Optimal Path Partitions in Subcubic and Almost-Subcubic Graphs

Tomáš Masařík
Institute of Informatics, University of Warsaw, Poland

Michał Włodarczyk
Institute of Informatics, University of Warsaw, Poland

Mehmet Akif Yıldız
Centrum Wiskunde & Informatica, Amsterdam, The Netherlands

Abstract

We consider the problem of partitioning the edges of a graph into as few paths as possible. This is a subject of the classic conjecture of Gallai and a recurring topic in combinatorics. Regarding the complexity of partitioning a graph optimally, Peroché [Discrete Appl. Math. 1984] proved that it is NP-hard already on graphs of maximum degree four, even when we only ask if two paths suffice.

We show that the problem is solvable in polynomial time on subcubic graphs and then we present an efficient algorithm for "almost-subcubic" graphs. Precisely, we prove that the problem is fixed-parameter tractable when parameterized by the edge-deletion distance to a subcubic graph. To this end, we reduce the task to model checking in first-order logic extended by disjoint-paths predicates (FO+DP) and then we employ the recent tractability result by Schirrmacher, Siebertz, Stamoulis, Thilikos, and Vigny [LICS 2024].

2012 ACM Subject Classification Theory of computation → Graph algorithms analysis

Keywords and phrases path partitions, parameterized algorithms, subcubic graphs, model checking, disjoint paths

Digital Object Identifier 10.4230/LIPIcs.WG.2026.32

Related Version *Full Version*: `https://arxiv.org/abs/2602.12925` [16]

Funding *Tomáš Masařík*: Supported by the Polish National Science Centre SONATA-17 grant number 2021/43/D/ST6/03312.
Michał Włodarczyk: Supported by the Polish National Science Centre SONATA-19 grant number 2023/51/D/ST6/00155.
Mehmet Akif Yıldız: This work was carried out during the tenure of an ERCIM "Alain Bensoussan" Fellowship Programme.

1 Introduction

The famous conjecture of Gallai [12] states that the edge set of any n-vertex connected graph can be partitioned into at most $\lceil n/2 \rceil$ paths. It has been confirmed on several graph classes, including planar graphs [2], graphs of degree at most five [3], graphs with no even degree vertices [5], and graphs of treewidth at most four [13]. In general the conjecture remains open and currently the best bound is $\lfloor 2n/3 \rfloor$ [5, 19]. Notably, Lovász proved that every graph can be decomposed using $\lfloor n/2 \rfloor$ paths and cycles [12].

In the recent years, path partitioning and covering problems have also been studied from the algorithmic perspective (see [15] for a survey), in particular in parameterized complexity. Depending on the variant, the goal to cover either the vertices or the edges of a given graph using as few paths as possible. One may require the paths to be pairwise disjoint and/or to be shortest paths [6, 1, 4, 9]. While the natural parameter is the number of paths [6, 1, 7], the problem has also been studied from the perspective of structural parameterization [4, 9, 7] and parameterization below a guarantee [8].

We consider the problem in the original variant studied by Gallai and for a graph G we define $\mathsf{pn}(G)$ as the smallest number of edge-disjoint paths in G whose union is $E(G)$. Notably, Peroché [17] showed that deciding whether $\mathsf{pn}(G) \le 2$ is NP-hard even for graphs G with maximum degree $\Delta(G) = 4$. First, we complement this hardness with a polynomial-time algorithm to compute $\mathsf{pn}(G)$ when G is subcubic, i.e., $\Delta(G) \le 3$. The sharp transition between $\Delta(G) = 3$ and $\Delta(G) = 4$ leads to the question of what happens in graphs that are *almost subcubic*. For a graph G, we define its *subcubic edge-deletion number* – $\mathsf{sen}(G)$ – as the smallest number of edges one needs to remove from G to make it subcubic (cf. [14]). We show that determining $\mathsf{pn}(G)$ is fixed-parameter tractable (FPT) with respect to the parameter $\mathsf{sen}(G)$, providing a significant extension of the case $\Delta(G) = 3$.

▶ **Theorem 1.** *There is an algorithm that, given a graph G, computes $\mathsf{pn}(G)$ in time $f(\mathsf{sen}(G)) \cdot |G|^{\mathcal{O}(1)}$, where f is a computable function.*

Organization of the paper. We begin with an informal exposition of the main technical ideas in Section 2. In Section 3 we fix notation and discuss basic graph-theoretical notions. The proof of Theorem 1 spans Sections 4-7. In particular, in Sections 5 and 6, we introduce the concept of a pattern and prove its several properties. Then, in Section 7, we reduce the problem to model checking and conclude the proof. Due to the space constraints, we omit the proofs of Lemmas 2, 6, 7, and 12, see [16] for details. We mark the respective statements with ★ symbol.

2 Overview

Subcubic graphs. Let $\mathsf{odd}(G)$ denote the number of odd-degree vertices in G. Clearly, $\mathsf{pn}(G) \ge \mathsf{odd}(G)/2$ because each odd-degree vertex must be an endpoint of some path. Moreover, when all vertices in G have odd degrees then the lower bound is attained (Corollary 4). We show that in a subcubic graph the degree-2 vertices can be effectively dissolved, resulting in a convenient formula for $\mathsf{pn}(G)$. An additional argument is necessary when G contains a *pan cycle*, i.e., a cycle with exactly one degree-3 vertex. Let $\mathrm{pan}(G)$ denote the number of such cycles in G. Then the following formula determines $\mathsf{pn}(G)$ of a subcubic graph G and entails a polynomial algorithm to compute $\mathsf{pn}(G)$.

▶ **Lemma 2** (Subcubic Path Partition ★)**.** *Let G be a connected graph with $\Delta(G) \le 3$. Then it holds that*

$$\mathsf{pn}(G) = \begin{cases} 2 & \text{if } G \text{ is a cycle or a subdivision of a diamond,} \\ \mathrm{odd}(G)/2 + \mathrm{pan}(G) & \text{otherwise.} \end{cases}$$

Towards general case: guessing a partial solution. We present a brief outline of the algorithm parameterized by $\mathsf{sen}(G)$, starting by an XP algorithm. Let $V_4 \subseteq V(G)$ be the set of vertices of degree at least four in G. It is easy to see that $k := \sum_{v \in V_4} \deg(v)$ is of order $\mathcal{O}(\mathsf{sen}(G))$. This also upper bounds the number of edge-disjoint paths that can pass through a vertex from V_4. Let $\mathcal{P}_{\mathsf{opt}}$ denote the family of such paths in some fixed optimal solution. Since $|\mathcal{P}_{\mathsf{opt}}| \le k$, we can afford guessing some information about each $P \in \mathcal{P}_{\mathsf{opt}}$. Let V_X denote the set of endpoints of all the paths in $\mathcal{P}_{\mathsf{opt}}$ union with V_4. For each path $P \in \mathcal{P}_{\mathsf{opt}}$, we guess the sequence of vertices from V_X that are visited by P – we call this the *trace* of P and a collection of traces is called a *pattern* (see Figure 3 on Page 7). The number of patterns is $f(k) \cdot n^{\mathcal{O}(k)}$ and we can enumerate all of them in XP time. For a fixed pattern F, we can check if there exists a path family $\mathcal{P}_F$ matching F (and compute it) using the known FPT algorithm for finding edge-disjoint paths parameterized by the number of paths [11].

Let F be the pattern encoding $\mathcal{P}_{\mathsf{opt}}$ and $\mathcal{P}_F$ be the computed family of edge-disjoint paths matching F. Next, let $G_F = G - \mathcal{P}_F$ be the graph obtained from G by removing all edges from the paths in $\mathcal{P}_F$. Similarly, let $G_{\mathsf{opt}} \coloneqq G - \mathcal{P}_{\mathsf{opt}}$; we have $\mathsf{pn}(G) = \mathsf{pn}(G_{\mathsf{opt}}) + |\mathcal{P}_{\mathsf{opt}}|$. Since every edge incident to V_4 is covered by some $P \in \mathcal{P}_F$, the graph G_F is subcubic and so $\mathsf{pn}(G_F)$ is determined by Lemma 2. The crux of the analysis is to ensure that $\mathsf{pn}(G_{\mathsf{opt}}) = \mathsf{pn}(G_F)$ for one pattern F, even though the family $\mathcal{P}_F$ may differ from $\mathcal{P}_{\mathsf{opt}}$.

The main observation is that $\mathsf{odd}(G_{\mathsf{opt}}) = \mathsf{odd}(G_F)$ as this depends only on $\mathsf{odd}(G)$ and the pattern F. Hence, the optimal solution must use at least $\mathsf{odd}(G_F)/2$ paths apart from the ones touching V_4. We take advantage of Lemma 2 to justify that in fact $\mathsf{odd}(G_F)/2 + |\mathcal{P}_F|$ paths suffice to partition $E(G)$. This becomes challenging when some connected component of G_F is a cycle or a subdivided diamond or it contains a pan cycle. In our analysis, we effectively rule out these cases by either preprocessing G appropriately or postprocessing the family $\mathcal{P}_F$ by extending some paths (see Lemma 21). Consequently, $\mathsf{pn}(G)$ can be computed as the minimum of $\mathsf{odd}(G_F)/2 + |\mathcal{P}_F|$ over all patterns F that can be realized using edge-disjoint paths, which yields an XP algorithm.

