$$

$$L_i^h = \bigcup_{j\in[m]} P(q_i^h, \mathrm{mid}(P^{3-h}(i,j,p_i^h))),$$

$$R_r = \bigcup_{i\in[n]} P(a_r, u_r^i) \cup P(a_r, v_r^i) \cup P(b_r, u_r^i) \cup P(b_r, v_r^i) \cup P(c_r, u_r^i) \cup P(c_r, v_r^i), \text{ and}$$

$$\Pi^h(i,j,r) = P^h(i,j,a_r) \cup P^h(i,j,b_r) \cup P^h(i,j,c_r) \cup P^h(i,j,p_i^{3-h}).$$

For every $i \in [n]$, let

$$U_i = \bigcup_{h\in\{1,2\}} U_i^h \qquad H_i = \bigcup_{r\in\{1,2,3\}} H_{i,r} \qquad S_i = \bigcup_{h\in\{1,2\}} S_i^h$$

$$L_i = \bigcup_{h\in\{1,2\}} L_i^h \qquad \Pi_i = \bigcup_{j\in[m],r\in\{1,2,3\},h\in\{1,2\}} \Pi^h(i,j,r).$$

Let $\mathcal{F}$ be the union of all forced vertex gadgets, i.e.

$$\begin{aligned}\mathcal{F} = \bigcup_{i\in[n],j\in[m],r\in\{1,2,3\},h\in\{1,2\}} &(F(s_i^j, a_r) \cup F(s_i^j, c_r) \cup F(\pi_i^h, a_r) \cup F(\pi_i^h, c_r)\\ &\cup F^h(u_r^i, v_r^i) \cup F^h(i,j,a_r) \cup F^h(i,j,b_r) \cup F^h(i,j,c_r)\\ &\cup F^h(i,j,p_i^{3-h}) \cup F^{mid}(i,j,h) \cup F^{ecc}(i,j,h,r)).\end{aligned}$$

Next we introduce a lemma about forced set gadgets and this lemma is important for the correctness of the reduction.

▶ **Lemma 5.** *The following three statements are true for the instance (G', k).*

(a) *The vertex s_i^j for $i \in [n], j \in [m]$ resolves both pairs $\{p_i^1, q_i^1\}$ and $\{p_i^2, q_i^2\}$. Moreover, s_i^j does not resolve any vertex pair $\{p_{i'}^h, q_{i'}^h\}$ such that $i' \in [n], h \in \{1,2\}$ and $i' \neq i$.*

(b) *The vertices of any forced vertex gadget do not resolve any vertex pair of $\{\{p_i^h, q_i^h\} \mid i \in [n], h \in \{1,2\}\}$.*

(c) *Any vertex $v \in V(G') \setminus (X_i \cup \mathcal{F})$ resolves at most one vertex pair of $\{\{p_i^h, q_i^h\} \mid i \in [n], h \in \{1,2\}\}$.*

Proof. By the construction of G', $\mathrm{dist}(s_i^j, q_i^h) = |P(s_i^j, p_i^h)| + 2 = 20(n+1) + 2 > \mathrm{dist}(s_i^j, p_i^h)$ for $i \in [n], j \in [m]$ and $h \in \{1,2\}$. Thus any vertex of X_i resolves both pairs $\{p_i^1, q_i^1\}$ and $\{p_i^2, q_i^2\}$ for $i \in [n]$. For a vertex pair $\{p_{i'}^{h'}, q_{i'}^{h'}\}$ such that $i' \neq i$, there is a shortest path from s_i^j to $p_{i'}^{h'}$ or $q_{i'}^{h'}$ going through $c_{r'}$ and $\pi_{i'}^{h'}$ with some integer $r' \in \{1,2,3\}$. Thus a vertex $s \in X_i$ resolves exactly two vertex pairs of $\{\{p_i^h, q_i^h\} : i \in [n], h \in \{1,2\}\}$.

