

INDUCED CYCLES OF MANY LENGTHS

MARIA CHUDNOVSKY[†] AND ILYA MAIER[‡]

ABSTRACT. Let G be a graph and let $\text{cl}(G)$ be the number of distinct induced cycle lengths in G . We show that for $c, t \in \mathbb{N}$, every graph G that does not contain an induced subgraph isomorphic to K_{t+1} or $K_{t,t}$ and satisfies $\text{cl}(G) \leq c$ has bounded treewidth. As a consequence, we obtain a polynomial-time algorithm for deciding whether a graph G contains induced cycles of at least three distinct lengths.

1. INTRODUCTION

The set of all positive integers is denoted by \mathbb{N} and the set of all positive integers no greater than k is denoted by \mathbb{N}_k . For a set X , we denote the set of all subsets of X by 2^X . Graphs in this paper have finite vertex sets, no loops, and no parallel edges. Let G and H be two graphs. G is said to *contain* H if G contains an induced subgraph isomorphic to H . G is *H -free* if G does not contain H . For $X \subseteq V(G)$, we denote by $G[X]$ the subgraph of G induced by X , and we refer to X and $G[X]$ interchangeably. For standard graph theoretic terminology, the reader is referred to [8].

For a graph $G = (V(G), E(G))$, a *tree decomposition* (T, χ) of G consists of a tree T and a map $\chi : V(T) \rightarrow 2^{V(G)}$ with the following properties:

- (i) For every $v \in V(G)$, there exists $t \in V(T)$ such that $v \in \chi(t)$.
- (ii) For every $uv \in E(G)$, there exists $t \in V(T)$ such that $u, v \in \chi(t)$.
- (iii) For every $v \in V(G)$, the subgraph of T induced by $\{t \in V(T) \mid v \in \chi(t)\}$ is connected.

For every $t \in V(T)$, we refer to t as a *node of* and to $\chi(t)$ as a *bag of* (T, χ) . The *width* of a tree decomposition (T, χ) is $\max_{t \in V(T)} |\chi(t)| - 1$. The *treewidth* of G , denoted by $\text{tw}(G)$, is the minimum width of a tree decomposition of G . Additionally, we define $\text{cl}(G)$ to be the number of distinct induced cycle lengths in G , i.e.,

$$\text{cl}(G) = |\{l \in \mathbb{N} \mid G \text{ has an induced cycle of length } l\}|.$$

Berger, Seymour, and Spirkl [3] provided a polynomial-time algorithm for deciding whether there is a non-shortest u - v path in G for $u, v \in V(G)$. This algorithm can be straightforwardly applied to test whether all induced cycles in G have the same length. However, there is no simple way to apply this algorithm to test whether a graph G also contains induced cycles

Date: February 17, 2026.

[†] Princeton University, Princeton, NJ, USA. Supported by NSF Grant DMS-2348219, NSF Grant CCF-2505100, AFOSR grant FA9550-25-1-0275 and a Guggenheim Fellowship.

[‡] Princeton University, Princeton, NJ, USA. Supported by a scholarship of the German Academic Exchange Service (DAAD).

of at least three distinct lengths. In this paper, we provide a polynomial-time algorithm for doing exactly that.

Theorem 1.1. *Let G be a graph. There exists an algorithm with running time $O(|V(G)|^{22})$ that decides whether $\text{cl}(G) \geq 3$ and outputs a list of three induced cycles of distinct lengths if they exist.*

As a generalized version and a core subroutine of this algorithm, we also show that we can decide if $\text{cl}(G) \geq c$ for every $c \in \mathbb{N}$ in linear time given that G has bounded treewidth.

Theorem 1.2. *Let $c, k \in \mathbb{N}$ be constants and let G be a graph with $\text{tw}(G) \leq k$. There exists an algorithm with running time $O(|V(G)|)$ that decides whether $\text{cl}(G) \geq c$ and outputs a list of c induced cycles of distinct lengths if they exist.*

The majority of this paper is devoted to the proof of the following structural result that identifies a necessary condition on the structure of graphs with large treewidth.

Theorem 1.3. *For all $c, t \in \mathbb{N}$, there is a constant $f_{1.3} = f_{1.3}(c, t) \in \mathbb{N}$ such that for every graph G with $\text{tw}(G) \geq f_{1.3}$ one of the following holds.*

- (a) G contains K_{t+1} or $K_{t,t}$.
- (b) $\text{cl}(G) \geq c$.

1.1. Key structures. We start by defining some of the key structures used throughout this paper. For each structure S in G defined here, we will use S and the subgraph induced by G on the vertex set of S interchangeably.

For disjoint subsets $X, Y \subseteq V(G)$, we say that X and Y are *anticomplete* (or that X is *anticomplete to* Y) if there is no edge in G with an end in X and an end in Y , and that X *has a neighbor in* Y if X is not anticomplete to Y . For $x \in V(G)$, we say that x is *anticomplete to* Y if $\{x\}$ and Y are anticomplete. For a set \mathcal{X} of subsets of $V(G)$, we write $V(\mathcal{X}) = \bigcup_{X \in \mathcal{X}} X$.

An (s, d) -*dome* in G is a pair (S, P) such that

- S is a subdivided star in G with leaf set L of size s ;
- P is a path in G disjoint from S ;
- $V(S) \setminus L$ and $V(P)$ are anticomplete in G ;
- every vertex $v \in L$ has a neighbor in P ; and
- every vertex $u \in P$ is adjacent to at most d vertices in L (see Figure 1).

The *root* of the dome (S, P) is the unique vertex of S that has degree at least three in S . A dome is *aligned* if there exists an ordering l_1, \dots, l_s of the vertices of L and s pairwise disjoint subpaths u_i - P - v_i of P for $i \in \mathbb{N}_s$ (possibly of length zero) such that

- there is an end u of P such that P traverses $u, u_1, v_1, \dots, u_s, v_s$ in this order; and
- for every $i \in \mathbb{N}_s$, all neighbors of l_i in P are contained in u_i - P - v_i .

Note that for every $d \in \mathbb{N}$, an aligned (s, d) -dome is also an aligned $(s, 1)$ -dome.

Let $l, q \in \mathbb{N}$. We define a q -*banana* in G with ends x, y , denoted $B_{x,y}$, to be a pair $(\{x, y\}, \mathcal{P})$ where $\mathcal{P} = \{P_1, \dots, P_q\}$ is a set of q pairwise internally disjoint x - y paths. We define a q -*theta* in G with ends x, y , denoted $\Theta_{x,y}$, to be a q -banana in G such that for all $i \neq j \in \mathbb{N}_q$, the paths $P_i \setminus \{x, y\}$ and $P_j \setminus \{x, y\}$ are anticomplete to each other (see

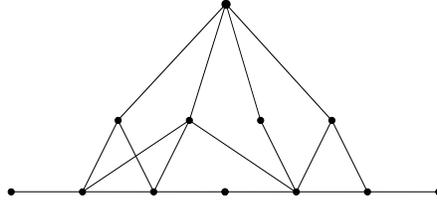


FIGURE 1. A $(4, 3)$ -dome that is not aligned.

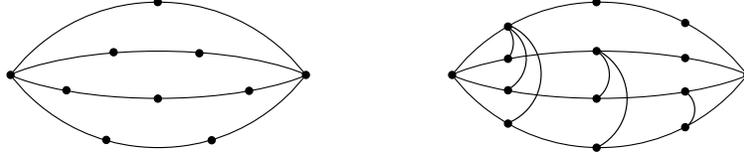


FIGURE 2. A 4-theta (left) and a rigid 4-banana (right).

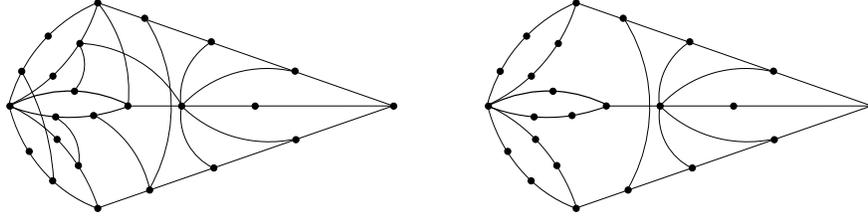


FIGURE 3. A $(3, 2)$ -kite (left) and a clean $(3, 2)$ -kite (right).

Figure 2). We call a q -banana with ends x, y *rigid* if for all $i, j \in \mathbb{N}_q$, the paths $P_i \setminus \{x, y\}$ and $P_j \setminus \{x, y\}$ are not anticomplete to each other (see Figure 2). We say that a banana $(\{x, y\}, \mathcal{P})$ is l -dense if $V(\mathcal{P})$ does not contain an l -theta with ends x, y .

For a triple of distinct vertices x, y, z , we define an (x, z) -tailed q -theta in G , denoted $\Theta_{x,y,z}$, to be a pair $(\Theta_{x,y}, P)$ such that

- $\Theta_{x,y}$ is a q -theta in G ;
- P is a $y-z$ path in G , called a *tail*;
- $P \setminus \{y\}$ and $V(\Theta_{x,y}) \setminus \{y\}$ are anticomplete.