Incorporating logic. Now we explain how to improve the above XP algorithm to an FPT one. The running time bottleneck comes from guessing the set V_X and so we redefine a pattern[1] by replacing the vertices from $V_X \setminus V_4$ by variables $x_1, x_2, \dots, x_\ell$ in each trace and additionally storing the degree of each $v \in V_X \setminus V_4$. The number of such patterns is a function of k only and it is still sufficient to uniquely determine the value of $\mathsf{odd}(G - \mathcal{P}_F)$ for any $\mathcal{P}_F$ matching F.

To check if there exists a path family $\mathcal{P}_F$ matching F, we seek an assignment of the variables $x_1, x_2, \dots, x_\ell$ in $V(G) \setminus V_4$ that (1) obeys the degree specification and so that (2) the pairs of vertices which occur consecutive in the traces can be connected by edge-disjoint paths. Then we try to adjust the computed family $\mathcal{P}_F$ to ensure $\mathsf{pn}(G - \mathcal{P}_F) = \mathsf{odd}(G - \mathcal{P}_F)/2$. However, if the assignment places endpoints of one path on a so-called *bull cycle* (see Lemma 9), then $G - \mathcal{P}_F$ has a cyclic component that cannot be reduced without increasing $\mathsf{odd}(G - \mathcal{P}_F)$. Hence, we need to add requirement (3) that each resulting path is *bull-free*, i.e., it avoids the above configuration. If all these conditions can be satisfied, we call the pattern F *feasible*. Restricting to bull-free paths allows us to again express $\mathsf{pn}(G)$ as the minimum of $\mathsf{odd}(G_F)/2 + |\mathcal{P}_F|$ over all feasible patterns F.

The crucial observation is that both bull-freeness and the degree specification can be expressed in the first-order logic (FO) after appropriate preprocessing of the graph. This allows us to rewrite the question of whether F is feasible using logic: precisely, the FO logic extended by disjoint-paths predicates (FO+DP) [10]. We utilize the recent result of Schirrmacher, Siebertz, Stamoulis, Thilikos, and Vigny [18] who proved that model checking of FO+DP is FPT on bounded-degree graphs[2], when parameterized by the formula length. More precisely, they considered predicates involving paths that are internally vertex-disjoint but we show that this is also sufficient on subcubic graphs. This allows us to check if a pattern is feasible in time $f(k) \cdot n^{\mathcal{O}(1)}$. Since the number of patterns is a function of k, we obtain an FPT algorithm.

[1] The full definition of a pattern is slightly more technical and also stores information about occurrences of $N_G(V_4)$ on the paths. See Definition 16.

[2] The theorem in [18] is more general and works for every graph class that excludes some graph as a topological minor.

3 Preliminaries

Throughout the paper, we use standard graph theory notation and terminology. We always work with simple and undirected graphs. We write $H \subseteq G$ to mean H is a *subgraph* of the graph G, that is $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$. The graph *induced* by $F \subseteq E(G)$, denoted by $G[F]$, is the subgraph of G with vertex set consisting of those vertices incident to edges in F and with edge set F. The graph *induced* by $S \subseteq V(G)$, denoted by $G[S]$, is the subgraph of G with the vertex set S and all edges with both endpoints in S.

For a collection $\mathcal{Q}$ of graphs, we write $V(\mathcal{Q}) = \bigcup_{Q \in \mathcal{Q}} V(Q)$ and $E(\mathcal{Q}) = \bigcup_{Q \in \mathcal{Q}} E(Q)$. For a collection $\mathcal{Q}$ of subgraphs of G, we write $G - \mathcal{Q}$ for the graph obtained from G by deleting all the edges in $E(\mathcal{Q})$. We denote the *degree* of a vertex v by $\deg_G(v)$. The *maximum degree* of G is $\Delta(G) := \max\{\deg(v) : v \in V(G)\}$. We call a graph G *subcubic* if $\Delta(G) \leq 3$. We say G is a *triangle* if $|V(G)| = 3 = |E(G)|$. We say G is a *diamond* if $|V(G)| = 4$ and $|E(G)| = 5$.

For a path $P = v_1 v_2 \dots v_k$ and $1 \leq i < j \leq k$, we write $v_i P v_j$ for the path $v_i v_{i+1} \dots v_{j-1} v_j$, that is the unique (v_i, v_j)-path along P. For an (a,b)-path P and (b,c)-path Q with $V(P) \cap V(Q) = \{b\}$, we write $aPbQc$ for the (a,c)-path that is obtained by *gluing* paths P and Q at the vertex b. We also combine the two conventions to write concisely $aPbQc$ for the concatenation of the (a,b)-subpath of P and the (b,c)-subpath of Q.

The *length* of path P, denoted by $|P|$, is the number of edges in P. We say a graph G is a *subdivision* of a graph H if G can be obtained from H by successively replacing edges of H with paths. Two paths P, Q are *edge-disjoint* if $E(P) \cap E(Q) = \emptyset$. Paths P, Q are *internally vertex-disjoint* if no vertex of one path appears as an internal vertex on the other path.

For a graph G, a collection of edge-disjoint paths $\mathcal{P}$ is said to be a *path partition*[3] for G if $\bigcup_{P \in \mathcal{P}} E(P) = E(G)$. The *path number* of G is defined as

$$\mathsf{pn}(G) := \min\{r : \text{there exists a path partition } \mathcal{P} \text{ of } G \text{ with } |\mathcal{P}| = r\}.$$

We write $\mathrm{odd}(G)$ for the number of odd degree vertices in G. We say that a cycle C in G is a *pan cycle* if there exists a unique vertex $v \in V(C)$ with $\deg_G(v) = 3$, and $\deg_G(w) = 2$ for all other $w \in V(C)$. We write $\mathrm{pan}(G)$ for the number of pan cycles in G. Next, we say that a cycle C in G is a *bull cycle* if there are two distinct vertices $u, v \in V(C)$ with $\deg_G(u) = \deg_G(v) = 3$, and $\deg_G(w) = 2$ for all other $w \in V(C)$. See Figure 1 for an illustration.

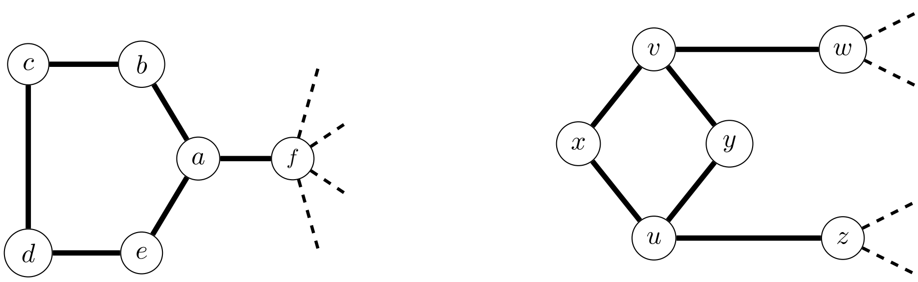


Figure 1 An example of a pan cycle $abcde$ ($\deg(a) = 3$ and $\deg(b) = \deg(c) = \deg(d) = \deg(e) = 2$) and a bull cycle $uyvx$ ($\deg(u) = \deg(v) = 3$ and $\deg(x) = \deg(y) = 2$).

[3] We remark that it is common in the combinatorial literature to use the name "path decomposition" and reserve "path partition" for the variant where one aims to partition the set of vertices. We decided to use "path partition" here to avoid confusion with the terminology related to pathwidth.

In any path partition of a graph G, each odd-degree vertex of G must be an endpoint of some path. This implies the following.

▶ **Observation 3.** *For any graph G, it holds that* $\mathsf{pn}(G) \geq \mathsf{odd}(G)/2$.

The proof of an old result of Lovász implies that the lower bound is attained when all the degrees are odd.

▶ **Corollary 4** ([12, 5])**.** *For a graph G with all the degrees odd, we have* $\mathsf{pn}(G) = \mathsf{odd}(G)/2$.

4 Almost-Subcubic Graphs

We begin with several observations that will come in useful in the proof of Theorem 1. First we preprocess the input to avoid inconvenient corner cases. In particular, we discard the pan cycles and shorten the bull cycles. The first facilitates application of Lemma 2 and the latter allows us to detect bull cycles in an FO formula.

▶ **Definition 5.** *We say that a graph is* nice *if it is: (a) connected, (b) not subcubic, (c) does not have any pan cycles, and (d) all bull cycles are triangles.*

A nice pair (G, V_4) *is given by a nice graph G and $V_4 \subseteq V(G)$ being the set of vertices of degree at least four in G.*

The following lemma justifies that we can restrict ourselves to nice graphs only.