First we claim that vertices of $\mathcal{F}$ do not resolve any vertex pair $\{p_{i'}^{h'}, q_{i'}^{h'}\}$ for $i' \in [n], h' \in \{1,2\}$. For any vertex $v \in F^h(u_r^i, v_r^i)$ for $i \in [n], r \in \{1,2,3\}, h \in \{1,2\}$, there is a shortest path from v to $p_{i'}^{h'}$ or $q_{i'}^{h'}$ going through a_r and $\pi_{i'}^{h'}$. Thus v does not resolve any vertex pair $\{p_{i'}^{h'}, q_{i'}^{h'}\}$ for $i' \in [n], h' \in \{1,2\}$. For any vertex $v \in F^{mid}(i,j,h) \cup F^{ecc}(i,j,h,r)$ for $i \in [n], j \in [m], h \in \{1,2\}, r \in \{1,2,3\}$, we can see that $\mathrm{dist}(v, p_{i'}^{h'}) = \mathrm{dist}(v, q_{i'}^{h'})$ with $i' = i$. There is a shortest path from v to $p_{i'}^{h'}$ or $q_{i'}^{h'}$ going through π_i^h, a_r and $\pi_{i'}^{h'}$ with $i' \neq i$. Thus v does not resolve any vertex pair $\{p_{i'}^{h'}, q_{i'}^{h'}\}$ for $i' \in [n], h' \in \{1,2\}$. For any vertex $v \in \mathcal{F} \setminus \bigcup_{i\in[n],j\in[m],r\in\{1,2,3\},h\in\{1,2\}}(F^h(u_r^i, v_r^i) \cup F^{ecc}(i,j,h,r) \cup F^{mid}(i,j,h))$, there is a shortest path from v to $p_{i'}^{h'}$ or $q_{i'}^{h'}$ going through $\pi_{i'}^{h'}$ with $i' = i$. There is a shortest path from v to $p_{i'}^{h'}$ or $q_{i'}^{h'}$ going through c_r (or a_r) and $\pi_{i'}^{h'}$ with $i' \neq i$. Thus v does not resolve any pair $\{p_{i'}^{h'}, q_{i'}^{h'}\}$. As a result, vertices of $\mathcal{F}$ do not resolve any vertex pair $\{p_{i'}^{h'}, q_{i'}^{h'}\}$ for $i' \in [n], h' \in \{1,2\}$.

Then we show that any vertex $v \in V(G') \setminus (X_i \cup \mathcal{F})$ resolves at most one pair of $\{p_i^1, q_i^1\}$ and $\{p_i^2, q_i^2\}$.

For a vertex $v \in U_i^h \setminus X_i$ for $i \in [n], h \in \{1,2\}$, $\mathrm{dist}(v, p_i^h) = \mathrm{dist}(v, q_i^h) - 2 < \mathrm{dist}(v, q_i^h)$. $\mathrm{dist}(v, q_i^{3-h}) = \mathrm{dist}(v, N_{s_i^j}(s_i^j, p_i^h)) + |P^{3-h}(i,j,p_i^h)| + 1 = \mathrm{dist}(v, p_i^{3-h})$. For a vertex pair $\{p_{i'}^{h'}, q_{i'}^{h'}\}$ such that $i' \neq i$, there is a shortest path from v to $p_{i'}^{h'}$ or $q_{i'}^{h'}$ going through $\pi_{i'}^{h'}$. Thus $v \in U_i^h \setminus X_i$ for $i \in [n], h \in \{1,2\}$ resolves exactly one vertex pair of $\{\{p_i^h, q_i^h\} : i \in [n], h \in \{1,2\}\}$.

Let $P(\mathrm{mid}(P^{3-h}(i,j,p_i^h)), N_{s_i^j}(s_i^j, p_i^h))$ be the subpath of $P^{3-h}(i,j,p_i^h)$ from $\mathrm{mid}(P^{3-h}(i,j,p_i^h))$ to $N_{s_i^j}(s_i^j, p_i^h)$. Let $\Lambda_i^h = (\bigcup_{j\in[m]} P(\mathrm{mid}(P^{3-h}(i,j,p_i^h)), N_{s_i^j}(s_i^j, p_i^h))) \setminus \{\mathrm{mid}(P^{3-h}(i,j,p_i^h)) \mid j \in [m]\}$. For a vertex $v \in \Lambda_i^h$ for $i \in [n], h \in \{1,2\}$, $\mathrm{dist}(v, p_i^h) = \mathrm{dist}(v, q_i^h) - 2 < \mathrm{dist}(v, q_i^h)$. $\mathrm{dist}(v, q_i^{3-h}) = \mathrm{dist}(v, \pi_i^{3-h}) + 1 = \mathrm{dist}(v, p_i^{3-h})$. For a vertex pair $\{p_{i'}^{h'}, q_{i'}^{h'}\}$ such that $i' \neq i$, there is a shortest path from v to $p_{i'}^{h'}$ or $q_{i'}^{h'}$ going through $\pi_{i'}^{h'}$. Thus $v \in \Lambda_i^h$ for $i \in [n], h \in \{1,2\}$ resolves exactly one vertex pair of $\{\{p_i^h, q_i^h\} : i \in [n], h \in \{1,2\}\}$.