For $k, q \in \mathbb{N}$, we define an (x, z) -tailed (k, q) -kite \mathcal{K} in G to be a set of k (x, z) -tailed q -thetas $\Theta_{x,y_1,z}, \dots, \Theta_{x,y_k,z}$ in G such that for all distinct $i, j \in \mathbb{N}_k$, $V(\Theta_{x,y_i,z}) \cap V(\Theta_{x,y_j,z}) = \{x, z\}$ (see Figure 3). For $i \in \mathbb{N}_k$, let $\Theta_{x,y_i,z} = (\Theta_{x,y_i}, P_i)$. We call a (k, q) -kite \mathcal{K} *clean* if for every $i \in \mathbb{N}_k$, every vertex of $V(\Theta_{x,y_i}) \setminus \{x, y_i\}$ has degree two in \mathcal{K} , and the set $\{y_1, \dots, y_k\}$ is a stable set (see Figure 3).

1.2. Proof outline and organization. We start by outlining the proof of Theorem 1.3. In Section 3, we show that if a graph G contains one of the basic obstructions (see Section 3 for a precise statement) or a large dome, then $\text{cl}(G) \geq c$. In Section 4, we apply this result to reduce Theorem 1.3 to Theorem 4.1, which states that a graph containing a large dense banana contains a large dome. We then prove Theorem 4.1 in Sections 5, 6 and 7.

In Section 5, we analyze the structure of rigid bananas. We show how to obtain either a theta $\Theta_{x,y}$ where each vertex in $N(x)$ has degree two in G , or a large dome. In Section 6, we apply this result to obtain many tailed thetas, which together form a kite. Then, we show how to either make this kite clean or obtain a large dome. In Section 7, we recursively obtain a sequence of kites, which we use to obtain a large dome (S, P) . The last kite in this sequence provides the path P , and all the previous kites provide the paths of the subdivided star S . This completes the proof of Theorem 4.1, and thus, also Theorem 1.3.

Then, in Section 8 we describe our “key subroutine”, Theorem 1.2. The input of this algorithm is a graph G with bounded treewidth. The algorithm finds a tree decomposition (T, χ) of G and, using dynamic programming over T , either finds c induced cycles of distinct lengths in G , or determines that $\text{cl}(G) < c$.

Finally, in Section 9, we prove Theorem 1.1. Suppose that G contains K_3 or $K_{2,2}$. Then, we can build on the algorithm that finds a non-shortest u - v path from [3] to decide whether $\text{cl}(G) \geq 3$. Thus, we may assume that G is $\{K_3, K_{2,2}\}$ -free. In this case, we apply Theorem 1.3 to reduce the problem to the case where G has bounded treewidth and then apply Theorem 1.2. This proves Theorem 1.1.

2. DEFINITIONS

Let $G = (V, E)$ be a graph. For $X \subseteq V$, we define *the neighborhood* of X , denoted $N(X)$, to be the set of all vertices in $V \setminus X$ with at least one neighbor in X . If $X = \{x\}$, then we write $N(x)$ for $N(\{x\})$.

An *induced minor* of G is a graph obtained from an induced subgraph of G by repeatedly contracting edges, deleting all arising loops and parallel edges. For a graph H , we say that G *contains* or *has* an induced H -minor if there is an induced minor of G isomorphic to H . The *line graph* of G , denoted $L(G)$, is the graph with vertex set $E(G)$ such that $e, f \in E(G)$ are adjacent in $L(G)$ if and only if e and f share an end in G . We say that a graph G is *t -tidy* if G is K_{t+1} -free and has no induced $K_{t,t}$ -minor.

Let $k \in \mathbb{N}$ and let P be a k -vertex graph that is a path. Then we write $P = p_1 \cdots p_k$ to mean that $V(P) = \{p_1, \dots, p_k\}$ and $E(P) = \{p_i p_{i+1} : i \in \mathbb{N}_{k-1}\}$. We call the vertices p_1 and p_k the *ends* of P and refer to $P \setminus \{p_1, p_k\}$ as the *interior* of P , denoted P^* . For a set of paths \mathcal{P} , we define the *interior of \mathcal{P}* , denoted \mathcal{P}^* , to be the set of interiors of paths in \mathcal{P} , i.e., $\mathcal{P}^* = \{P^* \mid P \in \mathcal{P}\}$. For a banana $B = (\{x, z\}, \mathcal{P})$, we use B^* and \mathcal{P}^* interchangeably.

For vertices $u, v \in P$, we denote by u - P - v the subpath of P from u to v . The *length* of a path is the number of edges in it. It follows that a path P has distinct ends if and only if P has non-zero length, and P has non-empty interior if and only if P has length at least two. Given a graph G , a *path in G* is an induced subgraph of G that is a path.

We define a *directed graph* to be a pair (V, D) where $D \subseteq V \times V$ is the set of directed edges. As before, we do not allow loops or parallel edges. A *tournament* is a directed graph $G = (V, D)$ such that for each pair $u, v \in V$ exactly one of $(u, v), (v, u)$ is in D . A tournament is called *transitive* if $(u, v), (v, w) \in D$ implies that $(u, w) \in D$. The *underlying graph* of a directed graph G is the graph G^- with the same vertex set and an edge uv if and only if $(u, v) \in D$ or $(v, u) \in D$. A well-known fact about tournaments is the following theorem.

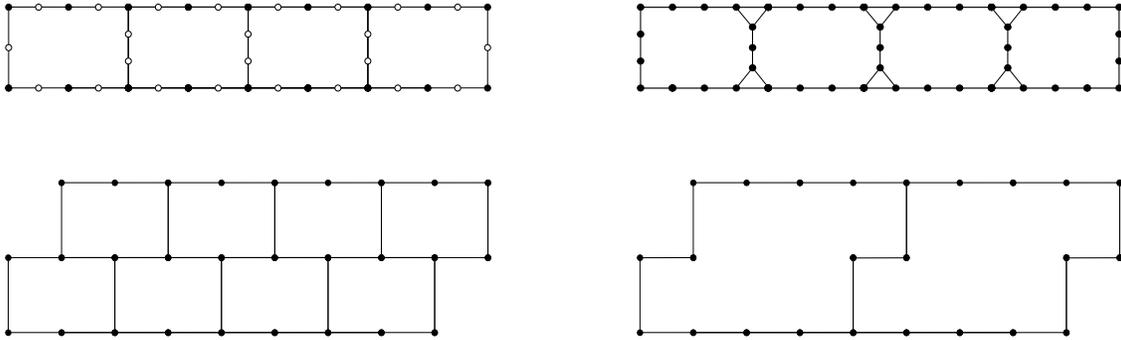


FIGURE 4. A proper subdivision of $W_{1 \times 4}$ (top left), its line graph (top right), the graph $W_{2 \times 4}$ (bottom left), and a proper subdivision of $W_{1 \times 2}$ (bottom right). Note that $W_{2 \times 4}$ contains a proper subdivision of $W_{1 \times 2}$.

Theorem 2.1 (Erdős, Moser [10]). *Each tournament on 2^{n-1} vertices contains a transitive tournament on n vertices.*

We will also use the following classical result.

Theorem 2.2 (Ramsey [11]). *For all $s, t \in \mathbb{N}$, every graph on at least s^t vertices has either a stable set of cardinality s or a clique of cardinality $t + 1$.*

3. SOME GOOD CASES

In this section, we establish some sufficient conditions for $\text{cl}(G)$ to be large.

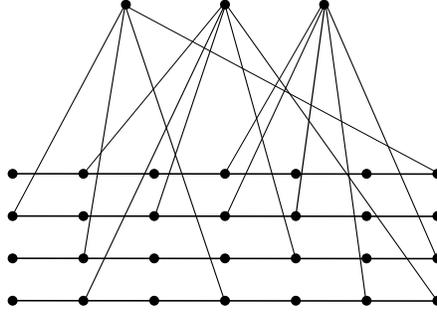
3.1. Removing basic obstructions. For every $r \in \mathbb{N}$, we denote by $W_{r \times r}$ the r -by- r hexagonal grid, also known as the r -by- r wall (see Figure 4). As the first step toward the proof of Theorem 1.3, we prove that certain graphs with large treewidth contain induced cycles of many different lengths. In this subsection, we prove Lemmas 3.3 and 3.4 that show that for every $c \in \mathbb{N}$ there exists $r \in \mathbb{N}$, such that if a graph G contains

- a subdivision of $W_{r \times r}$,
- the line graph of a subdivision of $W_{r \times r}$,
- a proper subdivision of K_r , or
- an induced $K_{r,r}$ -minor,

then $\text{cl}(G) \geq c$. We start by introducing some definitions and Theorem 3.1 from [7].