▶ **Lemma 6** (★)**.** *Let G be a connected graph that is not subcubic. One can transform G in polynomial time into a nice graph G_0 so that* $\mathsf{pn}(G) = \mathsf{pn}(G_0) + \mathrm{pan}(G)$ *and* $\mathsf{sen}(G) = \mathsf{sen}(G_0)$.

Instead of working with the parameter $\mathsf{sen}(G)$ directly, we consider the number of edges touching the high-degree vertices, that is, $\mathsf{high}(G) = \sum_{v \in V_4} \deg_G(v)$.

▶ **Lemma 7** (★)**.** *Let (G, V_4) be a nice pair. Then* $\mathsf{high}(G) \leq 8 \cdot \mathsf{sen}(G)$.

The special treatment of bull cycles is related to the following property.

▶ **Definition 8** (Bull-free Paths)**.** *A* bull triangle *is a bull cycle of length three. The two degree-3 vertices of a bull triangle form a* bull-pair*. A path is* bull-free *if its endpoints do not form a bull-pair.*[4] *A family of paths $\mathcal{P}$ is bull-free if every $P \in \mathcal{P}$ is bull-free.*

The next lemma allows us to assume that in an optimal path partition, every path visiting V_4 is bull-free.

▶ **Lemma 9.** *Let (G, V_4) be a nice pair, $\mathcal{P}$ be a path partition for G of minimum size, and $\mathcal{P}_4 \subseteq \mathcal{P}$ be the subfamily of paths that visit at least one vertex from V_4. Then $\mathcal{P}_4$ is bull-free.*

Proof. Suppose that $\mathcal{P}_4$ is not bull-free and let $P \in \mathcal{P}_4$ be the path with endpoints u, v that are the degree-3 vertices of a bull triangle $T = (u, v, w)$. Because $V(P) \cap V_4 \neq \emptyset$, the path P cannot be fully contained in T. This implies that $w \notin V(P)$ and $uv \notin E(P)$. Then, the triangle T forms a connected component of the graph $G \setminus E(P)$. Hence, $\mathcal{P}$ must contain two paths Q_1, Q_2 covering T. But appending w to the v-end of P yields a valid path P', depicted in Figure 2. Then $\mathcal{P}' := \mathcal{P} \setminus \{P, Q_1, Q_2\} \cup \{P', wuv\}$ is a path partition for G with $|\mathcal{P}'| < |\mathcal{P}|$, a contradiction. ◀

[4] Here, "bull-free" does not refer to the usual notion of graphs with no induced bull subgraph in the structural graph theory literature.

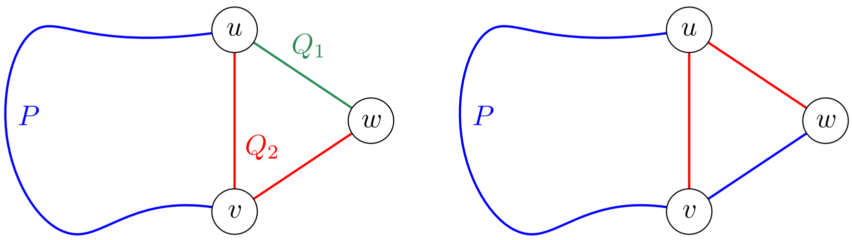


Figure 2 Extending the path P with the edge vw reduces the number of used paths by one.

5 Patterns

5.1 Traces and encoding

Let (G, V_4) be a nice pair. For a path partition $\mathcal{P}$ of G let $\mathcal{P}_4 \subseteq \mathcal{P}$ be the subfamily of paths that visit at least one vertex from V_4. It is easy to see that $|\mathcal{P}_4| \le \mathsf{high}(G)$ so we can afford guessing some information about each path $P \in \mathcal{P}_4$. Specifically, we will define a *pattern* encoding the endpoints of each $P \in \mathcal{P}_4$ and its visits in $N_G[V_4]$. However, we cannot afford guessing all the endpoints, so we will encode the relevant vertices outside $N_G[V_4]$ as variables and record their degrees. To make the exposition clearer, we first define a *trace*, being a representation of a single path, then we explain how the traces are extracted from $\mathcal{P}_4$, and finally we provide a full definition of a pattern.

▶ **Definition 10** (Trace). *Let (G, V_4) be a nice pair and ℓ be an integer. We define $X_\ell := \{x_1, \dots, x_\ell\}$. The elements of X_ℓ are called* variables.

*A (G, V_4, ℓ)-*trace *is a sequence over the alphabet $N_G[V_4] \cup X_\ell$ in which no symbol occurs more than once and no variable x_i appears next to any $v \in V_4$.*

The last condition reflects the fact that whenever a path P visits a vertex $v \in V_4$ then the consecutive vertices on P must belong to $N_G(v) \subseteq N_G[V_4]$ while the variables $x_1, \dots, x_\ell$ are meant to encode vertices outside $N_G[V_4]$. See Figure 3 for an example.

In the later arguments (Lemma 22) it will be helpful to avoid the same pair of consecutive symbols appearing twice in the traces. The following structure allows us to impose this property.

▶ **Definition 11** (Terminal Collection). *Let (G, V_4) be a nice pair and $\mathcal{P}$ be a family of edge-disjoint paths in G. A set of vertices $U \subseteq V(G)$ is called a* terminal collection *for $\mathcal{P}$ if*

1. *$N_G[V_4] \subseteq U$,*
2. *for each $P \in \mathcal{P}$ the endpoints of P belong to U,*
3. *for each $u, v \in U$, if two paths $P_1, P_2 \in \mathcal{P}$ visit both u and v then at least one of P_1, P_2 visits another vertex $w \in U$ between u and v.*

▶ **Lemma 12** (★). *Let (G, V_4) be a nice pair, $\mathcal{P}$ be a path partition of G, and $\mathcal{P}_4 \subseteq \mathcal{P}$ be the subfamily of paths that visit any vertex from V_4. Then $\mathcal{P}_4$ admits a terminal collection of size at most $16 \cdot \mathsf{high}(G)$.*

To facilitate working with the subfamily $\mathcal{P}_4 \subseteq \mathcal{P}$, we define a structure that encapsulates its properties.

▶ **Definition 13** (Covering family). *Let (G, V_4) be a nice pair and $\mathcal{Q}$ be a family of edge-disjoint paths in G. We say that $\mathcal{Q}$ is a* covering family *if every edge $e \in E(G)$ incident to V_4 is contained in some path from $\mathcal{Q}$.*

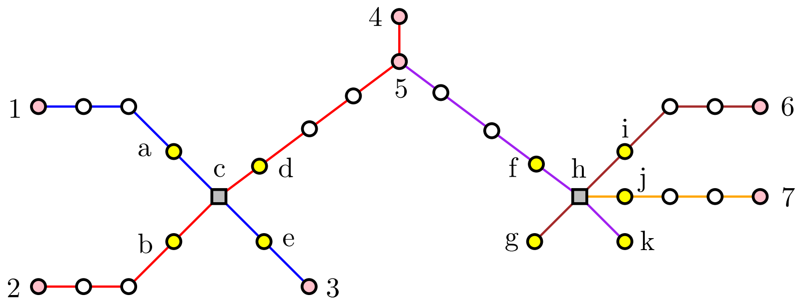


Figure 3 The vertices in V_4 (c and h) are gray squares, the ones in $N(V_4)$ are yellow disks labeled with the remaining letters, and the vertices in V_X are pink disks labeled with numbers. The highlighted vertices form a terminal collection for the covering family comprising the five paths represented by the edge colors. The pattern contains the following traces: red (x_2, b, c, d, x_5, x_4), blue (x_1, a, c, e, x_3), purple (x_5, f, h, k), brown (g, h, i, x_6), orange (h, j, x_7).

▶ **Definition 14** (Encoding). *Let (G, V_4) be a nice pair and $\mathcal{Q}$ be a covering family. Next, let $\mathcal{T}$ be a collection of (G, V_4, ℓ)-traces for some $\ell \in \mathbb{N}$ and d be a function $X_\ell \to \{1, 2, 3\}$. We say that the pair $(\mathcal{T}, d)$* encodes *$\mathcal{Q}$ if it can be obtained in the following way.*

Let U be some terminal collection for $\mathcal{Q}$. Next, let V_X be defined as $U \setminus N_G[V_4]$, $\ell = |V_X|$, and f be any bijection $X_\ell \to V_X$.

For each $P \in \mathcal{Q}$, we construct the trace T_P by orienting P in any direction, recording the occurrences of vertices from $N_G[V_4] \cup V_X$, and replacing each occurrence of $v \in V_X$ with $f^{-1}(v) \in X_\ell$. The obtained pair is $(\mathcal{T}, d)$ where $\mathcal{T}$ comprises traces T_P, for each $P \in \mathcal{Q}$, and $d\colon X_\ell \to \{1, 2, 3\}$ is defined as $d(x_i) = \deg_G(f(x_i))$.

The outcome of the above construction is not unique because the orientations of the paths, the terminal collection, and the bijection f are arbitrary.

5.2 Pattern definition

In order to define a combinatorial object describing the encoding, we first need more notation to work with traces.

