Parameterized Complexity of Isometric Path Partition: Treewidth and Diameter

Dibyayan Chakraborty
School of Computer Science, University of Leeds, UK

Oscar Defrain
LIS, Aix-Marseille University, CNRS, France

Florent Foucaud
Université Clermont Auvergne, CNRS, Mines Saint-Étienne, Clermont Auvergne INP, LIMOS, 63000 Clermont-Ferrand, France

Mathieu Mari
LIRMM, Université de Montpellier, CNRS, Montpellier, France

Prafullkumar Tale
Indian Institute of Science Education and Research Pune, India

Abstract

In the Isometric Path Partition problem, the input is a graph G with n vertices and an integer k, and the objective is to determine whether the vertices of G can be partitioned into k vertex-disjoint shortest paths. We investigate the parameterized complexity of the problem when parameterized by the treewidth (tw) of the input graph, arguably one of the most widely studied parameters. Courcelle's theorem [Information & Computation, 1990] shows that graph problems that are expressible as MSO formulas of constant size admit FPT algorithms parameterized by the treewidth of the input graph. This encompasses many natural graph problems. However, many metric-based graph problems, where the solution is defined using some metric-based property of the graph (often the distance) are not expressible as MSO formulas of constant size. These types of problems, Isometric Path Partition being one of them, require individual attention and often draw the boundary for the success story of parameterization by treewidth.

We show that Isometric Path Partition is W[1]-hard when parameterized by treewidth (in fact, even pathwidth (pw)), answering the question by Dumas et al. [SIDMA, 2024], Fernau et al. [TCS, 2025], and confirming the aforementioned tendency. We complement this hardness result by designing a tailored dynamic programming algorithm running in $n^{O(\mathsf{tw})}$ time. This dynamic programming approach also results in an algorithm running in time $\mathsf{diam}^{O(\mathsf{tw}^2)} \cdot n^{O(1)}$, where diam is the diameter of the graph. It is known that Isometric Path Partition remains NP-hard on graphs of diameter 2; hence, the combination of both parameters is necessary to obtain a tractable algorithm. Note that the dependency on treewidth is unusually high, as most problems that are FPT for treewidth admit algorithms running in time $2^{O(\mathsf{tw})} \cdot n^{O(1)}$ or $2^{O(\mathsf{tw} \log(\mathsf{tw}))} \cdot n^{O(1)}$. However, we rule out the possibility of a significantly faster algorithm, showing that Isometric Path Partition does not admit an algorithm running in time $\mathsf{diam}^{o(\mathsf{pw}^2/(\log^3(\mathsf{pw})))} \cdot n^{O(1)}$, assuming the Randomized-ETH.

2012 ACM Subject Classification Theory of computation → Fixed parameter tractability

Keywords and phrases Isometric path partition, parameterized complexity, treewidth, diameter, Randomized ETH

Digital Object Identifier 10.4230/LIPIcs.WG.2026.11

Related Version *Full Version*: `https://arxiv.org/abs/2508.05448`

Funding *Florent Foucaud*: ANR project GRALMECO (ANR-21-CE48-0004), French government IDEX-ISITE initiative 16-IDEX-0001 (CAP 20-25), International Research Center "Innovation Transportation and Production Systems" of the I-SITE CAP 20-25, and the CNRS IRL ReLaX.
Prafullkumar Tale: Supported by ARG-MATRICS Grant titled MaPLE (ARGM/2026/0089).

Acknowledgements The authors would like to thank the organizers and participants of the 2024 Workshop of the AGC team of LIMOS at Saint-Jacques-d'Ambur for the relaxed yet inspiring atmosphere, where this research was initiated.

1 Introduction

In this paper, we investigate the parameterized complexity of a metric-based optimization problem known as Isometric Path Partition, that deals with partitioning the vertex set of an input graph into a given number of isometric (i.e., shortest) paths.

Metric-based optimization problems. The main subject of metric graph theory is the investigation and characterization of graph classes and graph problems, where the graphs are equipped with a metric [4,23]. It is a central topic in mathematics and computer science with far-reaching applications in group theory [1,43], matroid theory [5], learning theory [24–27], and computational biology [6]. One of the most natural metrics related to graphs is the shortest-path distance between two vertices. On the algorithmic side, many problems related to network monitoring, transportation networks, information retrieval, or computational learning can often be formulated as problems on graphs in which the objective is to find vertices that satisfy specified distance-related properties. We use "metric-based optimization problems" as an umbrella term for such problems. This includes many important and classic graph problems, such as Single Source Shortest Paths, Distance d-Dominating Set (also called (k, d)-Center), Distance d-Independent Set (also called d-Scattered Set), Metric Dimension, Geodetic Set, Isometric Path Cover, etc.

Some of these problems have been cornerstones in the development of classic as well as parameterized algorithms and complexity [8,12,20,37,46,47,54], as they behave quite differently from their more "local" (neighborhood-based) counterparts such as Vertex Cover, Independent Set or Dominating Set.

In parameterized analysis, we associate each instance I with a parameter ℓ, and are interested in an algorithm with running time $f(\ell) \cdot |I|^{O(1)}$ for some computable function f. Parameterized problems that admit such an algorithm are called *fixed parameter tractable* (FPT) parameterized by ℓ. On the other hand, W[1]-hardness categorizes problems that are unlikely to have FPT algorithms. A parameterized problem is in XP if it admits an algorithm running in time $|I|^{f(\ell)}$ for some computable function f.

Limitations of Treewidth. Many problems admit FPT algorithms when parameterized by treewidth, a parameter that quantifies tree-likeness of the graph. Courcelle's celebrated theorem [29] states that the class of graph problems expressible in Monadic Second-Order (MSO) Logic of constant size is fixed-parameter tractable (FPT) when parameterized by the treewidth of the graph [30, Chapter 7]. Although one can express many graph properties using MSO formulas of constant size, there is no such formula to encode the following: given a subset of vertices and two specified vertices s, t, this subset forms an isometric path (i.e., a shortest path) between s and t [53]. This hinders the application of Courcelle's theorem to metric-based optimization problems.

Consider the example of Dominating Set and its generalization Distance d-Dominating Set. The objectives of these problems are to find a subset of vertices S such that any vertex in $V(G) \setminus S$ is at distance at most 1 and at most d, respectively, from at least one vertex in S. As the distance requirement in the first problem is upper bounded by a constant, it is expressible as an MSO formula of constant size, resulting in an FPT algorithm parameterized by treewidth. However, this is not the case for the latter problem. In fact, if d is part of the input, it is known that DISTANCE d-DOMINATING SET is W[1]-hard when parameterized by treewidth [17]. There are similar results for INDEPENDENT SET and its generalization DISTANCE d-INDEPENDENT SET [47]. Similarly, the metric-based optimization problems GEODETIC SET and METRIC DIMENSION are even NP-hard when the treewidth of the graph is a constant [54,70]. Hence, metric-based optimization problems require individual attention and often draw the boundary for the success story of parameterization by treewidth stemming from Courcelle's theorem.

Isometric Path Partition. Isometric (i.e., shortest) paths in graphs and vertex-partitioning are among the most fundamental constructs in the area of graph algorithms. In this article, we consider an interesting metric-based optimization problem known as ISOMETRIC PATH PARTITION, whose objective is to partition the vertex set of a graph into a prescribed number of isometric paths. Formally it is defined as follows.

> ISOMETRIC PATH PARTITION
> **Input:** A graph G and an integer k.
> **Output:** Is there a partition of the vertex set of G into k sets, each of them forming an isometric path in G?

Algorithmic aspects of ISOMETRIC PATH PARTITION received increasing attention in recent years [18,34,59]. (We discuss the related literature in detail later.) It is also related to other (non-metric based) path problems such as the celebrated HAMILTONIAN PATH (and its generalization PATH PARTITION) or DISJOINT PATHS, which are fundamental and have numerous applications [3,58,66].

Our results. As our first result, we show that the problem is XP parameterized by treewidth.