A *constellation* is a graph \mathfrak{c} in which there is a stable set $S_{\mathfrak{c}}$ such that every component of $\mathfrak{c} \setminus S_{\mathfrak{c}}$ is a path, and each vertex $s \in S_{\mathfrak{c}}$ has at least one neighbor in each component of $\mathfrak{c} \setminus S_{\mathfrak{c}}$. We say that \mathfrak{c} is *ample* if no two vertices in $S_{\mathfrak{c}}$ have a common neighbor. We denote by $\mathcal{L}_{\mathfrak{c}}$ the set of all components $\mathfrak{c} \setminus S_{\mathfrak{c}}$ (each of which is a path), and denote the constellation \mathfrak{c} by the pair $(S_{\mathfrak{c}}, \mathcal{L}_{\mathfrak{c}})$. For $l, s \in \mathbb{N}$, by an (s, l) -constellation we mean a constellation \mathfrak{c} with $|S_{\mathfrak{c}}| = s$ and $|\mathcal{L}_{\mathfrak{c}}| = l$ (see Figure 5).

Theorem 3.1 (Chudnovsky, Hajebi, Spirkl [7]). *For all $l, r, s \in \mathbb{N}$, there is a constant $f_{3.1} = f_{3.1}(l, r, s) \in \mathbb{N}$ such that every graph G with an induced $K_{f_{3.1}, f_{3.1}}$ -minor contains one of the following:*

FIGURE 5. An ample $(3, 4)$ -constellation.

- (a) $K_{r,r}$, a subdivision of $W_{r \times r}$, or the line graph of a subdivision of $W_{r \times r}$;
- (b) an ample (s, l) -constellation.

We will also need the following well-known result.

Theorem 3.2 (Erdős, Szekeres [9]). *For every $r \in \mathbb{N}$, every sequence of distinct real numbers with length at least $r^2 + 1$ contains a monotonically increasing or a monotonically decreasing subsequence of length $r + 1$.*

Now, we are ready to prove Lemmas 3.3 and 3.4.

Lemma 3.3. *For every $c \in \mathbb{N}$, there is a constant $f_{3.3} = f_{3.3}(c)$ such that if a graph G contains*

- (a) a subdivision of $W_{f_{3.3} \times f_{3.3}}$,
- (b) the line graph of a subdivision of $W_{f_{3.3} \times f_{3.3}}$, or
- (c) a proper subdivision of $K_{f_{3.3}}$,

then $\text{cl}(G) \geq c$.

Proof. Define $f_{3.3} = 2c^2$. Suppose that G contains an induced subgraph H isomorphic to a subdivision of $W_{f_{3.3} \times f_{3.3}}$ or a proper subdivision of $K_{f_{3.3}}$. Then, observe that H contains an induced subgraph H' isomorphic to a proper subdivision of $W_{1 \times c^2}$ (see Figure 4). Let H' be formed by the paths P, Q and R_1, \dots, R_{c^2+1} .

By Theorem 3.2, there is a subsequence $R_{i_1}, \dots, R_{i_{c+1}}$ such that the lengths of the paths in this subsequence are either increasing or decreasing. We may assume that the lengths are increasing by reindexing the paths. Then, the induced subgraph H'' formed by the paths P, Q and $R_{i_1}, \dots, R_{i_{c+1}}$ contains at least c induced cycles of strictly increasing lengths. Thus, $\text{cl}(G) \geq c$ as required.

Now, suppose that G contains an induced subgraph H isomorphic to the line graph of a subdivision of $W_{f_{3.3} \times f_{3.3}}$. Observe that H contains an induced subgraph H' isomorphic to the line graph of a proper subdivision of $W_{1 \times c^2}$. By an argument analogous to the one in the previous paragraph, it follows that $\text{cl}(G) \geq c$ as required. ■

Lemma 3.4. *For all $c, t \in \mathbb{N}$, there is a constant $f_{3.4} = f_{3.4}(c, t)$ such that if a $K_{t,t}$ -free graph G has an induced $K_{f_{3.4}, f_{3.4}}$ -minor, then $\text{cl}(G) \geq c$.*

Proof. Define

$$\begin{aligned} r &= \max\{t, f_{3.3}(c)\} \\ l &= (c+1)(c!+1) \\ f_{3.4} &= f_{3.1}(l, r, c+1) \end{aligned}$$

By Theorem 3.1, G contains

- (a) $K_{r,r}$, a subdivision of $W_{r \times r}$, the line graph of a subdivision of $W_{r \times r}$, or
- (b) an ample $(c+1, l)$ -constellation.

Suppose that $\text{cl}(G) < c$. Since G is $K_{t,t}$ -free and $t \leq r$, Lemma 3.3 implies that G contains an ample $(c+1, l)$ -constellation \mathfrak{c} . Let $\mathcal{L}_{\mathfrak{c}} = \{L_1, \dots, L_l\}$. For each $i \in \mathbb{N}_l$, let $L'_i = v_i^1 - \dots - v_i^{k_i}$ with $k_i \in \mathbb{N}$ be a minimal subpath of L_i such that each $s \in S_{\mathfrak{c}}$ has a neighbor in L'_i and let $s(i)$ be a vertex of $S_{\mathfrak{c}}$ such that its only neighbor in L'_i is v_i^1 . By the choice of l , there exists a vertex s_1 and a set $I \subseteq \mathbb{N}_l$ of size $c!+1$ such that $s_1 = s(i)$ for each $i \in I$.

For each $i \in I$ and $s \in S_{\mathfrak{c}}$, let $j = j(i, s) \in \mathbb{N}$ be minimum such that s is adjacent to v_i^j , i.e., $j(i, s)$ is the index of the first neighbor of s in L'_i . Because \mathfrak{c} is ample and $|I| = c!+1$, there are $q, w \in I$ and an ordering s_1, \dots, s_{c+1} of the vertices of $S_{\mathfrak{c}}$ such that $j(q, s_a) < j(q, s_b)$ and $j(w, s_a) < j(w, s_b)$ for $a < b$, i.e., the first neighbors of s_1, \dots, s_{c+1} appear in the same order in both L'_q and L'_w . Finally, observe that $s_1 - v_q^1 - v_q^{j(q, s_1)} - s_1 - v_w^{j(w, s_1)} - v_w^1 - s_1$ for $i = 2, \dots, c+1$ forms c induced cycles of increasing lengths. Thus, $\text{cl}(G) \geq c$ as required. ■

3.2. Handling domes. In this subsection, we prove Lemma 3.6 that shows that if a graph G contains a large enough dome, then it also contains induced cycles of many distinct lengths. As a result, the proof of Theorem 1.3 will be reduced to finding large domes in graphs. We start by introducing a lemma, which follows immediately from Lemma 4.4 in [6].

Lemma 3.5 (Chudnovsky, Hajebi, Spirkl [6]). *For all $d, s, t \in \mathbb{N}$, there is a constant $f_{3.5} = f_{3.5}(d, s, t)$ such that every graph G with an $(f_{3.5}, d)$ -dome contains*

- (a) an aligned $(s, 1)$ -dome; or
- (b) an induced $K_{t,t}$ -minor.

We prove:

Lemma 3.6. *Let G be a graph with no $K_{t,t}$ -induced minor. For all $c, d, t \in \mathbb{N}$, there exists a constant $f_{3.6} = f_{3.6}(c, d, t)$ such that if G contains an $(f_{3.6}, d)$ -dome, then $\text{cl}(G) \geq c$.*

Proof. Define $s = c^2 + 1$ and $f_{3.6} = f_{3.5}(d, s, t)$. By Lemma 3.5, G contains an aligned $(s, 1)$ -dome (S, P) . Let x be the root of S and let $L = \{l_1, \dots, l_s\}$ be its leaf set such that P traverses $N(l_1) \cap P, \dots, N(l_s) \cap P$ in this order. For $i \in \mathbb{N}_s$, let S_i be the $x-l_i$ path in S .

By Theorem 3.2, there is a sequence of $c+1$ paths $S_{i_1}, \dots, S_{i_{c+1}}$ of either increasing or decreasing length. We may assume that the lengths are increasing by reindexing if needed. For $j = 2, \dots, c+1$, let P_{i_j} be the subpath of P such that $l_{i_1} - P_{i_j} - l_{i_j}$ is a path. Note that the paths $P_{i_2}, \dots, P_{i_{c+1}}$ have strictly increasing lengths. Therefore, $x - S_{i_1} - l_{i_1} - P_{i_j} - l_{i_j} - S_{i_j} - x$ for $j = 2, \dots, c+1$ are induced cycles in G of strictly increasing lengths. It follows that $\text{cl}(G) \geq c$ as claimed. ■

4. REDUCING TO A BANANA

In this section, we derive Theorem 1.3 using lemmas from Section 3, Theorem 4.1 stated shortly, and two results from [2] and [6]. We will prove Theorem 4.1 in Sections 5, 6 and 7.

Theorem 4.1. *For all $l, s, t \in \mathbb{N}$, there are constants $f_{4.1} = f_{4.1}(l, s, t)$ and $g_{4.1} = g_{4.1}(t)$ such that for every t -tidy graph G that contains an l -dense $f_{4.1}$ -banana $B_{x,z}$, there is an $(s, g_{4.1})$ -dome with root x in G .*

Let $k, m \in \mathbb{N}$ and let G be a graph. A (k, m) -block B in G is a pair (X, \mathcal{P}) where $X \subseteq V(G)$ and \mathcal{P} is a set of paths with ends in X such that

- $|X| = k$;
- for all $\{x, y\} \subseteq X$, there is a set $\mathcal{P}_{\{x,y\}} \subseteq \mathcal{P}$ of m pairwise internally disjoint x - y -paths;
- $\mathcal{P} = \cup_{\{x,y\} \subseteq X} \mathcal{P}_{\{x,y\}}$.