▶ **Definition 15** (Trace Notation). *For a (G, V_4, ℓ)-trace $T = (t_1, t_2, t_3, \ldots, t_r)$ we define* $\mathsf{edges}(T)$ *as the set of those unordered pairs among t_1t_2, $t_2t_3, \ldots, t_{r-1}t_r$ that contain a symbol from V_4. Next, we define* $\mathsf{ends}(T)$ *as the set of the remaining pairs from the above list. For a collection $\mathcal{T}$ of traces, we define* $\mathsf{edges}(\mathcal{T})$ *(resp.* $\mathsf{ends}(\mathcal{T})$*) as the union of* $\mathsf{edges}(T)$ *(resp.* $\mathsf{ends}(T)$*) over $T \in \mathcal{T}$. For $y \in N_G[v_4] \cup X_\ell$ we define $\deg_T(y)$ as:*

0: *if y does not appear on T,*
1: *if y is an endpoint of T,*
2: *otherwise.*

We are finally ready to properly define a *pattern*.

▶ **Definition 16** (Pattern). *A (G, V_4, ℓ)-*pattern *is a pair $(\mathcal{T}, d)$, where $\mathcal{T}$ is a collection of (G, V_4, ℓ)-traces and d is a function $X_\ell \to \{1, 2, 3\}$ with the following properties.*

1. *The sets* $\mathsf{ends}(T_1), \mathsf{ends}(T_2)$ *are disjoint for distinct $T_1, T_2 \in \mathcal{T}$.*
2. *The sets* $\mathsf{edges}(T_1), \mathsf{edges}(T_2)$ *are disjoint for distinct $T_1, T_2 \in \mathcal{T}$.*
3. *The set* $\mathsf{edges}(\mathcal{T})$ *equals the set of edges incident to V_4 in G.*
4. *For each $x_i \in X_\ell$ we have $\sum_{T \in \mathcal{T}} \deg_T(x_i) \le d(x_i)$.*
5. *For each $v \in N_G(V_4)$ we have $\sum_{T \in \mathcal{T}} \deg_T(v) \le \deg_G(v)$.*

The conditions (2, 3) ensure that a pattern encodes all the information about the edges incident to V_4, i.e., which path utilizes which edges. The last two conditions reflect the fact that the number of times a vertex is "used" in a pattern is bounded by its degree. They will also come in useful for estimating the total number of different patterns. Next, we need to justify that any $(\mathcal{T}, d)$ obtained via Definition 14 is indeed a valid pattern. We also utilize Lemma 12 to bound the number of variables needed in an encoding.

▶ **Lemma 17.** *Let (G, V_4) be a nice pair, $\mathcal{Q}$ be a covering family, and $(\mathcal{T}, d)$ be a pair constructed according to Definition 14. Then $(\mathcal{T}, d)$ is a (G, V_4, ℓ)-pattern. Moreover, there exists a (G, V_4, ℓ)-pattern encoding $\mathcal{Q}$ which satisfies $\ell \le 16 \cdot \mathsf{high}(G)$.*

Proof. First we show that each $T \in \mathcal{T}$ is a (G, V_4, ℓ)-trace. Clearly, each symbol appears at most once on T because a path P visits each vertex at most once. Next, no variable x_i appears next to $v \in V_4$ because the vertices consecutive to v on P belong to $N_G[V_4]$ while $V_X \cap N_G[V_4] = \emptyset$.

Now we check the conditions of Definition 16. Since the set U was chosen as a terminal collection for $\mathcal{Q}$, no pair of vertices appears twice as consecutive in the collection of traces. This ensures that the first two conditions hold. Next, let $vu \in E(G)$ be an edge incident to $v \in V_4$. Then vu appears on some path $P \in \mathcal{Q}$ and so $vu \in \mathsf{edges}(\mathcal{T})$. To see the last two conditions, observe that when a trace T encodes a path P then for each $v \in N_G[V_4]$ we have $\deg_T(v) = \deg_P(v)$. Furthermore, when $x_i \in X_\ell$ and $u = f(x_i) \in V_X$ then $\deg_T(x_i) = \deg_P(u)$ and $d(x_i) = \deg_G(u)$.

Finally, we argue that ℓ can be bounded by $16 \cdot \mathsf{high}(G)$. In Definition 14 we can utilize any set U that forms a terminal collection for $\mathcal{Q}$. Lemma 12 shows that there exists such U of size at most $16 \cdot \mathsf{high}(G)$. ◀

6 Decoding a Pattern

We will now explain how $\mathsf{pn}(G)$ can be determined by examining patterns appearing in G.

▶ **Definition 18** (Odd Number of a Pattern). *Let (G, V_4) be a nice pair, $\ell \in \mathbb{N}$, and $(\mathcal{T}, d)$ be a (G, V_4, ℓ)-pattern. We say that a vertex $v \in N_G[V_4]$ (resp. variable $v \in X_\ell$)* loses oddity *if $\deg_G(v)$ (resp. $d(v)$) is <u>odd</u> and $\sum_{T \in \mathcal{T}} \deg_T(v)$ is <u>odd</u>. Similarly, a vertex $v \in N_G[V_4]$ (resp. variable $v \in X_\ell$)* gains oddity *if $\deg_G(v)$ (resp. $d(v)$) is <u>even</u> and $\sum_{T \in \mathcal{T}} \deg_T(v)$ is <u>odd</u>.*

We define the odd number $\mathsf{odd}(\mathcal{T}, d)$ *of a pattern $(\mathcal{T}, d)$ as the number of elements in $N_G[V_4] \cup X_\ell$ that gain oddity minus the number of elements that lose oddity.*

The utility of the above definition stems from the following lemma, which allows us to read $\mathsf{odd}(G - \mathcal{Q})$ by examining only the pattern encoding $\mathcal{Q}$.

▶ **Lemma 19.** *Let (G, V_4) be a nice pair, $\mathcal{Q}$ be a covering family in (G, V_4), $\ell \in \mathbb{N}$, and $(\mathcal{T}, d)$ be a (G, V_4, ℓ)-pattern encoding $\mathcal{Q}$. Then $\mathsf{odd}(G - \mathcal{Q}) = \mathsf{odd}(G) + \mathsf{odd}(\mathcal{T}, d)$.*

Proof. The quantity $\mathsf{odd}(G - \mathcal{Q}) - \mathsf{odd}(G)$ equals the number of vertices $v \in V(G)$ that are odd in $G - \mathcal{Q}$ and even in G, minus the number of vertices that are even in $G - \mathcal{Q}$ and odd in G. This transition occurs exactly when $\sum_{P \in \mathcal{Q}} \deg_P(v) = \sum_{T \in \mathcal{T}} \deg_T(v)$ is odd. Observe that when $v \in V(G)$ is not an endpoint of any path $P \in \mathcal{Q}$ then its oddity is not affected, so it suffices to inspect only the vertices recorded in the encoding. When $\sum_{P \in \mathcal{Q}} \deg_P(v)$ is odd then, depending on whether $\deg_G(v)$ is even or odd, vertex v gains or loses oddity, respectively. Consequently, $\mathsf{odd}(G - \mathcal{Q}) - \mathsf{odd}(G) = \mathsf{odd}(\mathcal{T}, d)$. ◀

The next two lemmas show that $\mathsf{pn}(G)$ can be expressed as $\min\left(\frac{\mathsf{odd}(G)+\mathsf{odd}(\mathcal{T},d)}{2}+|\mathcal{T}|\right)$ over all patterns $(\mathcal{T},d)$ that are *feasible* in G. Recall that for a path partition $\mathcal{P}$, we write $\mathcal{P}_4 \subseteq \mathcal{P}$ to denote the subfamily of paths visiting any vertex from V_4 and that $\mathcal{P}_4$ forms a covering family. We first show that a pattern encoding $\mathcal{P}_4$ provides a lower bound on $|\mathcal{P}|$.

▶ **Lemma 20.** *Let (G, V_4) be a nice pair, $\mathcal{P}$ be a path partition of G, $\mathcal{P}_4 \subseteq \mathcal{P}$ be the corresponding covering family, $\ell \in \mathbb{N}$, and $(\mathcal{T}, d)$ be a (G, V_4, ℓ)-pattern encoding $\mathcal{P}_4$. Then*

$$|\mathcal{P}| \geq \frac{\mathsf{odd}(G)+\mathsf{odd}(\mathcal{T},d)}{2}+|\mathcal{T}|.$$

Proof. Recall that $|\mathcal{T}| = |\mathcal{P}_4|$ by Definition 14. Let $G' := G - \mathcal{P}_4$. By Observation 3 we have $\mathsf{pn}(G') \geq \mathsf{odd}(G')/2$. Next, Lemma 19 yields $\mathsf{odd}(G') = \mathsf{odd}(G) + \mathsf{odd}(\mathcal{T}, d)$ and so

$$|\mathcal{P}| - |\mathcal{T}| = |\mathcal{P} \setminus \mathcal{P}_4| \geq \mathsf{pn}(G') \geq \frac{\mathsf{odd}(G')}{2} = \frac{\mathsf{odd}(G)+\mathsf{odd}(\mathcal{T},d)}{2}. \qquad ◀$$

The more complicated direction is to show that the odd number of a feasible pattern can be used to attain an upper bound on $\mathsf{pn}(G)$. This argument relies on Lemma 2 for subcubic graphs and requires the encoded covering family to be bull-free. Recall from Lemma 9 that this holds in any optimal solution.