▶ **Theorem 1.** *ISOMETRIC PATH PARTITION admits an algorithm running in time $n^{O(\mathsf{tw})}$, where tw is the treewidth of G and n denotes its number of vertices.*

Theorem 1 establishes that ISOMETRIC PATH PARTITION behaves differently from some other metric-based problems, that are NP-hard even for constant treewidth [54,70].

We note that Theorem 1 implies a new (and explicit) XP algorithm for ISOMETRIC PATH PARTITION parameterized by solution size k: indeed, it is known that in a YES-instance, the pathwidth of a graph is upper-bounded by an exponential function of the solution size [33], so our algorithm directly provides an algorithm running in time $n^{2^{O(k)}}$. Different XP algorithms for solution size were given in [33] and [34] (note that these rely on meta-arguments such as variants of Courcelle's theorem or the DISJOINT PATHS problem).

The next natural question is whether the XP algorithm from Theorem 1 can be improved to an FPT algorithm? Recall that closely related "path problems" like HAMILTONIAN PATH and PATH PARTITION are both FPT parameterized by treewidth [31,39]. We show that this is unlikely to be the case for ISOMETRIC PATH PARTITION.

▶ **Theorem 2.** *ISOMETRIC PATH PARTITION is* W[1]-hard *when parameterized by the pathwidth, and hence, the treewidth of the input graph.*

Theorem 2 answers open questions from [33] and [34]. Moreover, Theorem 1 and Theorem 2 establish that ISOMETRIC PATH PARTITION belongs to the list of problems that are XP but W[1]-hard for treewidth. We refer to [9] for a discussion about such problems. It appears that certain common problem features yielding this behavior can be listed, for example, problems involving weights, lists, or iterative processes. Another kind of such feature is the fact of being metric-based, such as Metric Dimension [54], Geodetic Set [49] and Distance-d Dominating/Independent Set [46,47]. Our result confirms this trend and draws an interesting distinction with the related (path-based but not metric-based) Path Partition, which is FPT for treewidth [39].

For metric-based problems, another relevant parameter is the diameter (the maximum length of an isometric path). Unfortunately, Isometric Path Partition is NP-hard even on graphs of diameter 2 [19]: using the diameter alone as the parameter is not fruitful.

As a third result, we show that Isometric Path Partition becomes FPT when parameterized by both treewidth and diameter. (Note that parameterization by both diameter and treewidth has been explored earlier in the context of other metric-based problems [37,45].) To obtain this result, we use a dynamic programming scheme analogous to that of Theorem 1, but manage to reduce the number of states by storing more succinct information about the distances of the vertices to the bags of the decomposition.

▶ **Theorem 3.** *Isometric Path Partition admits an algorithm running in time $\mathsf{diam}^{O(\mathsf{tw}^2)} \cdot n^{O(1)}$ where diam is the diameter of G, tw its treewidth, and n its number of vertices.*

The dependency on treewidth is unusually high, as most natural problems that are FPT for treewidth admit algorithms running in time $2^{O(\mathsf{tw})} \cdot n^{O(1)}$ or $2^{O(\mathsf{tw}\log(\mathsf{tw}))} \cdot n^{O(1)}$. We however show that an improved algorithm achieving these types of running time is highly unlikely assuming the *Randomized-ETH*, which, informally speaking, postulates that there exists no randomized algorithm with expected running time $2^{o(n)}$ to solve n-variable 3-SAT [42].

▶ **Theorem 4.** *Unless the Randomized-ETH fails, Isometric Path Partition does not admit an algorithm running in time $\mathsf{diam}^{o(\mathsf{pw}^2/(\log^3(\mathsf{pw})))} \cdot n^{O(1)}$.*

This type of lower bounds, i.e., forbidding running times roughly of the form $2^{o(p^2)}$ for some parameter p, matched by an algorithm of this running time, are not very common in the literature. We refer here to the only other such results known to us [2,22,24,38,64,67,68] which hold for the parameters pathwidth, vertex cover number, or solution size.

Related work. Isometric Path Partition (under this name or the one of Shortest Path Partition) was introduced as a natural variation of the related Isometric Path Cover, which is motivated by applications in the cops and robber game [35]. Isometric Path Partition was studied from the structural point of view for specific graph families [36,59,63] and shown to be NP-complete in [59]. This holds even for bipartite graphs of diameter 4 [34], chordal graphs of diameter 2 [19] and split graphs [21]. Isometric Path Partition is known to be polynomial-time solvable on trees [15, 40, 41, 52], cographs [21], and chain graphs [21]. It can also be solved in polynomial time for any fixed number of solution paths by XP algorithms, using two different methods: see [33] and [34], respectively. Isometric Path Partition is also shown to be FPT when parameterized by the neighborhood diversity of the input graph, and also when parameterized by the dual parameter $n - k$ [34]. The variant of Isometric Path Partition for DAGs is W[1]-hard for solution size k [34].

The related problem Isometric Path Cover, where the objective is to cover the vertex set of the input graph with (not necessarily disjoint) isometric paths, is also studied [18,19,33] and is relevant in the context of machine learning [71].

Interesting upper bounds on the pathwidth of a graph that can be covered by a given number of isometric paths were recently obtained in [7,33].

Another related problem is Disjoint Shortest Paths, where we are given a set of terminal pairs and we need to find disjoint isometric paths connecting the pairs, is also studied: see [10, 56] and references therein. A global minimization variant called Shortest Disjoint Paths (where only the sum of lengths of the path needs to be minimized) is also studied [13, 60]. These two problems are variants of the celebrated Disjoint Paths problem [48], which is central in the theory of graph minor testing [66].

When the paths are not required to be isometric, we have the general Path Partition problem [28, 41] (also known under the names of Path Cover and Hamiltonian Completion), that generalizes Hamiltonian Path and is extensively studied. This problem is also important from a structural point of view, see [11, 44, 57]. The version where the paths need to be induced (or chordless) is also studied, see [34, 62].

Practical applications of path partition problems are numerous, for example, automatic translation [55], network routing [69], program testing [61] or parallel programming [65]. We refer to the surveys [3, 58] for more references on path partitioning (and covering) problems.

Organization of the paper. Due to space constraints, the proofs of all results are omitted from this extended abstract. Instead, we present a high-level overview intended to convey the central ideas and key insights of our algorithms and reductions. The reader is referred to the full version of this manuscript, for more details.

Preliminary observations are given in Section 2, our dynamic programming schemes are presented in Section 3, and our hardness results are described in Sections 4 and 5. We devote particular attention to Section 4, as it contains what we regard as the principal contribution of this work. Finally, we discuss future research directions in Section 6.

2 Preliminaries

We assume the reader to be familiar with the standard terminology from graph theory and parameterized complexity, and refer to [30, 32] and the long version of this work for more details on these notions. This section is devoted to stating easy properties that play a crucial role in simplifying our reductions in Sections 4 and 5.

Let us recall that the *length* of a path denotes its number of edges, not vertices. An *isometric path* (or *IP* for short) is a shortest path between its endpoints. An *IP-partition* of a graph G is a partition of the vertex set into isometric paths. The *size* of an IP-partition $\mathcal{P}$ of a graph G is the cardinality of $\mathcal{P}$.

The next property shows that any leaf may be assumed to be part of a path of length at least 1 in any optimal IP-partition.

▶ **Lemma 5** (Leaf lemma). *Let G be a graph, and D denote the set of vertices of G of degree 1. Let $v \in V(G)$ such that $N(v) \cap D = \{u\}$. Then, G has an IP-partition with minimum cardinality containing a path with u as an endpoint and of length at least 1.*

In the remaining of the article, we denote by *cherry* an induced path on three vertices with endpoints of degree 1. The next property shows that cherries may be assumed to be part of any optimal IP-partition.

▶ **Lemma 6** (Cherry lemma). *Let G be a graph, and D denote the set of vertices of G of degree 1. Let $v \in V(G)$ be a vertex such that $N(v) \cap D = \{u_1, u_2\}$. Then G has an IP-partition with minimum cardinality containing the path (u_1, v, u_2).*

We derive the following as a corollary of the Cherry lemma.