We say that B is *strong* if for all distinct pairs $\{x, y\}, \{x', y'\}$ of X , we have $V(\mathcal{P}_{\{x,y\}}) \cap V(\mathcal{P}_{\{x',y'\}}) = \{x, y\} \cap \{x', y'\}$; that is, each path $P \in \mathcal{P}_{\{x,y\}}$ is disjoint from each path $P' \in \mathcal{P}_{\{x',y'\}}$, except P and P' may share an end. We call B *anticomplete* if for every pair of vertices $x, y \in X$, the paths in $\mathcal{P}_{\{x,y\}}^*$ are pairwise anticomplete. We will need the following two results about blocks.

Theorem 4.2 (Abrishami, Alecu, Chudnovsky, Hajebi, Spirkl [2]). *Let G be a graph that does not contain $K_{t+1}, K_{t,t}$, a subdivision of $W_{t \times t}$, or the line graph of a subdivision of $W_{t \times t}$. For all $k, l, t \in \mathbb{N}$, there is a constant $f_{4.2} = f_{4.2}(k, l, t) \in \mathbb{N}$ such that if $\text{tw}(G) \geq f_{4.2}$, then G contains a strong (k, l) -block.*

Lemma 4.3 (Chudnovsky, Hajebi, Spirkl [6]). *For all $s, t \in \mathbb{N}$, there are constants $f_{4.3} = f_{4.3}(s, t) \in \mathbb{N}$ and $g_{4.3} = g_{4.3}(s) \in \mathbb{N}$ with the following property. Let G be a K_{t+1} -free graph and let B be a strong anticomplete $(f_{4.3}, g_{4.3})$ -block in G . Then, G contains*

- (a) a proper subdivision of K_s , or
- (b) an induced $K_{s,s}$ -minor.

We are now ready to prove Theorem 1.3, which we restate.

Theorem 1.3. *For all $c, t \in \mathbb{N}$, there is a constant $f_{1.3} = f_{1.3}(c, t) \in \mathbb{N}$ such that for every graph G with $\text{tw}(G) \geq f_{1.3}$ one of the following holds.*

- (a) G contains K_{t+1} or $K_{t,t}$.
- (b) $\text{cl}(G) \geq c$.

Proof. Define

$$t' = \max\{t, f_{3.3}(c), f_{3.4}(c, t)\}$$

$$k = f_{4.3}(t', t')$$

$$l = g_{4.3}(t')$$

$$d = g_{4.1}(t')$$

$$s = f_{3.6}(c, d, t')$$

$$q = f_{4.1}(l, s, t')$$

$$f_{1.3} = f_{4.2}(k, q, t')$$

Let G be a graph with $\text{tw}(G) \geq f_{1.3}$. Suppose that G is $\{K_{t+1}, K_{t,t}\}$ -free and that $\text{cl}(G) < c$. By the choice of t' , Lemmas 3.3 and 3.4 imply that G does not contain

- $K_{t'+1}$,
- $K_{t',t'}$,
- a subdivision of $W_{t' \times t'}$,
- the line graph of a subdivision of $W_{t' \times t'}$,
- a proper subdivision of $K_{t'}$, nor
- an induced $K_{t',t'}$ -minor.

Then, by Theorem 4.2, G has a strong (k, q) -block $B = (X, \mathcal{P})$. By the observations above and Lemma 4.3, G does not contain a strong anticomplete (k, l) -block. Thus, G contains an l -dense q -banana B' . By Theorem 4.1, G contains an (s, d) -dome D . Finally, by Lemma 3.6, it follows that $\text{cl}(G) \geq c$, which is a contradiction. This completes the proof of the theorem. \blacksquare

5. ANALYZING BANANAS

In this section, we analyze the structure of rigid bananas and prove Lemmas 5.2 and 5.3 that will be useful for us in Section 6. We start with a few definitions and Theorem 5.1 from [1].

A vertex v in a graph G is said to be a *branch vertex* if v has degree more than two. By a *caterpillar* we mean a tree C with maximum degree three such that there exists a path P of C where all branch vertices of C belong to P . We call a maximal such path P the *spine* of C . (This definition of a caterpillar is non-standard for two reasons: a caterpillar is often allowed to be of arbitrary maximum degree, and a spine often contains all vertices of degree more than one.) By a *subdivided star* we mean a graph isomorphic to a subdivision of the complete bipartite graph $K_{1,\delta}$ for some $\delta \geq 3$. In other words, a subdivided star is a tree with exactly one branch vertex, which we call its root. For a graph H , a vertex v of H is said to be *simplicial* if $N_H(v)$ is a clique. We denote by $\mathcal{Z}(H)$ the set of all simplicial vertices of H . Note that for every tree T , $\mathcal{Z}(T)$ is the set of all leaves of T . An edge e of a tree T is said to be a *leaf-edge* of T if e is incident with a leaf of T . It follows that if H is the line graph of a tree T , then $\mathcal{Z}(H)$ is the set of all vertices in H corresponding to the leaf-edges of T .

Theorem 5.1 (Abrishami, Alecu, Chudnovsky, Hajebi, Spirkl [1]). *For every integer $h \geq 1$, there exists an integer $f_{5.1} = f_{5.1}(h) \geq 1$ with the following property. Let G be a connected graph with no clique of cardinality h . Let $S \subseteq G$ such that $|S| \geq f_{5.1}$, $G \setminus S$ is connected, and every vertex of S has a neighbor in $G \setminus S$. Then there is a set $S' \subseteq S$ with $|S'| = h$ and an induced subgraph H of $G \setminus S$ for which one of the following holds.*

- (a) H is a path and every vertex of S' has a neighbor in H ; or
- (b) H is a caterpillar, or the line graph of a caterpillar, or a subdivided star. Moreover, every vertex of S' has a unique neighbor in H and $H \cap N(S') = \mathcal{Z}(H)$.

We will now apply this theorem to show how to carve a theta from a rigid banana. First, we do it with the additional assumption that all the neighbors of one of the ends of the banana have degree two.

Lemma 5.2. *Let $q, s, t \in \mathbb{N}$ and let G be a t -tidy graph. There are constants $f_{5.2} = f_{5.2}(q, s, t)$ and $g_{5.2} = g_{5.2}(q)$ such that if G contains a rigid $f_{5.2}$ -banana $B = (\{x, z\}, \mathcal{P})$ such that each vertex in $N(x) \cap V(B)$ has degree two in B , then one of the following holds.*

- (a) $G \setminus \{z\}$ contains a q -theta $(\{x, y\}, \mathcal{Q})$.
- (b) G contains an $(s, g_{5.2})$ -dome with root x .

Proof. Define $g_{5.2} = \max\{q, 3\}$, $h = \max\{q, s, t\}$, $f_{5.2} = f_{5.1}(h)$.

Recall that G is t -tidy and observe that $S = N(x) \cap V(B)$ is a stable set in B . Also, observe that $G' = G[V(B^*) \setminus S]$ is connected (since B is rigid and each vertex in S has degree 1 in $G[V(B^*)]$) and that each vertex in S has a neighbor in G' . Applying Theorem 5.1, we obtain a set $S' \subseteq S$ of size h and an induced subgraph H of G' for which one of the following holds.

- (a) H is a path and every vertex of S' has a neighbor in H ; or
- (b) H is a caterpillar, or the line graph of a caterpillar, or a subdivided star. Moreover, every vertex of S' has a unique neighbor in H and $H \cap N(S') = \mathcal{Z}(H)$.

Suppose that case (a) holds. If some vertex y in H is adjacent to at least q vertices of S' , then x, y and $S' \cap N(y)$ form a q -theta as claimed. Otherwise, $\tilde{G} = G[\{x\} \cup S' \cup V(H)]$ contains an $(s, g_{5.2})$ -dome with root x as claimed.

Thus, we may assume that case (b) holds. Note that $h \geq q$. Suppose that H is a subdivided star with root r . Then, \tilde{G} contains a q -theta $\Theta_{x,r}$ as claimed. Since $V(H) \subseteq V(B^*)$, it's clear that $\Theta_{x,r}$ does not contain z .

Therefore, suppose that H is a caterpillar or the line graph of a caterpillar. Then, observe that each vertex in H has degree at most three. Thus, \tilde{G} contains an $(s, g_{5.2})$ -dome with root x as claimed. ■

Next, we justify the additional assumption of Lemma 5.2.

Lemma 5.3. *Let $q, s, t \in \mathbb{N}$ and let G be a t -tidy graph. There are constants $f_{5.3} = f_{5.3}(q, s, t)$ and $g_{5.3} = g_{5.3}(t)$ such that if G contains a rigid $f_{5.3}$ -banana $B = (\{x, z\}, \mathcal{P})$, then one of the following holds.*

- (a) G contains a rigid q -banana $B' = (\{x, z\}, \mathcal{P}')$ such that each vertex in $N(x) \cap V(B')$ has degree two in $G[V(B')]$.
- (b) G contains an $(s, g_{5.3})$ -dome with root x .