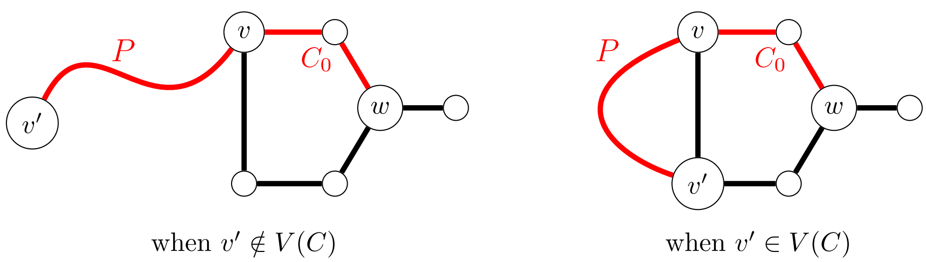
▶ **Lemma 21.** *Let (G, V_4) be a nice pair, $\mathcal{Q}$ be a bull-free covering family in (G, V_4), $\ell \in \mathbb{N}$, and $(\mathcal{T}, d)$ be a (G, V_4, ℓ)-pattern encoding $\mathcal{Q}$. Then*

$$\mathsf{pn}(G) \leq \frac{\mathrm{odd}(G)+\mathrm{odd}(\mathcal{T},d)}{2}+|\mathcal{T}|.$$

Proof. We prove the lemma by constructing a path partition $\mathcal{P}_0$ for G such that

$$|\mathcal{P}_0| = \frac{\mathrm{odd}(G)+\mathrm{odd}(\mathcal{T},d)}{2}+|\mathcal{T}|.$$

Let $G' := G - \mathcal{Q}$. As G' is subcubic, we can apply Lemma 2 for every subgraph of G'. The main idea is to extend some paths in $\mathcal{Q}$ to obtain another bull-free covering family $\mathcal{Q}'$ of the same size in a way that the subgraph $G'' := G - \mathcal{Q}'$ satisfies $\mathrm{odd}(G'') = \mathrm{odd}(G')$ and it has a path partition $\mathcal{Q}$ of size $\mathrm{odd}(G')/2$. This occurs when G'' has no pan cycle and no connected component which is either a cycle or a subdivided diamond. To construct $\mathcal{Q}'$, we gradually extend paths in $\mathcal{Q}$ in three steps. Note that $\deg_G(u) \leq 3$ for all non-isolated vertices $u \in V(G')$ since $\mathcal{Q}$ is a covering family. This, in particular, implies that for every vertex $u \in V(G')$ satisfying $\deg_{G'}(u) = 2 \neq \deg_G(u)$, we have $\deg_G(u) = 3$, and there exists a unique path in $\mathcal{Q}$ containing u, where u is in fact an endpoint of this path.



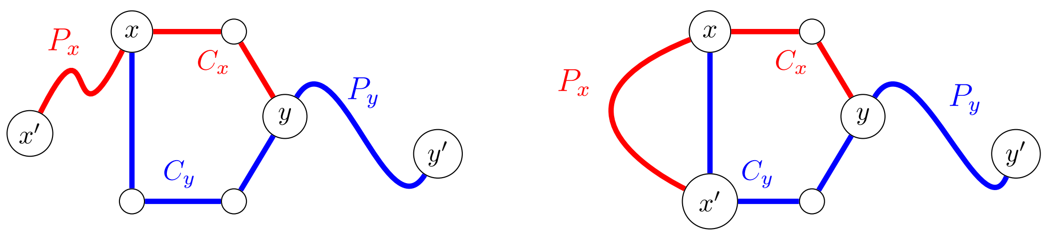
■ **Figure 4** We can extend $P \in \mathcal{Q}$ to discard the pan cycle C in G'.

First, assume that G' has a pan cycle C, and let $w \in V(C)$ be the unique vertex satisfying $\deg_{G'}(w) = 3$. It follows that $\deg_G(w) = 3$ using $\deg_{G'}(w) \le \deg_G(w) \le 3$. As G is nice, it contains no pan cycle and so there exists $v \in V(C)$ satisfying $\deg_G(v) \neq \deg_{G'}(v) = 2$. This implies that $v \neq w$, and that there exists an (v, v')-path $P \in \mathcal{Q}$ for some $v' \in V(G)$. Note that $V(C) \cap (V(P) \setminus \{v, v'\}) = \emptyset$ because every vertex from $V(G) \setminus V_4$ that is internal in P has degree at most one in G'. If $v' \in V(C)$, let C_0 be the (w, v)-path on C not containing v'. Otherwise, let C_0 be any of the two (w, v)-paths on C. By extending P to $v'PvC_0w$, depicted in Figure 4, we can decrease the number of pan cycles in G' by ensuring that $\mathsf{odd}(G')$ remains the same because we only reverse the parities of $\deg_{G'}(v)$ and $\deg_{G'}(w)$. Moreover, it is clear that v' and w do not form a bull-pair. By applying the same procedure successively for every pan cycle in G', we end up with a bull-free covering family $\mathcal{Q}_1$ satisfying $|\mathcal{Q}_1| = |\mathcal{Q}|$ such that the subgraph $G_1 := G - \mathcal{Q}_1$ satisfies $\mathsf{pan}(G_1) = 0$ and $\mathsf{odd}(G_1) = \mathsf{odd}(G')$.

Second, assume that G_1 has a connected component that is a cycle C. Since G is connected and not subcubic, there are edges between $V(C)$ and $V(G) \setminus V(C)$, which implies that $W := \{w \in V(C) : \deg_G(w) \neq 2\}$ is non-empty. Moreover, using $\mathsf{pan}(G) = 0$, we see that $|W| \neq 1$. For every $w \in W$, we write P_w for the unique path in $\mathcal{Q}_1$ containing w. In fact, w is an endpoint of P_w. As before, we observe that no internal vertex of P_w can lie on C.

If $|W| \ge 3$, we can choose $x, y \in W$ for which $P_x \neq P_y$. If $|W| = 2$, letting $W = \{x, y\}$, we claim that $P_x \neq P_y$. Indeed, when $|W| = 2$, C is a bull cycle in G. As G is nice, it follows that C is in fact a triangle. As $\mathcal{Q}_1$ is bull-free and $\{x, y\}$ form a bull-pair, it follows that $P_x \neq P_y$. Let x' and y' be the other endpoints of P_x and P_y, respectively. We will extend P_x and P_y to cover C depending on whether x' and y' lie on C.

If at most one of x' and y' belongs to C, without loss of generality, assume that $y' \notin V(C)$. If $x' \in V(C)$, let C_x be the (x, y)-path on C not containing x'. If $x' \notin V(C)$, let C_x be any (x, y)-path on C. Let C_y be the (x, y)-path on C other than C_x. Then, we can extend P_x and P_y to $x'P_xxC_xy$ and $y'P_yyC_yx$, respectively, depicted in Figure 5.



when $x', y' \notin V(C)$ when $x' \in V(C)$ and $y \notin V(C)$

Figure 5 When at most one of x' and y' belongs to C, then two (x, y)-paths of C can be appended to P_x and P_y.

If $x', y' \in V(C)$, without loss of generality, we can assume that $\{x, x', y, y'\}$ lie on the cycle C with the order (x, x', y, y') or (x, y, x', y') clockwise. For the first case, let C_x (resp. C_y) be the (x', y')-path on C containing x. We can extend P_x and P_y to $xP_xx'C_yy'$ and $yP_yy'C_xx'$, respectively, depicted in Figure 6 (left). For the second case, we can partition C into an (x, y)-path C_1, a (y, x')-path C_2, an (x', y')-path C_3, and a (y', x)-path C_4 such that $E(C) = E(C_1) \cup E(C_2) \cup E(C_3) \cup E(C_4)$. We can extend P_x and P_y to $yC_1xP_xx'C_3y'$ and $x'C_2yP_yy'C_4x$, respectively, as shown in Figure 6 (right).

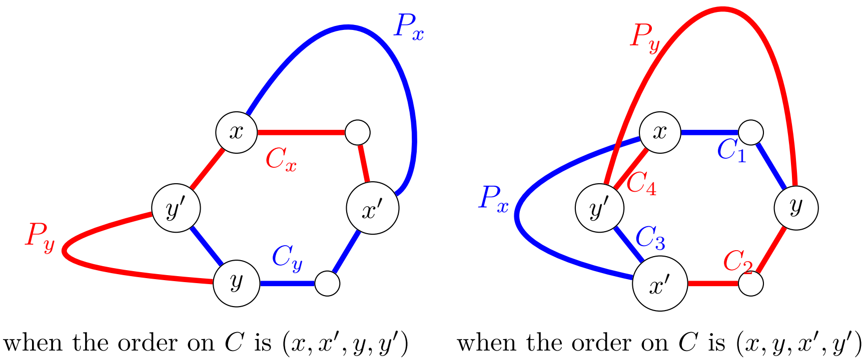


Figure 6 If $x', y' \in V(C)$, we can partition C into either two or four parts, depending on the order of x, y, x', y' on C, such that every part can be appended to P_x or P_y.