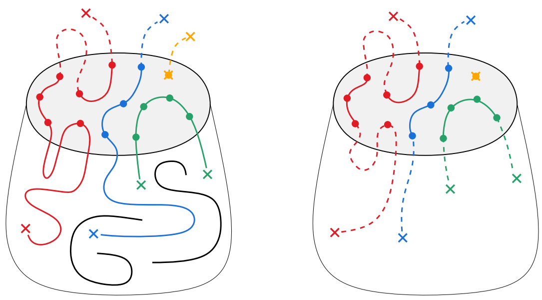

Figure 1 Illustration of a solution intersected with the graph induced by the union of bags from a subtree of the tree-decomposition (left) and its signature (right). Colors indicate which pieces originate from a same solution path. Dotted lines represent the connections outside of the bag, and crosses represent the locations of solution path endpoints.

▶ **Lemma 7** (Twin-cherries lemma). *Suppose that G contains a pair of cherries with an isometric path connecting their middle vertex, and that every other vertex in G is only connected to this subgraph via the middle vertices of the cherries. Then G has an IP-partition with minimum cardinality containing the two cherries as well as the internal part of the path connecting their middle vertex.*

Let us highlight a consequence of adding cherries to a graph, which will provide insight for our hardness reductions in Sections 4 and 5. As stated in Lemma 6, any minimum IP-partition $\mathcal{P}$ of a graph G can be assumed to contain any cherry. Thus, cherries can be added to a graph while assuming that no path in a minimum IP-partition other than cherries use these newly added vertices. Consequently, by adding a cherry to the graph, and making its middle vertex adjacent to a set A of vertices, we are able to reduce the distance between elements of A to at most 2, without changing the "structure" of the IP-partition, in the sense that cherries can be ignored from the set of vertices that are reachable by other isometric paths. Moreover, this metric reduction can be adapted to arbitrary distances by using twin-cherries as described in Lemma 7.

3 Dynamic programming schemes

In this section, we briefly describe a dynamic programming algorithm solving Isometric Path Partition in XP time parameterized by the width of a given tree-decomposition. The details can be found in the long version of this work. This algorithm roughly consists of storing, for each bag of a decomposition, the possible intersections of isometric paths partitions with the bags, an indication of which intersected piece comes from which path (via a coloring function), an indication of how the obtained pieces are connected outside the bags (via pairing functions), together with the location of their endpoints, which are used (via their distance to the bag) to ensure that these pieces of paths are coherent with those of an isometric path. We call signature of a solution the concatenation of each such information; see Figure 1 for an illustration. The states of the dynamic programming algorithm consist of the signatures, and their value is defined as the minimum order of a solution which is compatible with the signature. Although this approach appears to follow standard techniques, it requires careful attention, as we need to ensure that solution paths are isometric.

The algorithm computes (and stores) the values of each possible state in a bottom-up fashion, starting with the leaf nodes of the decomposition, and finishing at its root node, where the value of optimal solution is computed. The fact that each state can be correctly computed follows by induction on the type of nodes in the decomposition. Moreover, we prove that the number of distinct signatures is bounded above by $n^{O(w)}$, where n is the number of vertices, w is the width of the decomposition, and the value of one state can be computed within this time. Finally, since a tree-decomposition of width $w \leq 2\mathsf{tw}$ can be computed in time $2^{O(\mathsf{tw})} \cdot n$ [51], and can be transformed into a nice tree-decomposition with $O(n)$ nodes in $O(wn)$ time [14, 50], we derive Theorem 1.

The dependence on n in the number of states of our algorithm comes from the fact that we store the endpoints of paths (to verify the distances and ensure that pieces of paths are coherent) and that these endpoints may lie anywhere in G. In the long version of this work, we argue that we can improve the time bound of the algorithm to FPT by performing these verifications without storing arbitrary endpoints, and derive Theorem 3.

4 Hardness with Respect to Pathwidth

In this section, we highlight a reduction showing that the Isometric Path Partition problem is W[1]-hard when parameterized by the pathwidth (and hence treewidth) of the input graph, thereby proving Theorem 2. The reduction is from Multicolored Clique, which is W[1]-hard when parameterized by the solution size; see, e.g., [30, Chapter 13].

Multicolored Clique
Input: A graph G, an integer k, and a partition $(V_1, V_2, \dots, V_k)$ of $V(G)$ such that V_i is an independent set and $|V_i| = n$, i.e., $V_i = \{v^i_1, \dots v^i_n\}$, for every $i \in [k]$.
Question: Does there exist a clique in G containing one vertex from V_i for every $i \in [k]$?

Note that G has kn vertices. In an instance of Multicolored Clique, the sets $V_1, \dots, V_k$ are called *color classes*, and the goal can be rephrased as deciding whether there exists a multicolored clique in G.

4.1 Overview of the Reduction

The reduction takes as input an instance $(G, k, (V_1, V_2, \dots, V_k))$ of Multicolored Clique and returns an instance $(H, \text{poly}(n, k))$ of Isometric Path Partition in polynomial time where H is of polynomial size in k and n. The graph constructed has pathwidth $O(k^2)$. For a better comprehension of the coming reduction, it is convenient here to interpret the Multicolored Clique problem as selecting $\binom{k}{2}$ edges in a way that this set is incident to exactly one vertex of each set V_i.

The structure of H is organized as follows. For each $1 \leq i \leq k$, there is a *semi-grid* Γ_i that corresponds to the set V_i. A semi-grid has a grid-like structure with $O(k)$ rows and $O(n)$ columns. See Figure 2 for an illustration. The columns of Γ_i correspond to the vertices in V_i. Semi-grids are connected together via their left and right boundaries by gadgets encoding edges, as shown in Figure 3: for each edge (u, v) with $u \in V_i$ and $v \in V_j$, there is an edge gadget that connects Γ_i and Γ_j on appropriate rows, whose indices depend on i and j.

To facilitate the analysis of the reduction, a number of cherries are part of the construction. Recall that by Lemma 6, we can always assume an IP-partition of minimum cardinality to contain such cherries. Once we made this assumption, there are only two relevant ways of partitioning an edge gadget encoding $(u, v) \in E(G)$:

- Either we decide to partition the edge gadget optimally, i.e., by selecting four isometric paths, as shown in Figure 5 (right). Choosing this red partition corresponds to *not* selecting (u, v) in the Multicolored Clique solution. In this case, no other vertices can be covered by such paths: we say that they are *non-extendable*.

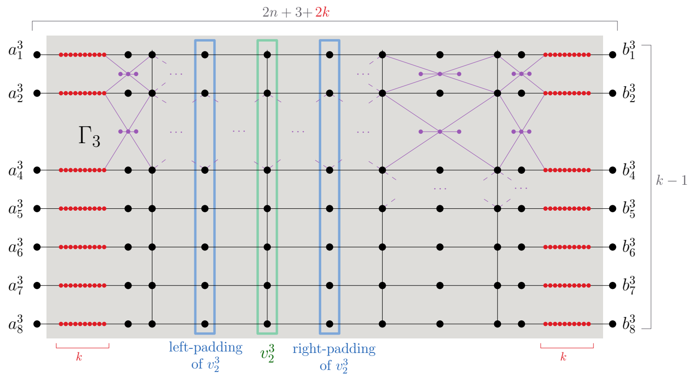

Figure 2 The final stage of the semi-grid used to represent the color class V_i in the reduction, with $i = 3$ and $k = 8$ here. Each vertical path corresponds to a vertex in the set V_i. Each row will be used to connect V_i to another color class V_j, $j \neq i$, with edge gadgets encoding all the adjacencies between V_i and V_j; this is why no row is indexed 3 in Γ_3, and that the number of rows is $k - 1$. Grid cherries are depicted in purple.