Proof. Define

$$\begin{aligned} d &= g_{5.3} = t^t \\ r &= \binom{s}{d} d \\ w &= 2^{s+r-1} \end{aligned}$$

$$f = f_{5.3} = q^{w-1}$$

Suppose that G does not contain an $(s, g_{5.3})$ -dome with root x . Let $\mathcal{P} = \{P_1, \dots, P_f\}$ and let $x_i = N(x) \cap P_i$ for $i \in \mathbb{N}_f$. Define a directed graph H with vertex set \mathbb{N}_f where $(i, j) \in E(H)$ if and only if x_i has a neighbor in P_j^* . First, we claim that

(1) H^- does not contain a clique of size w .

Suppose that C is a clique in H^- of size w . Then, $H[C]$ contains a tournament on w vertices. By Theorem 2.1, H contains a transitive tournament on $s + r$ vertices, say $\{a_1, \dots, a_s, b_1, \dots, b_r\}$ where (a_i, b_j) is an edge for all $i \in \mathbb{N}_s, j \in \mathbb{N}_r$. Recall that this means that x_{a_i} has a neighbor in $P_{b_j}^*$. Suppose that there exists a $j \in \mathbb{N}_r$ such that each vertex of $P_{b_j}^*$ has at most d neighbors in $X = \{x_{a_1}, \dots, x_{a_s}\}$. Note that $G[\{x\} \cup X]$ is a star with root x and that each vertex in X has a neighbor in $P_{b_j}^*$. Thus, G contains an $(s, g_{5.3})$ -dome with root x , which is a contradiction.

Therefore, for every $j \in \mathbb{N}_r$, $P_{b_j}^*$ contains a vertex v_j that has at least d neighbors in X . Since $r = \binom{s}{d}d$, there exist sets $X' \subseteq X, V' \subseteq \{v_1, \dots, v_r\}$ such that $|X'| = |V'| = d$ and xv is an edge for all $x \in X', v \in V'$. Finally, recall that G is t -tidy. Thus, both X' and V' contain stable sets X'', V'' of size t , respectively. However, $G[X'' \cup V'']$ is isomorphic to $K_{t,t}$, which is a contradiction. This completes the proof of (1).

Theorem 2.2 and (1) imply that H^- contains a stable set S of size q , say $S = \{s_1, \dots, s_q\}$. Let $B' = (\{x, z\}, \mathcal{P}')$ where $\mathcal{P}' = \{P_{s_1}, \dots, P_{s_q}\}$. Note that for all distinct $i, j \in \mathbb{N}_q$, x_{s_i} does not have a neighbor in $P_{s_j}^*$. Thus, x_{s_i} has degree two in B' as required. ■

6. FROM A BANANA TO A KITE

In this section, we prove Lemma 6.1 that shows how to obtain a kite from a rigid dense banana and then show how to make this kite clean in Theorem 6.2.

Lemma 6.1. *For all $k, l, q, s, t \in \mathbb{N}$, there are constants $f_{6.1} = f_{6.1}(k, l, q, s, t)$ and $g_{6.1} = g_{6.1}(q, t)$ with the following property. Let G be a t -tidy graph that contains a rigid l -dense $f_{6.1}$ -banana $\hat{B}_{x,z}$. Then, G contains an (x, z) -tailed (k, q) -kite or an $(s, g_{6.1})$ -dome with root x .*

Proof. Define

$$\begin{aligned} p &= f_{5.2}(2q + l, s, t) \\ f_{6.1} &= f_{5.3}(pk, s, t) \\ g_{6.1} &= \max\{g_{5.2}(q), g_{5.3}(t)\} \end{aligned}$$

Suppose that G does not contain an $(s, g_{6.1})$ -dome with root x . Lemma 5.3 implies that G contains a rigid pk -banana $B = (\{x, z\}, \mathcal{P})$ such that every vertex in $N(x) \cap V(B)$ has degree two in B . Partition \mathcal{P} into k sets $\mathcal{P}_1, \dots, \mathcal{P}_k$ of size p each. For $i \in \mathbb{N}_k$, let $B_i = (\{x, z\}, \mathcal{P}_i)$ be the resulting banana. Apply Lemma 5.2 to each B_i to obtain a $(2q + l)$ -theta $\hat{\Theta}_i = (\{x, y_i\}, \hat{\mathcal{Q}}_i)$. Note that z can have neighbors in at most l paths of $(\hat{\mathcal{Q}}_i)^*$ as \hat{B} is

l -dense. Thus, we can remove from $(\hat{\mathcal{Q}}_i)^*$ the paths that are not anticomplete to z to obtain a $2q$ -theta $\Theta_i = (\{x, y_i\}, \mathcal{Q}_i)$ such that z is anticomplete to $(\mathcal{Q}_i)^*$. We claim that

(2) For $i \in \mathbb{N}_k$, B_i contains an (x, z) -tailed q -theta.

Let P_i be the path in \mathcal{P}_i that contains y_i . Let v_i be the first vertex of P_i with a neighbor in Θ_i^* , traversing P_i from z to x . Such a vertex v_i exists because y_i is one possible candidate. Note that $v_i \neq z$ since z is anticomplete to $(\mathcal{Q}_i)^*$. Suppose that v_i has a neighbor in at least q paths of Θ_i^* , say $Q_{i,1}, \dots, Q_{i,q}$. For $j \in \mathbb{N}_q$, let $Q'_{i,j}$ be a path from x to v_i with interior in $Q_{i,j}$. Then, $x-Q'_{i,j}-v_i-P_i-z$ is an x - z path for every $j \in \mathbb{N}_q$ and thus they form an (x, z) -tailed q -theta.

Thus, we may assume that there is a set of q paths $\mathcal{Q}'_i \subseteq \mathcal{Q}_i$ such that v_i is anticomplete to $(\mathcal{Q}'_i)^*$. It follows that $v_i \neq y_i$. Let Q_i be some path in \mathcal{Q}_i such that v_i has a neighbor in Q_i^* . Such a path exists by the definition of v_i . Let Q'_i be a path from y_i to v_i with interior in Q_i . Note that Q'_i is anticomplete to x (since each vertex in $N(x)$ has degree two in B_i and one of its neighbors is x). Also, note that $Q'_i \setminus \{y_i\}$ is anticomplete to $(\mathcal{Q}'_i)^*$ (since paths in \mathcal{Q}^* are anticomplete as they are part of a theta). Thus, \mathcal{Q}'_i together with the tail $y_i-Q'_i-v_i-P_i-z$ form an (x, z) -tailed q -theta as required. ■

We are now ready to prove the main result of this section.

Theorem 6.2. *For all $k, l, q, s, t \in \mathbb{N}$, there are constants $f_{6.2} = f_{6.2}(k, l, q, s, t)$ and $g_{6.2} = g_{6.2}(q, t)$ with the following property. Let G be a t -tidy graph that contains a rigid l -dense $f_{6.2}$ -banana $\hat{B}_{x,z}$. Then, G contains a clean (x, z) -tailed (k, q) -kite or an $(s, g_{6.2})$ -dome with root x .*

Proof. Define

$$\begin{aligned} \phi' &= q + ks \\ \theta &= \binom{\phi'}{t} t \\ \gamma &= \theta^t \\ \psi &= 2^\gamma \\ \kappa' &= k^{\psi-1} \\ \kappa &= (\kappa')^t \\ \phi &= 3^{\kappa^2} (\phi')^{2\kappa^2} \\ f_{6.2} &= f_{6.1}(\kappa, l, \phi, s, t) \\ g_{6.2} &= \max\{t, g_{6.1}(q, t)\} \end{aligned}$$

Suppose that G does not contain an $(s, g_{6.2})$ -dome with root x . Lemma 6.1 implies that G contains an (x, z) -tailed (κ, ϕ) -kite $\mathcal{K} = \{(\Theta_{x,y_i}, P_i) \mid i \in \mathbb{N}_\kappa\}$. We will clean up the kite \mathcal{K} in three steps. First, we will make Θ_{x,y_i}^* and Θ_{x,y_j}^* pairwise anticomplete for $i \neq j$. Second, we will make the set $\{y_1, \dots, y_\kappa\}$ stable. Third, we will make Θ_{x,y_i}^* and P_j^* anticomplete for

$i \neq j$. Note that this implies that each vertex of Θ_{x,y_i}^* has degree two in the obtained kite. Therefore, our first claim is

(3) G contains an (x, z) -tailed (κ, ϕ') -kite $\mathcal{K}' = \{(\Theta'_{x,y_i}, P_i) \mid i \in \mathbb{N}_\kappa\}$ such that $(\Theta'_{x,y_i})^*$ and $(\Theta'_{x,y_j})^*$ are pairwise anticomplete for $i \neq j$.

The proof is exactly the same as the proof of Claim (1) in Lemma 3.6 in [6], and we will not repeat it here.