In all cases, we ensure that $\mathsf{odd}(G_1)$ remains the same as we only change degrees from 2 to 0, and we do not create a bull-pair. By applying the same procedure for every connected component that is a cycle, we end up with a bull-free covering family $\mathcal{Q}_2$ satisfying $|\mathcal{Q}_2| = |\mathcal{Q}_1| = |\mathcal{Q}|$ such that the subgraph $G_2 := G - \mathcal{Q}_2$ satisfies $\mathsf{pan}(G_2) = 0$, $\mathsf{odd}(G_2) = \mathsf{odd}(G_1) = \mathsf{odd}(G')$, and G_2 has no connected component that is a cycle.

Third, let G_2 have a connected component D that is a subdivision of a diamond. Let $x, y \in V(D)$ be the vertices with $\deg_{G_2}(x) = 3 = \deg_{G_2}(y)$. Note that $\deg_G(x) = 3 = \deg_G(y)$. Since G is not subcubic and connected, there are edges between $V(D)$ and $V(G) \setminus V(D)$, which implies that there exists $z \in V(D)$ satisfying $\deg_G(z) \neq \deg_{G'}(z) = 2$. Note that we can partition D into an (x, z)-path D_y and a (y, z)-path D_x with $E(D) = E(D_x) \cup E(D_y)$, depicted in Figure 7. It is clear that every $u \in V(D) \setminus \{x, y, z\}$ appears exactly one of D_x and D_y. Let P_z be the unique path in $\mathcal{Q}_2$ containing z, and let z' be the other endpoint of P_z. As $z' \notin V(D_x) \cap V(D_y)$, without loss of generality, we can assume that $z' \notin V(D_y)$. Then, we can extend P_z to $z'P_z z D_y x$ regardless of whether $z' \in V(D_x)$ or not. Note that we ensure that $\mathsf{odd}(G_2)$ remains the same because we only reverse the parities of $\deg_{G_2}(z)$ and $\deg_{G_2}(x)$. Effectively, the component C of G' with $\mathsf{odd}(C) = 2$ is replaced by the path D_x, also with $\mathsf{odd}(D_x) = 2$.

By this way we decrease the number of connected components of G_2 that are subdivisions of a diamond. It is also clear that we do not create any bull-pair. By applying this procedure exhaustively, we end up with a bull-free covering family $\mathcal{Q}'$ satisfying $|\mathcal{Q}'| = |\mathcal{Q}_2| = |\mathcal{Q}_1| = |\mathcal{Q}|$ such that the subgraph $G'' := G - \mathcal{Q}'$ satisfies $\mathsf{pan}(G') = 0$, $\mathsf{odd}(G'') = \mathsf{odd}(G_2) = \mathsf{odd}(G_1) = \mathsf{odd}(G')$, and no connected component of G'' is a cycle or a subdivision of a diamond.

We infer from Lemma 2 that $\mathsf{pn}(G'') = \mathsf{odd}(G'')/2 = \mathsf{odd}(G')/2$. Hence, there exists a path partition $\mathcal{P}''$ for G'' of size $\mathsf{odd}(G')/2$. Set $\mathcal{P}_0 = \mathcal{P}'' \cup \mathcal{Q}'$. Recall that $\mathsf{odd}(G') = \mathsf{odd}(G) + \mathsf{odd}(\mathcal{T}, d)$ by Lemma 19. As $|\mathcal{T}| = |\mathcal{Q}|$, we arrive at

$$\mathsf{pn}(G) \leq |\mathcal{P}_0| = \frac{\mathsf{odd}(G'')}{2} + |\mathcal{Q}'| = \frac{\mathsf{odd}(G')}{2} + |\mathcal{Q}| = \frac{\mathsf{odd}(G) + \mathsf{odd}(\mathcal{T}, d)}{2} + |\mathcal{T}|.$$

This concludes the proof of Lemma 21. ◀

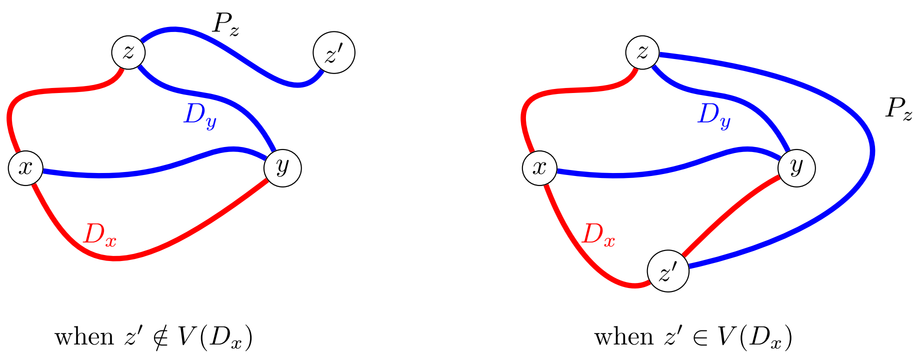


Figure 7 The partition of a diamond D into two edge-disjoint paths D_x and D_y. The path P_z can intersect with D only at z and z'.

7 Parameterized Algorithm

Recall that two paths are internally vertex-disjoint if no vertex of one path appears as an internal vertex on the other path.

In the algorithm we will need to check if a given pattern $(\mathcal{T}, d)$ is *feasible*, i.e., if there exists a bull-free covering family encoded by $(\mathcal{T}, d)$. First, we formulate feasibility via conditions that can be expressed using a logical formula over the subcubic graph $G - V_4$. In order to incorporate the logical meta-theorem from [18] we need to translate edge-disjointedness to internal vertex-disjointedness. The first direction follows from subcubicness while the opposite implication requires at least one endpoint of each two paths to be distinct. To ensure this, we exploit the concept of terminal collection from Definition 11.

▶ **Lemma 22.** *Let (G, V_4) be a nice pair, $\ell \in \mathbb{N}$, and $(\mathcal{T}, d)$ be a (G, V_4, ℓ)-pattern. Then there exists a bull-free covering family in (G, V_4) encoded by $(\mathcal{T}, d)$ if and only if there exists a mapping $f \colon X_\ell \cup N_G(V_4) \to V(G)$ that satisfies the following:*
1. *f is an identify on $N_G(V_4)$,*
2. *f is injective on X_ℓ and $f(X_\ell) \subseteq V(G) \setminus N_G[V_4]$,*
3. *$\deg_G(f(x_i)) = d(x_i)$ for each $x_i \in X_\ell$,*
4. *there exists a collection of internally vertex-disjoint paths in $G - V_4$ connecting the pairs in the collection $\{(f(y_1), f(y_2)) \colon y_1y_2 \in \mathsf{ends}(\mathcal{T})\}$,*
5. *for each $T \in \mathcal{T}$ with endpoints y_1, y_2, the pair $(f(y_1), f(y_2))$ is not a bull-pair in G.*

Proof. First we argue that when $(\mathcal{T}, d)$ encodes a covering family $\mathcal{Q}$ then the given conditions are satisfied. The only condition that is not obvious is (4). By the definition of the set $\mathsf{ends}(\mathcal{T})$, for each $y_1y_2 \in \mathsf{ends}(\mathcal{T})$ there exists a path $P \in \mathcal{Q}$ containing a subpath Q from $f(y_1)$ to $f(y_2)$. We need to prove that these subpaths are pairwise internally vertex-disjoint and contained in $G - V_4$. To see the latter fact, note that, by definition, $f(y_1), f(y_2) \notin V_4$ and Q cannot visit any $v \in V_4$ because then v would be recorded in the trace encoding P, implying that y_1, y_2 cannot be consecutive in this trace.

Next, we prove internal vertex-disjointedness. First consider the case where two subpaths Q_1, Q_2 contained in a single path $P \in \mathcal{Q}$. By the definition of a trace, these subpaths do not overlap on P. Since P visits each vertex at most once, the only common vertex on Q_1, Q_2 may be their common endpoint. Now suppose that $Q_1 \subseteq P_1$ and $Q_2 \subseteq P_2$ for $P_1, P_2 \in \mathcal{Q}$ and assume w.l.o.g. that there exists $v \in V(Q_1) \cap V(Q_2)$ and v is internal on Q_1. If v is also internal on Q_2 then $\deg_G(v) \geq 4$ because the paths P_1, P_2 are edge-disjoint. But then $v \in V_4$

and each occurrence of v is recorded in each trace $T \in \mathcal{T}$, implying that y_1, y_2 cannot be consecutive in the traces encoding P_1 or P_2; contradiction. It follows that v is an endpoint of Q_2. But then $v \in V_X$ and again every occurrence of v is recorded in each trace $T \in \mathcal{T}$, leading to the same contradiction as above. This concludes the justification of condition (4) and the first part of the proof.