- Or, we decide to partition the gadget with five isometric paths, as shown in Figure 5 (left). Choosing this green partition corresponds to selecting the edge (u, v) in the Multicolored Clique solution. Although this is not an optimal way to cover the gadget, choosing the green partition has a strong benefit: paths emerging from the gadget can penetrate inside semi-grids Γ_i and Γ_j to cover some vertices there, which potentially reduces the number of paths needed to cover the semi-grids. Penetrating inside semi-grids in that way, however, can only be done up to some column that is left uncovered, and that is referred to as the *crest* in Figure 3. In the ideal scenario where the instance of the Multicolored Clique problem is positive, almost all vertices of the semi-grids are covered by the green paths emerging from a selection of edges, except for the crest columns. For each semi-grid, this column corresponds to the vertex picked in the Multicolored Clique solution, and can be covered using only one isometric path.

4.2 Reduction

In this section, we give the description of the reduction, that we break by subsections for each gadget, and for the value of the solution size.

4.2.1 Semi-grids for Color Classes

Recall that k is the size of the partition of $V(G)$ and n is the number of vertices in each part. For every $i \in [k]$, we add a structure that we call *semi-grid*, denoted by Γ_i, aimed at representing the color class V_i, and that we shall define now. For convenience, we choose to present it starting with a $(k-1) \times (2n+3)$ grid, and later specify edges to be deleted, and subdivisions to be made. The following steps are better understood accompanied with Figure 2 which depicts the resulting construction.

- We start with a $(k-1) \times (2n+3)$ grid Γ_i (represented by fat black vertices in Figure 2) where the $k-1$ rows are purposely labeled with integers in $\{1, \dots, i-1\}$ and $\{i+1, \dots, k\}$, an indexing which will be convenient in the following. In other words, Γ_i should be thought of as the $k \times (2n+3)$ grid with rows labeled from 1 to k, and on which the row i has been removed, and adjacent rows made connected, while keeping the initial indexing.

The indexing of the columns is $0, 1, \dots, 2n+2$. Only the vertices in the leftmost and rightmost columns of the semi-grid will be adjacent to external vertices. They are called the *left* and *right borders* of Γ^i, and are denoted by a^i_j's and b^i_j's, respectively, with j being the index of the row.

- Each column of even index corresponds to a vertex in V_i. Formally, consider a labeling $\{v^i_1, \dots, v^i_n\}$ of the vertices of V_i. Then the $(2p)^{\text{th}}$ column corresponds to the vertex v^i_p for each $p \in [n]$. We refer to the $(2p-1)^{\text{th}}$ column and the $(2p+1)^{\text{th}}$ column as the *left-padding* and *right-padding* of v^i_p, respectively. Note that the left-padding of v^i_1 is the second column, which is distinct from the left border, and analogously for the right-padding of v^i_n which is distinct from the right border. The right-padding and left-padding of consecutive vertices coincide.
- We delete all the vertical edges of the grid both whose endpoints are *not* in the $(2p)^{\text{th}}$ column for some $p \in [n]$. In other words, the only vertical paths we keep are those in the columns corresponding to vertices of V_i.
- We subdivide the edges incident to the border k times. Formally, the reduction replaces each such edge with a path with k internal vertices. We consider all the new columns obtained in the semi-grid as "virtual columns" and consider that this change does not affect the indexing of the columns, for convenience. This allows us to denote the column corresponding to vertex v^i_p as the $(2p)^{\text{th}}$ column instead of the less intuitive notation of "$(2k+2p)^{\text{th}}$ column". The role of such vertices is to ensure that vertical paths of the grid will stay isometric even after the vertices of the left and right border are made adjacent to the vertices of edge gadgets.
- Finally, for each "cell" defined by two consecutive rows and two consecutive columns of the obtained grid-like structure, we add a cherry and connect its middle vertex to the four vertices lying at the intersection of these rows and columns. We do the same at the left of the first column (of index $2p$) by considering as a cell the two vertices of the column plus the last two subdivided vertices. We proceed analogously at the right of the last column. The goal of these cherries is to force isometric paths to be either horizontal or (almost) vertical in the obtained semi-grid. We call these cherries *grid cherries*; note that there are $(n+1) \cdot (k-2)$ such cherries.

4.2.2 Encoding Edges

Consider an edge (v^i_p, v^j_q) of G where $i \neq j \in [k]$ and $p, q \in [n]$. We start describing the part of the edge gadget that will later be attached to each of Γ_i and Γ_j and that we call *left and right cables*. In the following, let us define $N := 2n^2$. The following steps are better understood accompanied with Figure 3.

- We start by creating a simple path $(z_0, x_0, x_1, \dots, x_\ell)$ that we call *left cable* of (v^i_p, v^j_q) with respect to Γ_i, where $\ell = N - 2p - k$. Note that this value corresponds to N minus the distance of the column of v^i_p from the left border of Γ_i. We say that ℓ is the *length*, z_0 is the *core*, and x_ℓ is the *open end* of the cable.
- We create a second simple path $(z'_0, x'_0, x'_1, \dots, x'_{\ell'})$ that we call *right cable* of (v^i_p, v^j_q) with respect to Γ_i, this time with value $\ell' = N - 2(n-p+1) - k$. Note that the value ℓ' corresponds to N minus the distance of the column of v^i_p from the right border of Γ_i.
- We create the left cable $(z'''_0, x'''_0, x'''_1, \dots, x'''_{\ell'''})$ and the right cable $(z''_0, x''_0, x''_1, \dots, x''_{\ell''})$ of (v^i_p, v^j_q) with respect to Γ_j analogously, with i replaced by j, and p replaced by q.

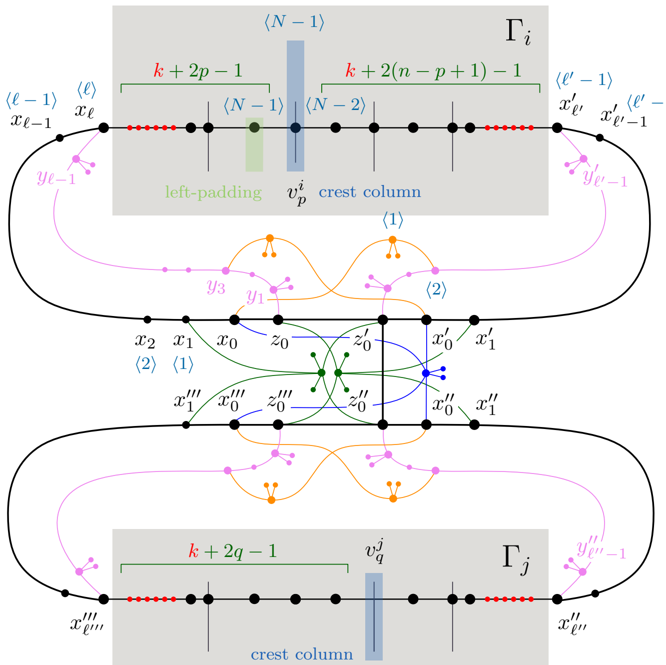

Figure 3 The edge gadget encoding an edge between two color classes. The start cherries are depicted in blue, the core cherries in green, the distance twin-cherries in pink, and the crossing cherries in orange. All these cherries can be assumed to be part of any minimum IP-partition by Lemmas 6 and 7, and hence may intuitively be omitted when analyzing the shape of other isometric paths. The distances from x_0 are depicted between brackets in blue; note that x_0 is equidistant to its crest column and its left-padding.

Note that the left and right cables associated with Γ_i may only differ by their length, which is determined by the distance of v^i_p from the left and right border of Γ_i. This distance is set so that x_0 lies at distance $N-1$ from the left padding of v^i_p, and analogously for x'_0 which lies at distance $N-1$ from the right padding of v^i_p. This p^{th} column is called the *crest* of Γ_i with respect to (v^i_p, v^j_q), and analogously for the crest of Γ_j with respect to (v^i_p, v^j_q).

We connect these four cables by adding the edges $z_0z'_0$, $z'_0z''_0$, and $z''_0z'''_0$ that we call *core edges*. The obtained path $(z_0, z'_0, z''_0, z'''_0)$ will be called the *core path* in the following.