For a vertex $v \in L_i^h \setminus \{\mathrm{mid}(P^h(i,j,p_i^{3-h})) \mid j \in [m]\}$ for $i \in [n], h \in \{1,2\}$, $\mathrm{dist}(v, q_i^h) = \mathrm{dist}(v, p_i^h) - 2 < \mathrm{dist}(v, p_i^h)$. There is a shortest path from v to p_i^{3-h} or q_i^{3-h} going through π_i^{3-h}. For a vertex pair $\{p_{i'}^{h'}, q_{i'}^{h'}\}$ such that $i' \neq i$, there is a shortest path from v to $p_{i'}^{h'}$ or $q_{i'}^{h'}$ going through $\pi_{i'}^{h'}$. Thus $v \in L_i^h \setminus \{\mathrm{mid}(P^h(i,j,p_i^{3-h})) \mid j \in [m]\}$ for $i \in [n], h \in \{1,2\}$ resolves exactly one vertex pair of $\{\{p_i^h, q_i^h\} : i \in [n], h \in \{1,2\}\}$.

For a vertex $v \in \Pi_i \cup S_i \cup H_i \setminus (X_i \cup \Lambda_i^1 \cup \Lambda_i^2)$ for $i \in [n]$, there is a shortest path from v to $p_{i'}^{h'}$ or $q_{i'}^{h'}$ going through $\pi_{i'}^{h'}$ with $i = i', h' \in \{1,2\}$. For a vertex pair $\{p_{i'}^{h'}, q_{i'}^{h'}\}$ such that $i' \neq i$, there is a shortest path from v to $p_{i'}^{h'}$ or $q_{i'}^{h'}$ going through $\pi_{i'}^{h'}$. Thus v does not resolve any vertex pair of $\{\{p_i^h, q_i^h\} : i \in [n], h \in \{1,2\}\}$.

For a vertex $v \in R_r$ for $r \in \{1,2,3\}$, there is a shortest path from v to p_i^h or q_i^h for $i \in [n], h \in \{1,2\}$ going through a_r and π_i^h. Thus v does not resolve any vertex pair of $\{\{p_i^h, q_i^h\} : i \in [n], h \in \{1,2\}\}$. This completes the proof for the lemma. ◀

By the properties of false twins, we need to choose exactly one vertex of the false twins (arbitrarily) of every forced vertex gadget in the resolving set of G', which we call a *forced vertex*. For convenience, we use $f(\cdot)$ to represent the chosen forced vertex of the corresponding gadget $F(\cdot)$. Then we have the following lemma.

▶ **Lemma 6.** *The forced vertices do not resolve any vertex pair* $\{u_r^i, v_r^i\} \in \mathcal{P}$ *for* $r \in \{1,2,3\}$ *and* $i \in [n]$.