Let \mathcal{K}' be as in (3). Since G is t -tidy, Theorem 2.2 implies that there is a set $I \subseteq \mathbb{N}_\kappa$ of size κ' such that $\{y_i \mid i \in I\}$ is a stable set. By renumbering if necessary, we may assume that $I = \mathbb{N}_{\kappa'}$. Let $\mathcal{K}'' = \{(\Theta'_{x,y_i}, P_i) \mid i \in \mathbb{N}_{\kappa'}\}$.

We are now ready for the third step. Consider a directed graph D with vertex set $\mathbb{N}_{\kappa'}$ and an edge from i to j if and only if P_i has neighbors in at least s paths of $(\Theta'_{x,y_j})^*$. We claim:

(4) D^- does not contain a clique of size ψ .

Suppose that C is a clique of size ψ in D^- . Then, $D[C]$ contains a tournament of size ψ . By Theorem 2.1, $D[C]$ contains a transitive tournament T of size $\gamma + 1$. Therefore, there exists a set of γ paths $\mathcal{P} = \{P_{i_1}, \dots, P_{i_\gamma}\}$ and a theta Θ'_{x,y_j} such that each path in \mathcal{P} has a neighbor in at least s paths of $(\Theta'_{x,y_j})^*$.

Suppose that there exists a path $P_i \in \mathcal{P}$ such that each vertex v of P_i has neighbors in at most t paths of $(\Theta'_{x,y_j})^*$. Then, $G[x \cup V((\Theta'_{x,y_j})^*) \cup V(P_i)]$ contains an $(s, g_{6.2})$ -dome with root x , which is a contradiction.

Thus, for every $m \in \mathbb{N}_\gamma$, there exists a vertex $p_{i_m} \in P_{i_m}$ such that p_{i_m} has neighbors in at least t paths of $(\Theta'_{x,y_j})^*$. Recall that G is t -tidy. By Theorem 2.2, there exists a set of θ vertices $p_{j_1}, \dots, p_{j_\theta}$ that form a stable set. Since $\theta = \binom{\phi'}{t} t$, there exist t vertices p_{k_1}, \dots, p_{k_t} that all have neighbors in the same t paths Q_1, \dots, Q_t of $(\Theta'_{x,y_i})^*$. However, contracting the paths Q_1, \dots, Q_t , yields an induced $K_{t,t}$ -minor, which is a contradiction as G is t -tidy. This completes the proof of (4).

By (4) and Theorem 2.2, D^- contains a stable set S of size k . This implies that there exists a set of k (x, z) -tailed thetas $\{(\Theta'_{x,y_j}, P_j) \mid j \in S\}$ such that, for $i \neq j$, the path P_i has neighbors in at most s paths of Θ'_{x,y_j} . For each pair P_i, Θ'_{x,y_j} remove from $(\Theta'_{x,y_j})^*$ the paths that have a neighbor in P_i . This will remove at most ks paths from each Θ'_{x,y_j} resulting in a collection of q -thetas $\Theta''_{x,y_{j_1}}, \dots, \Theta''_{x,y_{j_k}}$ and their corresponding tails P_{j_1}, \dots, P_{j_k} that together form a clean (k, q) -kite as claimed. This concludes the proof. ■

7. FROM KITES TO A DOME

We define a *binary tree* to be a rooted tree T such that each node $t \in V(T)$ has at most two children. A *full binary tree of radius r* is a binary tree in which every vertex that is not a leaf has exactly two children, and every leaf is at distance exactly r from the root. Let T_r denote the full binary tree of radius r . For $i \in \mathbb{N}_r$, we define the *i -th level set of T_r* , denoted

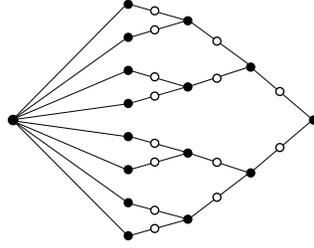


FIGURE 6. A 3-fleet.

by $L_i(T_r)$, as the set of vertices at distance i from the root of T_r . Note that $|L_i(T_r)| = 2^i$ and that $L_r(T_r)$ is the set of leaves of T_r . For a subdivision T'_r of T_r , we define $L_i(T'_r)$ to be $L_i(T_r)$. We define an r -fleet F in a graph G to be a pair consisting of a vertex x , called the anchor of the fleet, and a subdivision T'_r of T_r , such that $N(x) \cap V(T'_r) = L_r(T'_r)$ (see Figure 6). As with other structures, we will refer to a fleet and the graph induced by its vertex set interchangeably.

In this section, we complete the proof of Theorem 4.1. The proof strategy is to repeatedly apply Theorem 6.2 to show that G contains either an $(s, g_{4.1})$ -dome or an s -fleet F . However, as the next lemma shows, F also contains an $(s, 1)$ -dome. This completes the proof.

Lemma 7.1. *Let F be an s -fleet in a graph G with anchor x . Then, G contains an $(s, 1)$ -dome with root x .*

Proof. Let T'_s be the subdivided binary tree with root r in F . Let P be a path from r to a vertex in $L_s(T'_s)$ and let L be the set of vertices of degree one in $G[V(F) \setminus V(P)]$. Note that $|L| = s$ and that each vertex in L has exactly one neighbor in P , and each vertex of P is adjacent to at most one element of L . Finally, observe that $G[V(F) \setminus V(P)]$ contains a subdivided star S with root x and leaf set L . Thus, (S, P) is an $(s, 1)$ -dome with root x in G as required. ■

Theorem 4.1. *For all $l, s, t \in \mathbb{N}$, there are constants $f_{4.1} = f_{4.1}(l, s, t)$ and $g_{4.1} = g_{4.1}(t)$ such that for every t -tidy graph G that contains an l -dense $f_{4.1}$ -banana $B_{x,z}$, there is an $(s, g_{4.1})$ -dome with root x in G .*

Proof. Define

$$\begin{aligned} k_1 &= f_{6.2}(1, l, 2, s, t) \\ k'_1 &= l^{k_1-1} \\ k_i &= f_{6.2}(k'_{i-1}, l, 2, s, t) \text{ for } i = 2, \dots, s \\ k'_i &= l^{k_i-1} \text{ for } i = 2, \dots, s \\ f_{4.1} &= k'_s \\ g_{4.1} &= g_{6.2}(2, t) \end{aligned}$$

Suppose that G does not contain an $(s, g_{4.1})$ -dome with root x ; we will show that G contains an s -fleet.

We start with the following.

(5) Let $l, q \in \mathbb{N}$ and let $B_{x,z} = (\{x, z\}, \mathcal{P})$ be an l -dense banana with $|\mathcal{P}| \geq l^{q-1}$. Then, $B_{x,z}$ contains a rigid l -dense q -banana $B'_{x,z}$.

Define a graph H with vertex set \mathcal{P} and an edge between paths $P, Q \in \mathcal{P}$ if and only if P^* is not anticomplete to Q^* . Since $B_{x,z}$ is l -dense, H does not contain a stable set of size l . Therefore, Theorem 2.2 implies that H contains a clique \mathcal{P}' of size q . Now, $B'_{x,z} = (\{x, z\}, \mathcal{P}')$ is a rigid banana, as required. This proves (5).

Next, we prove:

(6) Let $B_{x,z}$ be an l -dense k'_s -banana that is a t -tidy graph with no $(s, g_{4.1})$ -dome with root x . Then, $B_{x,z}$ contains an s -fleet with anchor x .

We prove (6) by induction on s . We start with the base case. Let $B_{x,z} = (\{x, z\}, \mathcal{P})$ be an l -dense t -tidy k'_1 -banana with no $(s, g_{4.1})$ -dome with root x . By (5), $B_{x,z}$ contains a rigid k_1 -banana $B'_{x,z}$. By Theorem 6.2, $B'_{x,z}$ contains a clean (x, z) -tailed $(1, 2)$ -kite \mathcal{K} . Let $\Theta_{x,y}$ be the only theta of \mathcal{K} (in fact, $\Theta_{x,y}$ is an induced cycle). Then, $\Theta_{x,y}$ is a 1-fleet with anchor x . This completes the base case.

Now, suppose that we have proved (6) for every $s' < s$, and let $B_{x,z}$ be an l -dense t -tidy k'_s -banana with no $(s, g_{4.1})$ -dome with root x . By (5), $B_{x,z}$ contains a rigid l -dense k_s -banana $B'_{x,z}$. By Theorem 6.2, $B'_{x,z}$ contains a clean (x, z) -tailed $(k'_{s-1}, 2)$ -kite \mathcal{K} . Let $B''_{x,z}$ be obtained from \mathcal{K} by replacing each theta Θ_{x,y_i} of \mathcal{K} by an edge xy_i .

Since $B''_{x,z}$ is an induced minor of $B'_{x,z}$ obtained from \mathcal{K} by contracting only edges with at least one end of degree two in \mathcal{K} , and since $\{y_1, \dots, y_{k'_{s-1}}\}$ is a stable set, it follows that $B''_{x,z}$ is t -tidy, l -dense and does not contain an $(s, g_{4.1})$ -dome with root x .