To these cable and core edges, we add four types of cherries and twin-cherries, defined as follows. Note that these extra vertices will be assumed to be part of any optimal IP-partition thanks to Lemmas 6 and 7, hence, they should be considered as gadgets whose role is to modify the distances in the graph, without changing the "structure" of the remaining isometric paths.

- First we add a *start cherry*, in blue in Figure 3, whose middle vertex is adjacent to x_0, x'_0, x''_0, and x'''_0. The role of this cherry will be to simplify the analysis by ensuring these adjacent vertices to lie in distinct isometric paths of a minimum IP-partition.
- Then we add *core cherries*, in green in Figure 3, as follows. The *left* one has its middle vertex adjacent to the core vertices z'_0, z''_0 as well as to x_1 and x'''_1. The *right* one has its middle vertex adjacent to the core vertices z_0, z'''_0 as well as to x'_1 and x''_1. The role of these cherries will be to force that cables cannot be isometrically extended using the core of another cable.

- Then, we add a twin-cherry as follows: we create a path $y_1, \ldots, y_{\ell-1}$ with y_1 adjacent to z_0, $y_{\ell-1}$ adjacent to x_ℓ, and add two leaves to each of y_1 and $y_{\ell-1}$. We call this twin-cherry the *core twin-cherry* of the left path of (v_p^i, v_q^j) with respect to Γ_i. This is depicted in pink in Figure 3. The role of this twin-cherry is to make z_0 equidistant to $x_{\ell-1}, x_\ell$. We define the twin-cherry of the right path of (v_p^i, v_q^j) with respect to Γ_i analogously, and do the same for the left and right paths of (v_p^i, v_q^j) with respect to Γ_j.
- Finally, we add a *crossing cherry*, in orange in Figure 3, whose middle vertex is adjacent to y_3 and y_3', and call it the crossing cherry of (v_p^i, v_q^j) with respect to Γ_i. We define the crossing cherry associated to (v_p^i, v_q^j) with respect to Γ_j analogously. The role of these cherries is to make x_0 at distance $N-1$ from its associated crest column, so that it becomes equidistant to both this column and its left-padding.

This concludes the construction of an edge gadget for edge (v_p^i, v_q^j). We connect it to the appropriate semi-grids by identifying open ends of the paths with the vertices in the borders of the semi-grids Γ_i or Γ_j, depending on whether they are left paths, or right paths. More formally, we identify x_ℓ with vertex a_j^i, and $x'_{\ell'}$ with vertex b_j^i. Then we do the same for Γ_j by identifying $x'''_{\ell'''}$ with vertex a_i^j, and $x''_{\ell''}$ with vertex b_i^j; see Figure 3.

Let us stress the fact that one such edge gadget is added for each single edge (v_p^i, v_q^j) between V_i and V_j, and that all such gadgets are thus attached to the same j^{th} row of Γ_i, and to the same i^{th} row of Γ_j, via their open ends. In particular, open ends separate each edge gadget from the other gadgets and the associated semi-grid. In the following, we call *inner vertices* of an edge gadget the vertices of the edge gadget that are distinct from its open ends.

4.2.3 Valve Cherries

For each semi-grid Γ_i, we add two *valve cherries*: one on the left of the semi-grid, and one on the right. See Figure 4 for an illustration. The middle vertex of the left cherry is made adjacent to $x_{\ell-2}$ and $x_{\ell-1}$ in every edge gadget attached to the left border of the semi-grid. The middle vertex of the right cherry is made adjacent to $x'_{\ell-2}$ and $x'_{\ell-1}$ in every edge gadget attached to the right border. We argue in the long version of this work that the valve cherries prevent isometric paths to intersect distinct edge gadgets under additional assumptions.

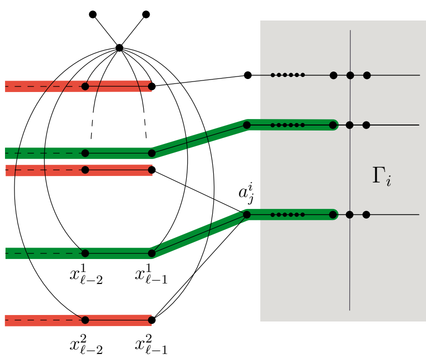

Figure 4 Valve cherries attached to the edge gadgets attached to a border of a semi-grid, here the left border. With the notation $x^1_{\ell-2}$ we mean the vertex $x_{\ell-2}$ in the edge gadget of e_1.

4.2.4 Solution Size

The reduction sets $k' = k \cdot (n+1) \cdot (k-2) + 23 \cdot |E(G)| + \binom{k}{2} + k + 2k$ and returns (H, k') as an instance of Isometric Path Partition.

We present an informal justification for the above value of k'. The first additive term corresponds to the number of grid cherries that are found in the semi-grids. The second term corresponds to the number of paths used to partition the vertices in edge gadgets, 19 of which will consist of cherries and twin-cherries. The third term corresponds to the number of selected edges: here, the partition needs to spend an extra isometric path for each selection. A path from the selected edges can be extended to partition almost all the vertices in the semi-grid except for their crest. If selected edges correspond to a multicolored clique in G, then the solution can partition the remaining crest vertices in semi-grids using k vertical isometric paths, which corresponds to the fourth additive term. The last additive term corresponds to the valve cherries.

Relying on properties that are deferred to the full version of this paper, we obtain that (H, k') is a Yes-instance of Isometric Path Partition if and only if (G, k) is a Yes-instance of Multicolored Clique. Moreover, the graph H has pathwidth $O(k^2)$. We conclude to Theorem 2, noting that the graph (H, k') can be computed in polynomial time given an instance of Multicolored Clique.

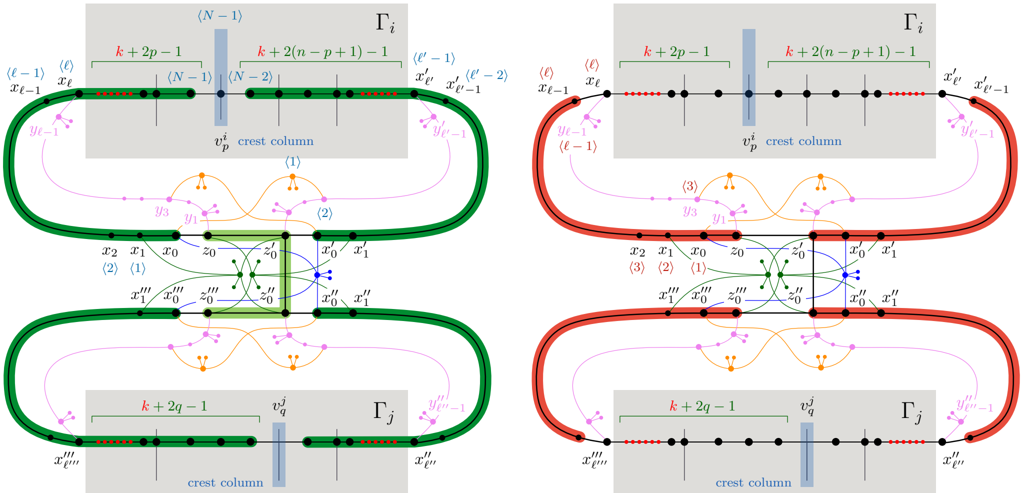

Figure 5 On the left, in green, four extendable paths and one core path partitioning all the vertices of an edge gadget, but those in cherries and twin-cherries. On the right, in red, four non-extendable paths partitioning the vertices that are neither open ends, nor part of a cherry or of a twin-cherry. (The distances from x_0 are depicted between brackets in blue, and those from z_0 are depicted between brackets in red.)

5 Lower Bound w.r.t. Pathwidth and Diameter

In this section, we briefly describe a reduction showing that Isometric Path Partition does not admit an algorithm running in time $O(\mathsf{diam}^{o(\mathsf{pw}^2/\log^3(\mathsf{pw}))})$, unless the Randomized ETH fails. This reduction is from a variant of Sparse 3-SAT that has been introduced by Gourvès et al. [42] and that we state here.