Proof. We fix arbitrary integers $i \in [n], j \in [m], r \in \{1,2,3\}, h \in \{1,2\}$. For the forced vertex $f^h(i,j,a_r)$, $\mathrm{dist}(f^h(i,j,a_r), u_{r'}^{i'}) = 2 + |P(\pi_i^h, a_{r'})| + |P(a_{r'}, u_{r'}^{i'})| = 2 + |P(\pi_i^h, a_{r'})| + |P(a_{r'}, v_{r'}^{i'})| = \mathrm{dist}(f^h(i,j,a_r), v_{r'}^{i'})$ for $i' \in [n], r' \in \{1,2,3\}$. Thus $f^h(i,j,a_r)$ does not resolve any vertex pair of $\mathcal{P}$. Similarly, the forced vertices $f^h(i,j,b_r)$, $f^h(i,j,c_r)$ and $f^h(i,j,p_i^{3-h})$ do not resolve any vertex pair of $\mathcal{P}$. For the forced vertex $f^{mid}(i,j,h)$, $\mathrm{dist}(f^{mid}(i,j,h), u_{r'}^{i'}) = \mathrm{dist}(f^{mid}(i,j,h), v_{r'}^{i'}) = |P^h(i,j,p_i^{3-h})|/2 + |P(\pi_i^h, a_{r'})| + |P(a_{r'}, u_{r'}^{i'})|$. Thus $f^{mid}(i,j,h)$ does not resolve any vertex pair of $\mathcal{P}$. For the forced vertex $f^{ecc}(i,j,h,r)$, $\mathrm{dist}(f^{ecc}(i,j,h,r), u_{r'}^{i'}) = \mathrm{dist}(f^{ecc}(i,j,h,r), v_{r'}^{i'}) = 10(n+1) + 1 + |P(\pi_i^h, a_{r'})| + |P(a_{r'}, u_{r'}^{i'})|$. Thus $f^{ecc}(i,j,h,r)$ does not resolve any vertex pair of $\mathcal{P}$.

We fix arbitrary integers $i \in [n], j \in [m], r \in \{1,2,3\}$. For the forced vertex $f(s_i^j, a_r)$, $\mathrm{dist}(f(s_i^j, a_r), u_r^{i'}) = 2 + |P(a_r, u_r^{i'})| = 2 + |P(a_r, v_r^{i'})| = \mathrm{dist}(f(s_i^j, a_r), v_r^{i'})$ for $i' \in [n]$. For the forced vertex $f(s_i^j, c_r)$, $\mathrm{dist}(f(s_i^j, c_r), u_r^{i'}) = 2 + |P(c_r, u_r^{i'})| = 2 + |P(c_r, v_r^{i'})| =$

$\mathrm{dist}(f(s_i^j, c_r), v_r^{i'})$ for $i' \in [n]$. Thus $f(s_i^j, a_r)$ and $f(s_i^j, c_r)$ do not resolve any vertex pair of $\mathcal{P}_r$. Similarly, $f(\pi_i^h, a_r)$ and $f(\pi_i^h, c_r)$ for $i \in [n], h \in \{1,2\}, r \in \{1,2,3\}$ do not resolve any vertex pair of $\mathcal{P}_r$. For vertex pairs of $\mathcal{P}_{r'}$ with $r' \in \{1,2,3\}$ and $r' \neq r$, $\mathrm{dist}(f(s_i^j, a_r), u_{r'}^{i'}) = 2+|P(a_r, \pi_i^1)|+|P(a_{r'}, \pi_i^1)|+|P(a_{r'}, u_r^{i'})| = 2+|P(a_r, \pi_i^1)|+|P(a_{r'}, \pi_i^1)|+|P(a_{r'}, v_r^{i'})| = \mathrm{dist}(f(s_i^j, a_r), v_{r'}^{i'})$ for $i' \in [n]$. $\mathrm{dist}(f(s_i^j, c_r), u_{r'}^{i'}) = 2 + |P(c_r, \pi_i^1)| + |P(a_{r'}, \pi_i^1)| + |P(a_{r'}, u_r^{i'})| = 2 + |P(c_r, \pi_i^1)| + |P(a_{r'}, \pi_i^1)| + |P(a_{r'}, v_r^{i'})| = \mathrm{dist}(f(s_i^j, a_r), v_{r'}^{i'})$ for $i' \in [n]$. Thus $f(s_i^j, a_r)$ and $f(s_i^j, c_r)$ do not resolve any vertex pair of $\mathcal{P}_{r'}$.