Now we argue that the given conditions are sufficient for an existence of a bull-free covering family encoded by $(\mathcal{T}, d)$. Let $V_X = f(X_\ell)$. By condition (2) we have $|V_X| = \ell$ and $V_X \cap N_G[V_4] = \emptyset$. The function d is defined in Definition 14 exactly as in condition (3).

For each $y_1 y_2$ in $\mathsf{ends}(\mathcal{T})$ there exists a path $Q(y_1, y_2)$ from $f(y_1)$ to $f(y_2)$ so that these paths are internally vertex-disjoint. For a trace $T \in \mathcal{T}$ let P_T be the concatenation of *subpaths* $Q(y_1, y_2)$ for $y_1 y_2 \in \mathsf{ends}(T)$ and edges from $\mathsf{edges}(T)$, in the order specified in T. Then P_T is a walk and to argue that P_T is a path we need to prove that each vertex $v \in V(G)$ appears on P_T at most once. By the definition of a trace, each symbol occurs on T at most once, so there may be at most two considered edges or subpaths having v as an endpoint, and then they overlap at v. Next, the subpaths $Q(y_1, y_2)$ are pairwise internally vertex-disjoint, that is, if v occurs on any of them then it cannot be an internal vertex on another one. Hence P_T is indeed a path.

Next, we show that the paths P_{T_1}, P_{T_2} are edge-disjoint for distinct $T_1, T_2 \in \mathcal{T}$. Consider an edge uv that appears on P_{T_1}. If it is incident to V_4 then $uv \in \mathsf{edges}(T_1)$ and we use the fact that sets $\mathsf{edges}(T_1)$ and $\mathsf{edges}(T_2)$ are disjoint so uv cannot appear on P_{T_2}. Otherwise, uv appears on some subpath $Q(y_1, y_2)$ for $y_1 y_2$ in $\mathsf{ends}(T_1)$. If any of u, v is internal on $Q(y_1, y_2)$ then, by internal vertex-disjointedness, it cannot appear in any other path in the collection. In the remaining case, u, v are the endpoints of $Q(y_1, y_2)$. But the sets $\mathsf{ends}(T_1)$ and $\mathsf{ends}(T_2)$ are disjoint so u, v cannot be the endpoints of any subpath of P_{T_2}. Also, none of u, v can then also appear as an internal vertex in any subpath of P_{T_2}. Hence, the paths P_{T_1}, P_{T_2} are indeed edge-disjoint.

Finally, condition (5) directly corresponds to the definition of being bull-free. Each edge incident to V_4 is included in $\mathsf{edges}(\mathcal{T})$ (as demanded in Definition 16) and so it is used by one of the constructed paths. Therefore, applying Definition 14 with $\mathcal{Q}$ being the constructed path family and $V_X = f(X_\ell)$ yields the pattern $(\mathcal{T}, d)$. The lemma follows. ◀

7.1 Incorporating logic

For a graph G, integer k, and $s_1, t_1, \ldots, s_k, t_k \in V(G)$, we say that $\mathsf{dp}_k[(s_1, t_1), \ldots, (s_k, t_k)]$ holds in G if in G there are internally vertex-disjoint paths between s_i and t_i, for $i \in [1, k]$.

We consider first-order formulas extended by the disjoint-paths predicates over graph structures. There are three types of an atomic formula: equality $x = y$, edge relation $E(x, y)$, and disjoint-paths relation $\mathsf{dp}_k[(x_1, y_1), \ldots, (x_k, y_k)]$ for some $k \in \mathbb{N}$. An FO+DP formula is built from atomic formulas using the Boolean operators $\neg, \wedge, \vee$, as well as existential and universal quantification $\exists x$, $\forall x$.

A variable x not in the scope of any quantifier is called a *free variable*. We write $\phi(\bar{x})$ to indicate that $\bar{x} = (x_1, \ldots, x_\ell)$ is the tuple of free variables in ϕ. For a tuple $\bar{v} = (v_1, \ldots, v_\ell)$ from $V(G)$ we write $G \vDash \phi(\bar{v})$ if ϕ holds in G after substitution $x_i \leftarrow v_i$ for $i \in [1, \ell]$.

We summon the main theorem from [18] in a simplified form. The original theorem works for more general graphs classes, excluding a fixed graph H as a topological minor. Note that every subcubic graph excludes $K_{1,4}$ as a topological minor, so the statement below is indeed a corollary of the theorem from [18].

▶ **Theorem 23** ([18]). *There is an algorithm that, given a subcubic graph G, an* FO+DP *formula $\phi(\bar{x})$, and $\bar{v} \in V(G)^{|\bar{x}|}$, decides whether $G \vDash \phi(\bar{v})$ in time $f(\phi) \cdot |V(G)|^3$, where f is a computable function.*

This theorem, together with Lemma 22, allows us to efficiently test if a given pattern encodes some bull-free covering family.

▶ **Lemma 24.** *Let (G, V_4) be a nice pair, $k = \mathsf{high}(G)$, $\ell \in \mathbb{N}$, and $(\mathcal{T}, d)$ be a (G, V_4, ℓ)-pattern. Then we can check whether $(\mathcal{T}, d)$ encodes some bull-free covering family in time $g(k, \ell) \cdot |V(G)|^3$, where g is a computable function.*

Proof. We shall reduce the problem to FO+DP model checking over the subcubic graph $H = G - V_4$. Let us order $N_G(V_4)$ arbitrarily as $\bar{v} = (v_1, v_2, \dots, v_r)$. Note that $r \leq k$. We introduce free variables $\bar{y} = (y_1, y_2, \dots, y_r)$ so that y_i will be assigned to v_i.

We write formula $\phi(\bar{y})$ starting from existential quantifiers $\exists x_1 \exists x_2 \dots \exists x_\ell$ which correspond to an assignment $X_\ell \to V(H)$. Together with the free variables, these encode a function $f \colon X_\ell \cup N_G(V_4) \to V(H)$ which is an identity on $N_G(V_4)$. Inside the quantifiers we need to check if f satisfies the conditions listed in Lemma 22.

First, it is easy to ensure that $f|_{X_\ell}$ is injective by a conjunction of $\binom{\ell}{2}$ inequalities $x_i \neq x_j$. Similarly, we enforce that $f(X_\ell) \cap N_G(V_4) = \emptyset$ by writing $\ell \cdot r$ inequalities $x_i \neq y_j$.

The next condition states that $\deg_G(f(x_i)) = d(x_i)$ for each $i \in [\ell]$. Note that $f(x_i) \notin N_G(v_4)$ implies that $\deg_G(f(x_i)) = \deg_H(f(x_i))$. Then for each $c \in \{1, 2, 3\}$ it is easy to write an FO formula checking that the degree of x_i in H equals c.

Condition (4) states that the pairs of vertices in $\mathsf{ends}(\mathcal{T})$ are connected by internally vertex-disjoint paths in H. This directly corresponds to a predicate $\mathsf{dp}_h[(s_1, t_1), \dots, (s_h, t_h)]$ where the variables s_i, t_i correspond to some x_j or y_j depending on whether it is a symbol from X_ℓ or $N_G(V_4)$. Note that due to the degree bound each symbol can appear at most 3 times in $\mathsf{ends}(\mathcal{T})$ hence $h \leq 3 \cdot (k + \ell)$.

Next, we need to ensure that for each trace $T \in \mathcal{T}$ the vertices assigned to the endpoints of T do not form a bull-pair in G. Recall that (u, v) is a bull-pair in G when $\deg_G(u) = \deg_G(v) = 3$, $uv \in E(G)$, and they have a common neighbor w of degree 2. We show that this can be expressed by an FO formula over H. We write this formula only for those traces whose endpoints have degree 3 in G – this information is known upfront because the assignment of each y_i is fixed and the degree of x_i is encoded using the function d. Suppose this is the case. Then $u, v, w \in V(H)$. Moreover, u, v are the only neighbors of w in G. Since $u, v \notin V_4$ we infer that $\deg_H(w) = \deg_G(w) = 2$. So (u, v) is a bull-pair in G if and only if $uv \in E(H)$ and they have a common neighbor w of degree 2 in H. This again can be easily expressed in FO over the graph H.

We have constructed an FO+DP formula $\phi(\bar{y})$ so that $H \vDash \phi(\bar{v})$ holds if and only if there exists a bull-free covering family in (G, V_4) encoded by $(\mathcal{T}, d)$. The length of this formula is $\mathcal{O}((k + \ell)^2)$. Finally, by Theorem 23 we can test if $H \vDash \phi(\bar{v})$ in time $g(k, \ell) \cdot |V(G)|^3$. ◀

7.2 Wrapping-up

The final ingredient towards Theorem 1 is a bound on the number of different patterns to consider.