Sparse 3-SAT (variant)
Input: An integer n which is a perfect square, and a 3-SAT formula with at most n variables and at most n clauses such that each variable appears in at most 3 clauses. Moreover, a partition $\{V_1, \ldots, V_{\sqrt{n}}\}$ of the set of variables V and a partition $\{C_1, \ldots, C_{\sqrt{n}}\}$ of the set of clauses C such that each part is of size at most $\sqrt{n}$ and for every $i, j \in [\sqrt{n}]$ the cardinality of the set $\{(x, c) : x \in V_i,\, c \in C_j,\, x \in c\}$ is at most 1.
Question: Does there exist a satisfying assignment of the formula?

In [42] is shown that unless the Randomized ETH fails, Sparse 3-SAT (without the last constraint on the cardinality) does not admit an algorithm running in time $2^{o(n)}$. (The fact that hardness still holds for our variant is argued in the long version of this work.)

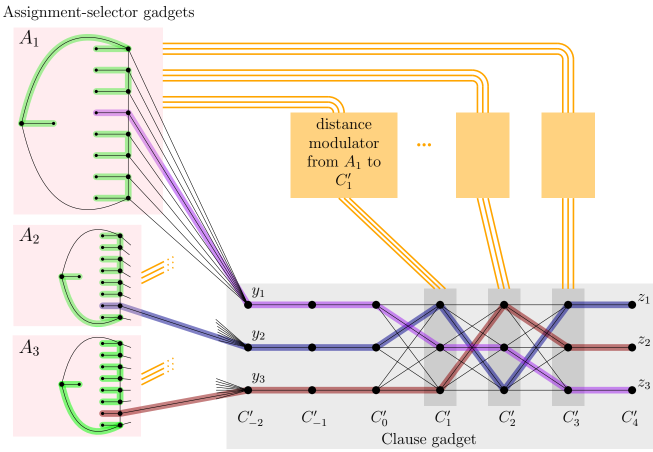

Figure 6 A schematic diagram of the reduction where $\sqrt{n} = 3$. The assignment-selection paths are shown in purple, blue and red. and their left endpoint corresponds to the assignment chosen. For each of them, the vertices covered on $C'_1, \ldots, C'_{\sqrt{n}}$ are the clauses satisfied by this assignment. Distance modulators prevent an assignment-selection path to cover a clause that is not satisfied by the corresponding assignment.

The reduction takes as input an instance $(\phi, \{V_1, \ldots, V_{\sqrt{n}}\}, \{C_1, \ldots, C_{\sqrt{n}}\})$ of Sparse 3-SAT, runs in time $2^{O(\sqrt{n})}$, and returns an instance (G_ϕ, k) of Isometric Path Partition where $k = 2^{O(\sqrt{n})}$ as depicted in Figure 6. The pathwidth and diameter of G_ϕ are $O(\sqrt{n} \log n)$ and $O(\sqrt{n})$, respectively. Altogether, these bounds imply that Isometric Path Partition does not admit an $O(\mathsf{diam}^{o(\mathsf{pw}^2/\log^3(\mathsf{pw}))})$-time algorithm, unless the Randomized ETH fails.

The graph of the reduction consists of three parts: the *assignment gadget* that encodes the possible assignments of the variables, a *clause gadget* that contains a vertex for each clause, and *distance modulator* gadgets that encode the formula.

For each variable group V_i, the *assignment-selector gadget* A_i consists of vertices corresponding to the $2^{\sqrt{n}}$ possible (partial) assignments of the variables in this group. We refer to the long version of this work for details on this assignment-selector gadget. It is designed in a way that only one path can leave the gadget. Altogether, these $\sqrt{n}$ paths, called *assignment-selection paths*, encode an assignment of the variables.

The clause gadget consists of a chain of $\sqrt{n}$ sets $C'_1, \ldots, C'_{\sqrt{n}}$ each representing a clause set, together with four additional sets C'_{-2}, C'_{-1}, C'_0 and $C'_{\sqrt{n}+1}$ that will force some properties on isometric paths. Each of these sets consists of $\sqrt{n}$ vertices, and each clause of the formula is associated to one of these vertices. The assignment gadget is connected to the clause gadget in a way that the assignment-selection paths need to traverse all the sets $C'_1, \ldots, C'_{\sqrt{n}}$, ideally, covering all clause vertices on the way. To force an assignment-selector path to only cover clauses that are satisfied by the corresponding assignment, we rely on the fact that these paths must be isometric. To do this, we add *distance modulator* gadgets that shorten by 1 the distance between assignments vertices and clauses not satisfied by this assignment.

To ensure that distance modulators only act as a metric-changer and not as an alternative way for the assignement-selector paths, we use a number of cherries as in Section 2.

6 Conclusion

We studied the parameterized complexity of Isometric Path Partition, proving that it admits an XP-algorithm but is W[1]-hard when parameterized by treewidth (and pathwidth). This complements results from [33] and [34], and answers open questions mentioned therein. In addition, we obtained an FPT algorithm (with running time $\mathsf{diam}^{O(\mathsf{tw}^2)} \cdot n^{O(1)}$) parameterized by diameter and treewidth, and proved a conditional randomized randomized-ETH-based lower bound that differs from the algorithm running time only by a poly-logarithmic factor in the exponent. As noted before, this type of running time is relatively rare in the literature, even among metric-based problems. Our results shows that Isometric Path Partition behaves more like other metric-based problems with respect to treewidth and diameter, as opposed to its non-metric counterpart (Path Partition, see [39]). Nevertheless, its precise behavior is unique among such metric-based problems, rendering its study interesting.

As a future research question, we wonder whether the problem is FPT or W[1]-hard when parameterized by solution size k. Recall that there is an algorithm running in time $f(k) \cdot n^{O(k)}$ [33], and that the treewidth is upper-bounded by a function of k in any Yes-instance [33]. Note that the analogous problem is known to be W[1]-hard for k on DAGs [34], but the authors reported being unable to prove the analogous result for undirected graphs.

Another interesting question is whether Isometric Path Partition becomes FPT for treewidth on planar graphs. This question has been raised for other metric-based problems such as Metric Dimension [16], and remains open there as well.

References

1 Ian Agol. The virtual Haken conjecture (with an appendix by Ian Agol, Daniel Groves and Jason Manning). *Documenta Mathematica*, 18:1045–1087, 2013.

2 Akanksha Agrawal, Daniel Lokshtanov, Saket Saurabh, and Meirav Zehavi. Split contraction: The untold story. *ACM Transactions on Computation Theory (TOCT)*, 11(3):1–22, 2019. doi:10.1145/3319909.

3 Giovanni Andreatta and Francesco Mason. Path covering problems and testing of printed circuits. *Discrete applied mathematics*, 62(1-3):5–13, 1995. doi:10.1016/0166-218X(94)00142-Z.

4 Hans-Jurgen Bandelt and Victor Chepoi. Metric graph theory and geometry: a survey. *Contemporary Mathematics*, 453(49-86):1–1, 2008.

5 Hans-Jürgen Bandelt, Victor Chepoi, and Kolja Knauer. Coms: complexes of oriented matroids. *Journal of Combinatorial Theory, Series A*, 156:195–237, 2018. doi:10.1016/J.JCTA.2018.01.002.

6 Hans-Jürgen Bandelt and Andreas WM Dress. Split decomposition: a new and useful approach to phylogenetic analysis of distance data. *Molecular Phylogenetics and Evolution*, 1(3):242–252, 1992.

7 Julien Baste, Lucas de Meyer, Ugo Giocanti, Étienne Objois, and Timothé Picavet. A polynomial bound on the pathwidth of graphs edge-coverable by k shortest paths. In *43rd International Symposium on Theoretical Aspects of Computer Science, STACS 2026*, LIPIcs, pages 10:1–10:17, 2026. doi:10.4230/LIPIcs.STACS.2026.10.

8 Rémy Belmonte, Fedor V Fomin, Petr A Golovach, and MS Ramanujan. Metric dimension of bounded tree-length graphs. *SIAM Journal on Discrete Mathematics*, 31(2):1217–1243, 2017. `doi:10.1137/16M1057383`.