We fix arbitrary integers $i \in [n], r \in \{1,2,3\}$. For the forced vertex $f^1(u_r^i, v_r^i)$ or $f^2(u_r^i, v_r^i)$, obviously it does not resolve the vertex pair $\{u_r^i, v_r^i\}$. For a vertex pair $\{u_r^{i'}, v_r^{i'}\}$ with $i' \in [n]$ and $i' \neq i$, $\mathrm{dist}(f^1(u_r^i, v_r^i), u_r^{i'}) = 2 + |P(a_r, u_r^i)| - 1 + |P(a_r, u_r^{i'})| = 2 + |P(a_r, u_r^i)| - 1 + |P(a_r, v_r^{i'})| = \mathrm{dist}(f^1(u_r^i, v_r^i), v_r^{i'})$. For a vertex pair $\{u_{r'}^{i'}, v_{r'}^{i'}\}$ with $i' \in [n]$ and $r' \in \{1,2,3\}$ and $r' \neq r$, $\mathrm{dist}(f^1(u_r^i, v_r^i), u_{r'}^{i'}) = 2 + |P(a_r, u_r^i)| - 1 + |P(\pi_i^1, a_r)| + |P(\pi_i^1, a_{r'})| + |P(a_{r'}, u_{r'}^{i'})| = \mathrm{dist}(f^1(u_r^i, v_r^i), v_{r'}^{i'})$. As a result, $f^1(u_r^i, v_r^i)$ does not resolve any vertex pair of $\mathcal{P}$. For a vertex pair $\{u_r^{i'}, v_r^{i'}\}$ with $i' \in [n]$ and $i' \neq i$, $\mathrm{dist}(f^2(u_r^i, v_r^i), u_r^{i'}) = 2 + |P(a_r, u_r^i)| - 2 + |P(a_r, u_r^{i'})| = 2 + |P(a_r, u_r^i)| - 2 + |P(a_r, v_r^{i'})| = \mathrm{dist}(f^2(u_r^i, v_r^i), v_r^{i'})$. For a vertex pair $\{u_{r'}^{i'}, v_{r'}^{i'}\}$ with $i' \in [n]$, $r' \in \{1,2,3\}$ and $r' \neq r$, $\mathrm{dist}(f^2(u_r^i, v_r^i), u_{r'}^{i'}) = 2+|P(a_r, u_r^i)|-2+|P(\pi_i^1, a_r)|+|P(\pi_i^1, a_{r'})|+|P(a_{r'}, u_{r'}^{i'})| = \mathrm{dist}(f^2(u_r^i, v_r^i), v_{r'}^{i'})$. As a result, $f^2(u_r^i, v_r^i)$ does not resolve any vertex pair of $\mathcal{P}$. This completes the proof for the lemma. ◀

▶ **Lemma 7** (Soundness). *If G' has a resolving set of size at most $34nm + 19n$, then $(G, n, \chi, \mathcal{P})$ is a yes-instance.*

Proof. Suppose that $\mathcal{S}$ is a resolving set for G' of size at most $34nm + 19n$. Let $\hat{\mathcal{S}} = \mathcal{S} \cap X$. (Recall that $X = \bigcup_{i=1}^{n} \{s_i^1, ..., s_i^m\}$.) We claim that $\hat{\mathcal{S}}$ is solution for $(G, n, \chi, \mathcal{P})$. Note that for the false twins $\{u, u'\}$ of a forced vertex gadget, no vertex resolves the vertex pair $\{u, u'\}$ except u (or u'). It follows that $\mathcal{S}$ contains $34nm + 18n$ forced vertices since there are $34nm + 18n$ forced vertex gadgets in G'. Since X has no intersection with the vertex set of all forced vertex gadgets, $|\hat{\mathcal{S}}| \leq n$. By Lemma 5, we get that $|\hat{\mathcal{S}} \cap X_i| = 1$ for each $i \in [n]$. Thus $|\hat{\mathcal{S}}| = n$. By Lemma 6 and the assumption that $\mathcal{S}$ is a resolving set for G', $\hat{\mathcal{S}}$ resolves every pair $\{u_r^i, v_r^i\}$ in G' for $r \in \{1,2,3\}$ and $i \in [n]$. We can check that the distance between s_i^j and $u_r^{i'}$ in G' (and the distance between s_i^j and $v_r^{i'}$ in G') for $i \in [n], j \in [m], i' \in [n], r \in \{1,2,3\}$ is the same as that in G. Thus $\hat{\mathcal{S}}$ is a solution for $(G, n, \chi, \mathcal{P})$. ◀

4.4 Treewidth bound of the graph

Since the completeness proof takes a large amount of space, before proceeding to that, we first show that G' is of constant treewidth. In fact, we will prove a slightly stronger statement that G' is of constant pathwidth by giving a search strategy with a constant number of searchers.