Inductively, $B''_{x,z}$ contains an $(s-1)$ -fleet F' with anchor x . Renumbering if necessary, we may assume that $y_1, \dots, y_{2^{s-1}}$ are the neighbors of x in F' . Replacing each of the edges xy_i in F' by the cycle Θ_{x,y_i} , we obtain an s -fleet F with anchor x , as claimed. This completes the proof of (6).

Now, let $B_{x,z}$ be as in the statement of the theorem. By (6), $B_{x,z}$ contains an s -fleet F with root x . By Lemma 7.1, F contains an $(s, 1)$ -dome with root x , which is a contradiction. This concludes the proof. ■

8. DETECTING INDUCED CYCLES

In this section, we prove Theorem 1.2.

Throughout this and the next section, we will be using the notation $G = (V, E)$ with $|V| = n$ for $n \in \mathbb{N}$. For a tree decomposition (T, χ) where T is a binary tree, and a node $t \in V(T)$ whose parent is $p \in V(T)$, we define the *adhesion* of t , denoted $A(t)$, to be $\chi(t) \cap \chi(p)$. The adhesion of the root node r is the empty set.

Define a *descendant of* to be the reflexive and transitive relation on the nodes of T such that the children of a node t are its descendants. Thus, t is a descendant of itself, and all of its children and their descendants are also descendants of t . With this, let $d(t)$ be the set of descendants of t in T . Additionally, let G_t be the graph $G[\cup_{v \in d(t)} \chi(v)]$. We will need the following theorem:

Theorem 8.1 (Bodlaender, Kloks [4, 5]). *Let $k \in \mathbb{N}$ be a constant. For a graph G , there exists an algorithm with running time $O(n)$ that decides whether $\text{tw}(G) \leq k$ and if so outputs a tree decomposition (T, χ) of G of width at most k such that T is a binary tree with $O(n)$ nodes.*

Let $t \in T$ be a node and let $Q \subseteq \chi(t)$ be an ordered set $\{q_1, \dots, q_l\}$ such that $N(q_i) \cap Q \subseteq \{q_{i-1}, q_{i+1}\}$ for $2 \leq i \leq l-1$. Let $R \subseteq \{(q_1, q_2), \dots, (q_{l-1}, q_l), (q_l, q_1)\}$ such that if $q_i q_{(i \bmod l)+1} \in E(G)$, then $(q_i, q_{(i \bmod l)+1}) \in R$. Let $X \subseteq A(t) \setminus Q$. We define a *path system* $S_{Q,R,X}$ in G_t to be a set of paths $\{P_r \mid r \in R\}$ in G_t such that for each $r \in R$,

- P_r has ends $q_i, q_{(i \bmod l)+1}$ where $r = (q_i, q_{(i \bmod l)+1})$,
- P_r^* is disjoint from $P_{r'}$ for $r' \neq r$,
- $P_r^* \subseteq V(G_t) \setminus (\chi(t) \cup N(X))$, and
- each vertex of P_r^* has degree two in $V(S_{Q,R,X})$.

We will treat $S_{Q,R,X}$ and $G[V(S_{Q,R,X})]$ interchangeably. Note that a path system is either a cycle or a graph in which every component is a path. The *length* of a path system $S_{Q,R,X}$ is defined as $|E(S_{Q,R,X})|$. The *length variation* of Q, R, X in G_t , denoted by $\text{lv}(Q, R, X, G_t)$, is the set of all possible lengths of a path system $S_{Q,R,X}$ in G_t , i.e.,

$$\text{lv}(Q, R, X, G_t) = \{l \in \mathbb{N} \mid \text{there exists a path system } S_{Q,R,X} \text{ in } G_t \text{ of length } l\}.$$

For $c \in \mathbb{N}$, we define the *c-bounded length variation* of Q, R, X in G_t , denoted by $\text{lv}_c(Q, R, X, G_t)$, to be an arbitrary subset of $\text{lv}(Q, R, X, G_t)$ of size $\min\{c, |\text{lv}(Q, R, X, G_t)|\}$.

We are now ready to present the first algorithm.

Theorem 1.2. *Let $c, k \in \mathbb{N}$ be constants and let G be a graph with $\text{tw}(G) \leq k$. There exists an algorithm with running time $O(|V(G)|)$ that decides whether $\text{cl}(G) \geq c$ and outputs a list of c induced cycles of distinct lengths if they exist.*

Proof. First, we describe an algorithm that outputs a set \mathcal{C} of size $\min\{c, \text{cl}(G)\}$ such that for each $l \in \mathcal{C}$, G contains an induced cycle of length l . Later, we will describe how to add backtracking to this algorithm to output not only the lengths but the actual induced cycles.

Using the algorithm from Theorem 8.1, find a tree decomposition (T, χ) of width at most k . Assume that each node $t \in V(T)$ that is not a leaf has two children where if needed let $\chi(t') = \emptyset$ for the missing child t' . This increases the number of nodes by at most a factor of two, so the total number of nodes is still $O(n)$.

We will solve the problem via dynamic programming over T . We will maintain a set \mathcal{C} (initialized to \emptyset) that contains lengths of induced cycles in G . Once the size of \mathcal{C} reaches c , we simply output it. Additionally, for each node $t \in V(T)$, we will maintain a table M_t indexed by Q, R, X such that each entry $M_t[Q, R, X]$ is a set of integers with the property that if $l \in M_t[Q, R, X]$, then there exists a path system $S_{Q,R,X}$ in G_t of length l , and

moreover $|M_t[Q, R, X]| \leq c$. By the end of the run of the algorithm, $M_t[Q, R, X]$ will contain $\text{lv}_c(Q, R, X, G_t)$.

Let $t \in V(T)$ be a node that is not a leaf and let t_1, t_2 be the two children of t . Intuitively, our algorithm does the following: to compute an entry $M_t[Q, R, X]$, it guesses which vertices of $\chi(t_1) \setminus A(t_1)$ and $\chi(t_2) \setminus A(t_2)$ are needed to connect the pairs in R and in what order they appear on the corresponding paths, and then queries appropriate path systems in M_{t_1} and M_{t_2} . While doing the lookups in M_{t_1} and M_{t_2} , we specify the sets $X_1 \subseteq A(t_1), X_2 \subseteq A(t_2)$ of vertices that need to be anticomplete to the interiors of the paths of the path systems found in G_{t_1}, G_{t_2} to ensure correctness. We then also check if $S_{Q,R,X}$ forms an induced cycle in G_t and record its length.

We process the nodes bottom up, from the leaves to the root, processing each node once all of its children have been processed. After all the nodes have been processed, we either output “ $\text{cl}(G) < c$ ” if the size of \mathcal{C} is smaller than c , or otherwise we output \mathcal{C} . Let us now describe how to process each node. We will prove correctness later.

Base Case:

We start by processing the leaves of T . Let $t \in V(T)$ be a leaf node in T . For each ordered set $Q = \{q_1, \dots, q_l\} \subseteq \chi(t)$ such that $N(q_i) \cap Q \subseteq \{q_{i-1}, q_{i+1}\}$ for $2 \leq i \leq l-1$, let $R = \{(q_i, q_{(i \bmod l)+1}) \mid q_i q_{(i \bmod l)+1} \in E(G)\}$. For all sets $X \subseteq A(t) \setminus Q$, we add $|E(Q)|$ to $M_t[Q, R, X]$. We then check if $G[Q]$ is an induced cycle, and if so, add its length to \mathcal{C} .

General Case:

Let $t \in V(T)$ be a node such that all the descendants of t have already been processed. Denote the two children of t by t_1 and t_2 . Our goal is to compute the table M_t and then query it to record the lengths of all induced cycles in G_t that have not yet been recorded while processing G_{t_1} and G_{t_2} .

Iterate over

- all ordered sets $Q = \{q_1, \dots, q_l\} \subseteq \chi(t)$ such that $N(q_i) \cap Q \subseteq \{q_{i-1}, q_{i+1}\}$ for $2 \leq i \leq l-1$,
- all sets $R \subseteq \{(q_1, q_2), \dots, (q_{l-1}, q_l), (q_l, q_1)\}$ such that if $q_i q_{(i \bmod l)+1} \in E(G)$, then $(q_i, q_{(i \bmod l)+1}) \in R$, and
- all sets $X \subseteq A(t) \setminus Q$.

For each such triple Q, R, X , let $R' = R \setminus \{(q_i, q_{(i \bmod l)+1}) \mid q_i q_{(i \bmod l)+1} \in E(G)\} = \{r_1, \dots, r_{|R'|}\}$, i.e., R' contains the pairs of R that are not yet connected. Iterate through all vectors I in $\{1, 2\}^{|R'|}$ such that if $I_j = i$, then $\{q, q'\} \subseteq A(t_i)$ where $r_j = (q, q')$. Intuitively, the vector I indicates whether a pair $r \in R'$ should be connected by a path in G_{t_1} or G_{t_2} . For each such vector, iterate through all $|R'|$ -tuples $(V_1, \dots, V_{|R'|})$ such that

- each V_j is an ordered subset of $\chi(t_i) \setminus (A(t_i) \cup N(X))$ where $i = I_j$,
- V_i and V_j are disjoint and anticomplete for $i \neq j$, and
- each V_j is anticomplete to $Q \setminus \{q, q'\}$ where $r_j = (q, q')$.