9 Rémy Belmonte, Eun Jung Kim, Michael Lampis, Valia Mitsou, and Yota Otachi. Grundy distinguishes treewidth from pathwidth. *SIAM Journal on Discrete Mathematics*, 36(3):1761–1787, 2022. `doi:10.1137/20M1385779`.

10 Matthias Bentert, André Nichterlein, Malte Renken, and Philipp Zschoche. Using a geometric lens to find k-disjoint shortest paths. *SIAM Journal on Discrete Mathematics*, 37(3):1674–1703, 2023.

11 Claude Berge. Path partitions in directed graphs. In *North-Holland Mathematics Studies*, volume 75, pages 59–63. Elsevier, 1983.

12 Benjamin Bergougnoux, Oscar Defrain, and Fionn Mc Inerney. Enumerating minimal solution sets for metric graph problems. *Algorithmica*, pages 1–24, 2025.

13 Andreas Björklund and Thore Husfeldt. Shortest two disjoint paths in polynomial time. *SIAM Journal on Computing*, 48(6):1698–1710, 2019. `doi:10.1137/18M1223034`.

14 Hans L. Bodlaender, Paul Bonsma, and Daniel Lokshtanov. The fine details of fast dynamic programming over tree decompositions. In *International Symposium on Parameterized and Exact Computation*, pages 41–53. Springer, 2013. `doi:10.1007/978-3-319-03898-8_5`.

15 F.T. Boesch, S. Chen, and J.A.M. McHugh. On covering the points of a graph with point disjoint paths. In *Graphs and Combinatorics*, pages 201–212. Springer, 1974.

16 Édouard Bonnet and Nidhi Purohit. Metric dimension parameterized by treewidth. *Algorithmica*, 83(8):2606–2633, 2021. `doi:10.1007/S00453-021-00808-9`.

17 Glencora Borradaile and Hung Le. Optimal dynamic program for r-domination problems over tree decompositions. In *11th International Symposium on Parameterized and Exact Computation, IPEC 2016, August 24-26, 2016, Aarhus, Denmark*, volume 63 of *LIPIcs*, pages 8:1–8:23. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2016. `doi:10.4230/LIPIcs.IPEC.2016.8`.

18 Dibyayan Chakraborty, Jérémie Chalopin, Florent Foucaud, and Yann Vaxès. Isometric path complexity of graphs. In *48th International Symposium on Mathematical Foundations of Computer Science, MFCS 2023*, volume 272 of *LIPIcs*, pages 32:1–32:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. `doi:10.4230/LIPIcs.MFCS.2023.32`.

19 Dibyayan Chakraborty, Antoine Dailly, Sandip Das, Florent Foucaud, Harmender Gahlawat, and Subir Kumar Ghosh. Complexity and algorithms for isometric path cover on chordal graphs and beyond. In *Proceedings of the 33rd International Symposium on Algorithms and Computation, ISAAC*, volume 248 of *LIPIcs*, pages 12:1–12:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. `doi:10.4230/LIPIcs.ISAAC.2022.12`.

20 Dibyayan Chakraborty, Florent Foucaud, and Anni Hakanen. Distance-based (and path-based) covering problems for graphs of given cyclomatic number. *Discrete Mathematics*, 348(11):114595, 2025. `doi:10.1016/J.DISC.2025.114595`.

21 Dibyayan Chakraborty, Haiko Müller, Sebastian Ordyniak, Fahad Panolan, and Mateusz Rychlicki. Covering and partitioning of split, chain and cographs with isometric paths. In *MFCS 2024*, pages 39–1, 2024.

22 Dipayan Chakraborty, Florent Foucaud, Diptapriyo Majumdar, and Prafullkumar Tale. Tight (double) exponential bounds for identification problems: Locating-dominating set and test cover. In *35th International Symposium on Algorithms and Computation, ISAAC 2024*, LIPIcs. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. `doi:10.4230/LIPIcs.ISAAC.2024.19`.

23 Jérémie Chalopin, Manoj Changat, Victor Chepoi, and Jeny Jacob. First-order logic axiomatization of metric graph theory. *Theoretical Computer Science*, 993:114460, 2024. `doi:10.1016/J.TCS.2024.114460`.

24 Jérémie Chalopin, Victor Chepoi, Fionn Mc Inerney, and Sébastien Ratel. Non-clashing teaching maps for balls in graphs. In *The Thirty Seventh Annual Conference on Learning Theory*, volume 247 of *Proceedings of Machine Learning Research*, pages 840–875. PMLR, 2024. URL: `https://proceedings.mlr.press/v247/chalopin24a.html`.

25 Jérémie Chalopin, Victor Chepoi, Fionn Mc Inerney, Sébastien Ratel, and Yann Vaxès. Sample compression schemes for balls in graphs. *SIAM Journal on Discrete Mathematics*, 37(4):2585–2616, 2023. doi:10.1137/22M1527817.

26 Jérémie Chalopin, Victor Chepoi, Shay Moran, and Manfred K. Warmuth. Unlabeled sample compression schemes and corner peelings for ample and maximum classes. *Journal of Computing and System Sciences*, 127:1–28, 2022. doi:10.1016/J.JCSS.2022.01.003.

27 Victor Chepoi, Kolja Knauer, and Manon Philibert. Ample completions of oriented matroids and complexes of uniform oriented matroids. *SIAM Journal on Discrete Mathematics*, 36(1): 509–535, 2022. doi:10.1137/20M1355434.

28 Derek G Corneil, Barnaby Dalton, and Michel Habib. LDFS-based certifying algorithm for the minimum path cover problem on cocomparability graphs. *SIAM Journal on Discrete Mathematics*, 42(3):792–807, 2013. doi:10.1137/11083856X.

29 Bruno Courcelle. The monadic second-order logic of graphs. I. recognizable sets of finite graphs. *Information and Computation*, 85(1):12–75, 1990. doi:10.1016/0890-5401(90)90043-H.

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

31 Marek Cygan, Jesper Nederlof, Marcin Pilipczuk, Michał Pilipczuk, Johan MM Van Rooij, and Jakub Onufry Wojtaszczyk. Solving connectivity problems parameterized by treewidth in single exponential time. *ACM Transactions on Algorithms*, 18(2):17:1–17:31, 2022. doi:10.1145/3506707.

32 R. Diestel. *Graph Theory, 4th Edition*, volume 173 of *Graduate texts in mathematics*. Springer, 2012.

33 Maël Dumas, Florent Foucaud, Anthony Perez, and Ioan Todinca. On graphs coverable by k shortest paths. *SIAM Journal on Discrete Mathematics*, 38(2):1840–1862, 2024. doi:10.1137/23M1564511.

34 Henning Fernau, Florent Foucaud, Kevin Mann, Utkarsh Padariya, and Rajath Rao KN. Parameterizing path partitions. *Theoretical Computer Science*, 1028:115029, 2025. doi:10.1016/J.TCS.2024.115029.

35 D. C. Fisher and S. L. Fitzpatrick. The isometric number of a graph. *Journal of Combinatorial Mathematics and Combinatorial Computing*, 38(1):97–110, 2001.

36 Shannon L. Fitzpatrick, Richard J. Nowakowski, Derek A. Holton, and Ian Caines. Covering hypercubes by isometric paths. *Discret. Math.*, 240(1-3):253–260, 2001. doi:10.1016/S0012-365X(01)00197-2.

37 Florent Foucaud, Esther Galby, Liana Khazaliya, Shaohua Li, Fionn Mc Inerney, Roohani Sharma, and Prafullkumar Tale. Problems in NP can admit double-exponential lower bounds when parameterized by treewidth or vertex cover. In *51st International Colloquium on Automata, Languages, and Programming, ICALP 2024*, volume 297 of *LIPIcs*, pages 66:1–66:19, 2024. doi:10.4230/LIPIcs.ICALP.2024.66.