▶ **Lemma 8.** *The pathwidth of G' is at most 24.*

Proof. Following the characterization of pathwidth by Kirousis and Papadimitriou [11], we give a search strategy with 25 searchers. First, we put 9 searchers on $\bigcup_{r \in \{1,2,3\}} \{a_r, b_r, c_r\}$. The 9 searchers remain there until the end of the whole searching process. The searching process consists of two phases. We search the "left" part of G' in the first phase and the "right" part of G' in the second phase.

The first phase of the searching process consists of n rounds. At the beginning of the i-th round ($i \in [n]$), we put 6 searchers on $\bigcup_{h \in \{1,2\}} \{p_i^h, q_i^h, \pi_i^h\}$. Here when we say that we clean a path, this means that there are already two searchers guarding at the endpoints (or the neighbor of the endpoints) of this path and we use 3 extra searchers x, y, z such that x, y move alternately from one end of the path to the other end to clean the edges of the path. When a searcher, say x arrives at the connecting point of a forced vertex gadget, we put y, z on the false twins of this forced vertex gadget to clean the edges of this gadget and then after removing y, z, put y ahead of x to continue the alternating process unless x reaches the endpoint of this path. Then for each $j \in [m]$, we

- put 5 vertices on $N_{G'}(s_i^j)$.
- put 2 vertices on $\text{mid}(P^h(i,j,p_i^{3-h}))$ for $h \in \{1,2\}$.
- use 3 extra searchers to clean the paths $P(s_i^j, p_i^h)$ for $h \in \{1,2\}$, the paths $P(s_i^j, a_r)$, $P(s_i^j, b_r)$, $P(s_i^j, c_r)$ for $r \in \{1,2,3\}$, the paths $P^h(i,j,a_r)$, $P^h(i,j,b_r)$, $P^h(i,j,c_r)$, $P^h(i,j,p_i^{3-h})$ for $h \in \{1,2\}, r \in \{1,2,3\}$, the paths $P(\pi_i^h, a_r)$, $P(\pi_i^h, c_r)$ for $h \in \{1,2\}, r \in \{1,2,3\}$ and the path $P(q_i^h, \text{mid}(P^{3-h}(i,j,p_i^h)))$ for $h \in \{1,2\}$ successively (including all forced vertex gadgets attached to the vertices on these paths).
- remove the above 10 searchers that are still on the graph.

At the end of the i-th round, we remove the 6 searchers on $\bigcup_{h \in \{1,2\}} \{p_i^h, q_i^h, \pi_i^h\}$.

The second phase of the searching process consists of 3 rounds. During the r-th round ($r \in \{1,2,3\}$), we operate as follows. For each $i \in [n]$, we

- put 4 searchers on u_r^i, v_r^i and the connecting point of $F^h(u_r^i, v_r^i)$ for $h \in \{1,2\}$.
- use 2 extra searchers to clean the paths $P(a_r, u_r^i), P(b_r, u_r^i), P(c_r, u_r^i), P(a_r, v_r^i), P(b_r, v_r^i)$ and $P(c_r, v_r^i)$ (including the forced vertex gadgets $F^h(u_r^i, v_r^i)$ for $h \in \{1,2\}$ and the incident edges of the connecting vertex of $F^h(u_r^i, v_r^i)$).
- remove the above 6 searchers that are still on the graph.

This completes the description of the the search strategy.

As a result, the node search number of G' is at most 25. It follows that the pathwidth of G' is bounded by 24. ◀

4.5 Completeness of the reduction

For every forced vertex gadget of G', we choose a vertex from the false twins arbitrarily as a forced vertex and let the set of all chosen forced vertices be F. In this section, we show that if $(G, n, \chi, \mathcal{P})$ has a solution $\mathcal{S}$, then $\mathcal{S}' = \mathcal{S} \cup F$ is a resolving set of size at most $34nm + 19n$ for G'. Formally, we will prove the following lemma.