▶ **Lemma 25.** *Let (G, V_4) be a nice pair, $k = \mathsf{high}(G)$, and $\ell \in \mathbb{N}$. Then the number of (G, V_4, ℓ)-patterns is $\exp(\mathcal{O}((k + \ell) \log(k + \ell)))$ and they can be enumerated in such running time.*

Proof. We inspect Definition 16 to estimate the number of occurrences of each symbol from $\Sigma = N_G[V_4] \cup X_\ell$ is a pattern. Due to conditions (4) and (5) and the fact that symbols do not repeat in a single trace, each symbol from $N_G(V_4) \cup X_\ell$ can appear at most 3 times in a pattern. Condition (2) implies that the number of occurrences of each $v \in V_4$ is bounded by $\deg_G(v)$. Therefore, we can bound the total length of all traces by $3 \cdot (N_G(V_4) + \ell) + \sum_{v \in V_4} \deg_G(v) \le 4k + 3\ell$.

Consequently, every (G, V_4, ℓ)-pattern can be obtained by choosing a word of length at most $4k + 3\ell$ over the alphabet Σ, splitting it into subwords, and choosing function $d \colon X_\ell \to \{1, 2, 3\}$. Since $|\Sigma| \le 2k + \ell$, we can estimate the number of possibilities as $\exp(\mathcal{O}((k+\ell)\log(k+\ell)))$. The analysis above translates directly to an algorithm enumerating all the (G, V_4, ℓ)-patterns. ◀

Proof of Theorem 1. Each connected component of G can be treated independently. Moreover, Lemma 6 allows us to assume that we work with a nice pair (G, V_4). Let $k = \mathsf{high}(G)$; it is bounded by $8 \cdot \mathsf{sen}(G)$ due to Lemma 7. Hence, it suffices to give an FPT algorithm parameterized by k.

The algorithm enumerates all (G, V_4, ℓ)-patterns for $\ell \le 16 \cdot k$ using Lemma 25 in time $\exp(\mathcal{O}(k \log k))$. Then for each such pattern $(\mathcal{T}, d)$ we test if it encodes some bull-free covering family. Lemma 24 allows us to do this in time $g(k, 16 \cdot k) \cdot |V(G)|^3$. We output the minimum of $\frac{\mathrm{odd}(G)+\mathrm{odd}(\mathcal{T},d)}{2} + |\mathcal{T}|$ over all patterns $(\mathcal{T}, d)$ that passed the test.

We prove the correctness of this algorithm. Let α denote the computed value. Lemma 21 implies that $\mathsf{pn}(G) \le \alpha$. Let $\mathcal{P}$ be an optimal path partition of size $\mathsf{pn}(G)$ and $\mathcal{P}_4 \subseteq \mathcal{P}$ be the corresponding covering family. By Lemma 9 the family $\mathcal{P}_4$ is bull-free. By Lemma 17 there exists a (G, V_4, ℓ)-pattern $(\mathcal{T}, d)$ with $\ell \le 16 \cdot k$ that encodes $\mathcal{P}_4$. Consequently, $(\mathcal{T}, d)$ will be considered by the algorithm and so $\alpha \le \frac{\mathrm{odd}(G)+\mathrm{odd}(\mathcal{T},d)}{2} + |\mathcal{T}|$. Finally, we apply Lemma 20 to obtain that $\frac{\mathrm{odd}(G)+\mathrm{odd}(\mathcal{T},d)}{2} + |\mathcal{T}| \le |\mathcal{P}|$. We infer that $\mathsf{pn}(G) \le \alpha \le |\mathcal{P}| = \mathsf{pn}(G)$. Hence, $\alpha = \mathsf{pn}(G)$. This concludes the main proof. ◀

8 Conclusion

We have presented an FPT algorithm to compute $\mathsf{pn}(G)$ parametrized by the edge-deletion distance of G to a subcubic graph. A natural follow-up question is whether this parameter can be replaced by the vertex-deletion distance to a subcubic graph, which may be arbitrarily smaller than the edge-deletion distance. Is this problem in FPT or in XP? This question is non-trivial already for parameter value one: is there a polynomial algorithm that computes $\mathsf{pn}(G)$ of a graph G with only one vertex of degree higher than three?

While the Gallai conjecture remains open, Lovász proved that n paths and cycles suffice to partition the edges of every graph on $2n$ vertices [12]. In the regime of "parameterization above/below a guarantee" one asks whether we can effectively compute a solution that is slightly better than a certain combinatorial bound (see, e.g. [8]). The Lovász' bound gives rise to the following problem: given a graph on $2n$ vertices, determine whether $E(G)$ can be partitioned into $n - k$ path and cycles. Is this problem in FPT or in XP for the parameter k? As before, the existence of a polynomial algorithm for $k = 1$ is already open.

References

1 Julien Baste, Lucas De Meyer, Ugo Giocanti, Etienne Objois, and Timothé Picavet. A polynomial bound on the pathwidth of graphs edge-coverable by k shortest paths, 2025. `doi:10.48550/arXiv.2510.02901`.

2 Alexandre Blanché, Marthe Bonamy, and Nicolas Bonichon. Gallai's path decomposition in planar graphs, 2021. `arXiv:2110.08870`.

3 Marthe Bonamy and Thomas J. Perrett. Gallai's path decomposition conjecture for graphs of small maximum degree. *Discrete Mathematics*, 342(5):1293–1299, 2019. doi:10.1016/j.disc.2019.01.005.

4 Dibyayan Chakraborty, Oscar Defrain, Florent Foucaud, Mathieu Mari, and Prafullkumar Tale. Parameterized complexity of isometric path partition: treewidth and diameter, 2025. doi:10.48550/arXiv.2508.05448.

5 Nathaniel Dean and Mekkia Kouider. Gallai's conjecture for disconnected graphs. *Discrete Mathematics*, 213(1–3):43–54, 2000. doi:10.1016/S0012-365X(99)00167-3.

6 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.

7 Henning Fernau, Florent Foucaud, Kevin Mann, Utkarsh Padariya, and Rajath Rao K.N. Parameterizing path partitions. *Theoretical Computer Science*, 1028:115029, 2025. doi:10.1016/j.tcs.2024.115029.

8 Fedor V. Fomin, Petr A. Golovach, Nikola Jedličková, Jan Kratochvíl, Danil Sagunov, and Kirill Simonov. Path cover, Hamiltonicity, and independence number: An FPT perspective, 2025. arXiv:2403.05943.

9 Florent Foucaud, Atrayee Majumder, Tobias Mömke, and Aida Roshany-Tabrizi. Polynomial-time algorithms for path cover and path partition on trees and graphs of bounded treewidth, 2025. doi:10.48550/arXiv.2511.07160.

10 Petr A. Golovach, Giannos Stamoulis, and Dimitrios M. Thilikos. Model-checking for first-order logic with disjoint paths predicates in proper minor-closed graph classes. In Nikhil Bansal and Viswanath Nagarajan, editors, *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA 2023, Florence, Italy, January 22–25, 2023*, pages 3684–3699. SIAM, 2023. doi:10.1137/1.9781611977554.CH141.

11 Ken-ichi Kawarabayashi and Paul Wollan. A shorter proof of the graph minor algorithm: the unique linkage theorem. In *Proceedings of the Forty-Second ACM Symposium on Theory of Computing*, STOC '10, pages 687–694, New York, NY, USA, 2010. Association for Computing Machinery. doi:10.1145/1806689.1806784.

12 László Lovász. On covering of graphs. In Paul Erdős and Gyula O. H. Katona, editors, *Theory of Graphs (Proceedings of the Colloquium held at Tihany, Hungary, September 1966)*, pages 231–236. Academic Press, New York, 1968.

13 Changhong Lu and Niping Yi. The problem of path decomposition for graphs with treewidth at most 4. *Discrete Mathematics*, 347(6):113957, 2024. doi:10.1016/j.disc.2024.113957.

14 Diptapriyo Majumdar and Venkatesh Raman. Structural parameterizations of undirected feedback vertex set: FPT algorithms and kernelization. *Algorithmica*, 80(9):2683–2724, 2018. doi:10.1007/S00453-018-0419-4.

15 Paul Manuel. Revisiting path-type covering and partitioning problems, 2018. arXiv:1807.10613.

16 Tomáš Masařík, Michał Włodarczyk, and Mehmet Akif Yıldız. Optimal path partitions in subcubic and almost-subcubic graphs, 2026. doi:10.48550/arXiv.2602.12925.

17 Bernard Péroche. NP-completeness of some problems of partitioning and covering in graphs. *Discrete Applied Mathematics*, 8(2):195–208, 1984. doi:10.1016/0166-218X(84)90101-X.

18 Nicole Schirrmacher, Sebastian Siebertz, Giannos Stamoulis, Dimitrios M. Thilikos, and Alexandre Vigny. Model checking disjoint-paths logic on topological-minor-free graph classes. In Paweł Sobocinski, Ugo Dal Lago, and Javier Esparza, editors, *Proceedings of the 39th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2024, Tallinn, Estonia, July 8-11, 2024*, pages 68:1–68:12. ACM, 2024. doi:10.1145/3661814.3662089.

19 Lirong Yan. *On path decompositions of graphs.* PhD thesis, Arizona State University, 1998. Thesis (Ph.D.). ProQuest LLC, Ann Arbor, MI. MR2697319. ISBN: 978-0591-80735-6. URL: https://www.proquest.com/openview/59f62a730dfd1ec06fefb62e49e2f83d/.