38 Florent Foucaud, Esther Galby, Liana Khazaliya, Shaohua Li, Fionn Mc Inerney, Roohani Sharma, and Prafullkumar Tale. Metric dimension and geodetic set parameterized by vertex cover. In *42nd International Symposium on Theoretical Aspects of Computer Science, STACS*, volume 327 of *LIPIcs*, pages 33:1–33:20, 2025. doi:10.4230/LIPIcs.STACS.2025.33.

39 Florent Foucaud, Atrayee Majumder, Tobias Mömke, and Aida Roshany-Tabrizi. Polynomial-time algorithms for path cover on trees and graphs of bounded treewidth. In *Proceedings of the 11th International Conference on Algorithms and Discrete Applied Mathematics (CALDAM)*, pages 147–159, 2025. doi:10.1007/978-3-031-83438-7_13.

40 D. S. Franzblau and A. Raychaudhuri. Optimal Hamiltonian completions and path covers for trees, and a reduction to maximum flow. *The ANZIAM Journal*, 44(2):193–204, 2002.

41 S. Goodman and S. Hedetniemi. On the Hamiltonian completion problem. In *Graphs and Combinatorics*, pages 262–272. Springer, 1974.

42 L. Gourvès, A. Harutyunyan, M. Lampis, and N. Melissinos. Filling crosswords is very hard. *Theoretical Computer Science*, 982:114275, 2024. doi:10.1016/J.TCS.2023.114275.

43 Mikhael Gromov. Hyperbolic groups. In *Essays in group theory*, pages 75–263. Springer, 1987.

44 I Ben-Arroyo Hartman. Variations on the Gallai-Milgram theorem. *Discrete Mathematics*, 71(2):95–105, 1988. doi:10.1016/0012-365X(88)90063-5.

45 Thore Husfeldt. Computing Graph Distances Parameterized by Treewidth and Diameter. In *11th International Symposium on Parameterized and Exact Computation (IPEC)*, volume 63, pages 16:1–16:11, 2017. doi:10.4230/LIPIcs.IPEC.2016.16.

46 Ioannis Katsikarelis, Michael Lampis, and Vangelis Th Paschos. Structural parameters, tight bounds, and approximation for (k, r)-center. *Discrete Applied Mathematics*, 264:90–117, 2019. doi:10.1016/J.DAM.2018.11.002.

47 Ioannis Katsikarelis, Michael Lampis, and Vangelis Th Paschos. Structurally parameterized d-scattered set. *Discrete Applied Mathematics*, 308:168–186, 2022. doi:10.1016/J.DAM.2020.03.052.

48 Ken-ichi Kawarabayashi, Yusuke Kobayashi, and Bruce Reed. The disjoint paths problem in quadratic time. *Journal of Combinatorial Theory, Series B*, 102(2):424–435, 2012. doi:10.1016/J.JCTB.2011.07.004.

49 Leon Kellerhals and Tomohiro Koana. Parameterized complexity of geodetic set. *Journal of Graph Algorithms and Applications*, 26(4):401–419, 2022. doi:10.7155/JGAA.00601.

50 Ton Kloks. *Treewidth: computations and approximations*. Springer, 1994. doi:10.1007/BFB0045375.

51 Tuukka Korhonen. A single-exponential time 2-approximation algorithm for treewidth. In *62nd IEEE Annual Symposium on Foundations of Computer Science, FOCS*, pages 184–192. IEEE, 2021. doi:10.1109/FOCS52979.2021.00026.

52 Sukhamay Kundu. A linear algorithm for the Hamiltonian completion number of a tree. *Information Processing Letters*, 5(2):55–57, 1976. doi:10.1016/0020-0190(76)90080-6.

53 Denis Kuperberg. Public communication on the Theoretical Computer Science Stack Exchange., 2023. URL: https://cstheory.stackexchange.com/a/52509.

54 Shaohua Li and Marcin Pilipczuk. Hardness of metric dimension in graphs of constant treewidth. *Algorithmica*, 84(11):3110–3155, 2022. doi:10.1007/S00453-022-01005-Y.

55 Guohui Lin, Zhipeng Cai, and Dekang Lin. Vertex covering by paths on trees with its applications in machine translation. *Information Processing Letters*, 97(2):73–81, 2006. doi:10.1016/J.IPL.2005.09.006.

56 William Lochet. A polynomial time algorithm for the k-disjoint shortest paths problem. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 169–178. SIAM, 2021. doi:10.1137/1.9781611976465.12.

57 Changhong Lu and Qing Zhou. Path covering number and L(2,1)-labeling number of graphs. *Discrete Applied Mathematics*, 161(13):2062–2074, 2013.

58 Paul Manuel. Revisiting path-type covering and partitioning problems. *CoRR ArXiv preprint*, 2018. arXiv:1807.10613.

59 Paul Manuel. On the isometric path partition problem. *Discussiones Mathematicae Graph Theory*, 41(4):1077–1089, 2021. doi:10.7151/DMGT.2236.

60 Mathieu Mari, Anish Mukherjee, Michał Pilipczuk, and Piotr Sankowski. Shortest disjoint paths on a grid. In *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 346–365. SIAM, 2024. doi:10.1137/1.9781611977912.14.

61 Simeon C. Ntafos and S. Louis Hakimi. On path cover problems in digraphs and applications to program testing. *IEEE Transactions on Software Engineering*, SE-5(5):520–529, 1979. doi:10.1109/TSE.1979.234213.

62 Jun-Jie Pan and Gerard J Chang. Induced-path partition on graphs with special blocks. *Theoretical Computer Science*, 370(1-3):121–130, 2007. doi:10.1016/J.TCS.2006.10.020.

63 Irena Penev, R. B. Sandeep, D. K. Supraja, and S. Taruni. Isometric path partition: a new upper bound and a characterization of some extremal graphs, 2025. arXiv:2505.19913.

64 Michał Pilipczuk. Problems parameterized by treewidth tractable in single exponential time: A logical approach. In *Proc. of the 36th International Symposium on Mathematical Foundations of Computer Science (MFCS 2011)*, volume 6907 of *Lecture Notes in Computer Science*, pages 520–531. Springer, 2011. doi:10.1007/978-3-642-22993-0_47.

65 Shlomit S Pinter and Yaron Wolfstahl. On mapping processes to processors in distributed systems. *International Journal of Parallel Programming*, 16(1):1–15, 1987. doi:10.1007/BF01408172.

66 Neil Robertson and Paul D. Seymour. Graph minors. XIII. the disjoint paths problem. *Journal of Combinatorial Theory, Series B*, 63(1):65–110, 1995. doi:10.1006/JCTB.1995.1006.

67 Lars Rohwedder and Karol Węgrzycki. Fine-grained equivalence for problems related to integer linear programming. In *16th Innovations in Theoretical Computer Science Conference (ITCS 2025)*, pages 83:1–83:18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2025. doi:10.4230/LIPIcs.ITCS.2025.83.

68 Ignasi Sau and Uéverton dos Santos Souza. Hitting forbidden induced subgraphs on bounded treewidth graphs. *Information and Computation*, 281:104812, 2021. doi:10.1016/J.IC.2021.104812.

69 Anand Srinivas and Eytan Modiano. Finding minimum energy disjoint paths in wireless ad-hoc networks. *Wireless Networks*, 11(4):401–417, 2005. doi:10.1007/S11276-005-1765-0.

70 Prafullkumar Tale. Geodetic set on graphs of constant pathwidth and feedback vertex set number. In *20th International Symposium on Parameterized and Exact Computation, IPEC*, volume 358 of *LIPIcs*, pages 28:1–28:14, 2025. doi:10.4230/LIPIcs.IPEC.2025.28.

71 Maximilian Thiessen and Thomas Gärtner. Active learning of convex halfspaces on graphs. In *Proceedings of the 35th Conference on Neural Information Processing Systems, NeurIPS*, volume 34, pages 23413–23425. Curran Associates, Inc., 2021. URL: https://proceedings.neurips.cc/paper/2021/hash/c4bf1e24f3e6f92ca9dfd9a7a1a1049c-Abstract.html.