▶ **Lemma 9** (Completeness). *If $(G, n, \chi, \mathcal{P})$ is a yes-instance, then G' has a resolving set of size at most $34nm + 19n$.*

The proof of Lemma 9 consists of a list of lemmas. Suppose that $V(G') = V_1 \cup V_2 \cup ... \cup V_t$. Our general method is to show that for each $i \in [t]$, every internal vertex pair of V_i is resolved by $\mathcal{S}'$ and every vertex pair of $V_{i'} \times V_i$ for each $i' < i$ is resolved by $\mathcal{S}'$. Note that when we mention the vertex pairs of $V_{i'} \times V_i$, we ignore the vertex pairs with two identical vertices by default as it's meaningless in our problem. Due to the page constraint, the proof of Lemma 9 is deferred to the full version of the paper.

5 Conclusion

In this paper, we show that METRIC DIMENSION is NP-hard on graphs of treewidth at most 24. One of the key points in bounding the treewidth of G' is to maintain a vertex separation of constant size. In the first step of our construction, we need 9 vertices to be the vertex separation and convey the choice of the vertices in each color class X_i ($i \in [n]$). It seems hard to show NP-hardness of this problem on graphs of treewidth bounded by a constant

$c \leq 9$ using the techniques in this paper, so we mention this open problem again: is METRIC DIMENSION polynomial-time solvable on graphs of treewidth 2 or series-parallel graphs [1]? Another direction is about the parameterized complexity of METRIC DIMENSION. We ask the following two questions. Is METRIC DIMENSION FPT parameterized by the size of the resolving set on constant treewidth graph? Is METRIC DIMENSION FPT parameterized by both the size of the resolving set and the treewidth of the input graph?

References

1 Rémy Belmonte, Fedor V. Fomin, Petr A. Golovach, and M. S. Ramanujan. Metric dimension of bounded tree-length graphs. *SIAM J. Discret. Math.*, 31(2):1217–1243, 2017. doi:10.1137/16M1057383.

2 Édouard Bonnet and Nidhi Purohit. Metric dimension parameterized by treewidth. In Bart M. P. Jansen and Jan Arne Telle, editors, *14th International Symposium on Parameterized and Exact Computation, IPEC 2019, September 11-13, 2019, Munich, Germany*, volume 148 of *LIPIcs*, pages 5:1–5:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPIcs.IPEC.2019.5.

3 Josep Díaz, Olli Pottonen, Maria J. Serna, and Erik Jan van Leeuwen. Complexity of metric dimension on planar graphs. *J. Comput. Syst. Sci.*, 83(1):132–158, 2017. doi:10.1016/j.jcss.2016.06.006.

4 David Eppstein. Metric dimension parameterized by max leaf number. *J. Graph Algorithms Appl.*, 19(1):313–323, 2015. doi:10.7155/jgaa.00360.

5 Leah Epstein, Asaf Levin, and Gerhard J. Woeginger. The (weighted) metric dimension of graphs: Hard and easy cases. *Algorithmica*, 72(4):1130–1171, 2015. doi:10.1007/s00453-014-9896-2.

6 Henning Fernau, Pinar Heggernes, Pim van 't Hof, Daniel Meister, and Reza Saei. Computing the metric dimension for chain graphs. *Inf. Process. Lett.*, 115(9):671–676, 2015. doi:10.1016/j.ipl.2015.04.006.

7 Frank Harary and Robert A Melter. On the metric dimension of a graph. *Ars Combin*, 2(1):191–195, 1976.

8 Sepp Hartung and André Nichterlein. 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*, pages 266–276. IEEE Computer Society, 2013. doi:10.1109/CCC.2013.36.

9 Richard M Karp. Reducibility among combinatorial problems. In *Complexity of computer computations*, pages 85–103. Springer, 1972.

10 Samir Khuller, Balaji Raghavachari, and Azriel Rosenfeld. Landmarks in graphs. *Discret. Appl. Math.*, 70(3):217–229, 1996. doi:10.1016/0166-218X(95)00106-2.

11 Lefteris M Kirousis and Christos H Papadimitriou. Interval graphs and seatching. *Discrete Mathematics*, 55(2):181–184, 1985.

12 Peter J Slater. Leaves of trees. *Congr. Numer*, 14(37):549–559, 1975.