Each set V_j contains the vertices of $\chi(t_{I_j})$ used to connect the pair r_j .

For each $q \in Q$, if $r_j = (q, q')$, then define $Z(q)$ to be a list of elements of V_j in the same order as they appear in V_j , and otherwise define $Z(q)$ to be an empty list. For $i \in \{1, 2\}$, define

- $R'_i = \{r_j \in R' \mid I_j = i\}$ (these are the pairs of R' assigned to be connected through G_{t_i}),
- $Q'_i = \{q \in Q \mid (q, q') \in R'_i \text{ or } (q', q) \in R'_i \text{ for some } q' \in Q\} = \{q_{p_1}, \dots, q_{p_i}\}$ where $p_1 < p_2 < \dots < p_i$ (these are the vertices of R'_i in the same order as they appear in Q),
- $Q_i = \{q_{p_1}, Z(q_{p_1}), q_{p_2}, Z(q_{p_2}), \dots, q_{p_i}, Z(q_{p_i})\} = \{q_1^i, \dots, q_{m_i}^i\}$ to be an ordered set,
- $R_i = \{(q_1^i, q_2^i), \dots, (q_{m_i-1}^i, q_{m_i}^i), (q_{m_i}^i, q_1^i)\} \setminus \{(q, q') \mid q, q' \in Q'_i \text{ and } qq' \notin E(G)\}$, and
- $X_i = (A(t_i) \cap (Q \cup X)) \setminus Q_i$.

With Q_i, R_i, X_i defined, query $L_i = M_{t_i}[Q_i, R_i, X_i]$ for $i = 1, 2$ (if $\chi(t_i) = \emptyset$, then $L_i = \emptyset$). Let $s = |E(Q) \setminus (E(Q_1) \cup E(Q_2))|$. For each pair $l_1 \in L_1, l_2 \in L_2$, add $s + l_1 + l_2$ to $M_t[Q, R, X]$. If $R = \{(q_1, q_2), \dots, (q_{l-1}, q_l), (q_l, q_1)\}$, then also add $s + l_1 + l_2$ to \mathcal{C} . This completes the description of the algorithm.

Running time:

Per each bag, the number of sets Q, R, X , vectors I , and $|R'|$ -tuples $(V_1, \dots, V_{|R'|})$ is bounded by $O(1)$ as the size of each bag is at most k (which is a constant). Forming the sets $R_1, R_2, Q_1, Q_2, X_1, X_2$ and checking that the set Q and the tuple $(V_1, \dots, V_{|R'|})$ are valid can be done in $O(k^2) = O(1)$ by looping over an appropriate submatrix of the adjacency matrix of G . Since each queried set in the tables M_{t_1}, M_{t_2} contains at most c elements, looping through it can also be done in $O(1)$. Thus, there exists a function $f : \mathbb{N} \rightarrow \mathbb{N}$ such that the running time per node is $O(f(k, c)) = O(1)$. The running time of the algorithm from Theorem 8.1 is $O(n)$, and since it returns a tree decomposition with $O(n)$ nodes, the total running time of our algorithm is $O(n)$.

Proof of correctness:

Consider an unrestricted version of our algorithm so as to not restrict the size of each entry in the tables M_t and the set \mathcal{C} to at most c . If the unrestricted version outputs a correct list of all induced cycle lengths of G , then to decide whether $\text{cl}(G) \geq c$, it is enough to detect only c path systems of different lengths for each triple Q, R, X as all such path systems are interchangeable. Thus, for the rest of the proof, we may work with the unrestricted version of our algorithm.

We prove that the algorithm produces a correct output by combining claims (7), (8), (9), and (10) presented below. We will prove claims (7) and (8) by induction on the size of the tree T .

(7) Let $t \in V(T)$, $Q = \{q_1, \dots, q_l\}$, $R \subseteq \{(q_1, q_2), \dots, (q_{l-1}, q_l), (q_l, q_1)\}$, and $X \subseteq A(t) \setminus V(Q)$. If G_t contains a path system $S_{Q,R,X}$, then $M_t[Q, R, X]$ contains the length of $S_{Q,R,X}$.

Base case: Note that in the base case when T is just one node, $R = \{(q_i, q_{(i \bmod l)+1}) \mid q_i q_{(i \bmod l)+1} \in E(G)\}$. Thus, the length of $S_{Q,R,X}$ is $|E(Q)|$, which will be recorded by the algorithm.

Inductive step: At some point, the algorithm will consider the triple Q, R, X as it iterates over all such triples. Since $S_{Q,R,X}$ is a path system, the set Q fulfills the condition that $N(q_i) \cap Q \subseteq \{q_{i-1}, q_{i+1}\}$ for $2 \leq i \leq l-1$, and the set R fulfills the condition that if $q_i q_{(i \bmod l)+1} \in E(G)$, then $(q_i, q_{(i \bmod l)+1}) \in R$. Thus, these checks will succeed. Let R' be defined as in the algorithm description. By the definition of a path system, it follows that for each $r_i = (q, q') \in R'$, the vertices q and q' are connected by a path P_i in either G_{t_1} or G_{t_2} .

Eventually, our algorithm will consider a vector I and an $|R'|$ -tuple $(V_1, \dots, V_{|R'|})$ such that $V_j = P_j \cap (\chi(t_i) \setminus (A(t_i) \cup N(X)))$ where $I_j = i$ for each $j \in \mathbb{N}_{|R'|}$. It is also clear that

- V_i and V_j are disjoint and anticomplete for $i \neq j$, and
- that each V_j is anticomplete to $Q \setminus r_j$.

Now, observe that the path systems found for Q_1, R_1, X_1 and Q_2, R_2, X_2 together with the set Q form a path system for Q, R, X . Inductively, the entries $M_{t_1}[Q_1, R_1, X_1]$ and $M_{t_2}[Q_2, R_2, X_2]$ have been computed correctly. Therefore, they contain the lengths l_1, l_2 of path systems $S_{Q_1, R_1, X_1}, S_{Q_2, R_2, X_2}$ that together with Q form a path system of the same length as $S_{Q,R,X}$. This completes the proof of (7).

(8) Let $t \in V(T)$, $Q = \{q_1, \dots, q_l\}$, $R \subseteq \{(q_1, q_2), \dots, (q_{l-1}, q_l), (q_l, q_1)\}$, and $X \subseteq A(t) \setminus V(Q)$. If $m \in M_t[Q, R, X]$, then there exists a path system $S_{Q,R,X}$ in G_t of length m .

Base case: In the base case when T is just one node, when m is added to $M_t[Q, R, X]$ it holds that $m = |E(Q)|$ and that Q forms a path system. Thus, the claim holds.

Inductive step: Consider the first time that m is added to $M_t[Q, R, X]$. Let $Q_1, Q_2, R_1, R_2, X_1, X_2$ and l_1, l_2 be defined as in the algorithm description. Inductively, the entries $M_{t_1}[Q_1, R_1, X_1]$ and $M_{t_2}[Q_2, R_2, X_2]$ have been computed correctly. Therefore, there exist path systems $S_{Q_1, R_1, X_1}, S_{Q_2, R_2, X_2}$ in G_{t_1}, G_{t_2} of lengths l_1, l_2 respectively.

By the definition of the sets $Q_1, Q_2, R_1, R_2, X_1, X_2$, we have that for each pair $(q, q') \in R$, either $qq' \in E(G)$ or q and q' are connected by a path in one of $S_{Q_1, R_1, X_1}, S_{Q_2, R_2, X_2}$. By the conditions imposed on the $|R'|$ -tuple $(V_1, \dots, V_{|R'|})$, it follows that for each $i \in |R'|$, each vertex of V_i has degree two in $Q \cup S_{Q_1, R_1, X_1} \cup S_{Q_2, R_2, X_2}$ and is anticomplete to $Q \setminus r_i$. Therefore, $Q \cup S_{Q_1, R_1, X_1} \cup S_{Q_2, R_2, X_2}$ forms a path system $S_{Q,R,X}$ in G_t of length m as required. This completes the proof of (8).

(9) If G contains an induced cycle C , then \mathcal{C} contains $|C|$.

Recall that we process nodes of T bottom up. Consider the first node $t \in V(T)$ such that C is contained in G_t . Note that $|C \cap \chi(t)| \geq 2$ as otherwise C is fully contained in either G_{t_1}

or G_{t_2} , both of which have been processed before t . Suppose that $C \cap \chi(t) = \{q_1, \dots, q_l\}$ such that q_1, \dots, q_l appear in C in this order. Note that by the definition of a tree decomposition, each pair $q_i, q_{(i \bmod l)+1}$ is connected by either an edge or a path in G_{t_1} or G_{t_2} . Then, C forms a path system $S_{Q,R,X}$ with $R = \{(q_1, q_2), \dots, (q_{l-1}, q_l), (q_l, q_1)\}$ and $X = \emptyset$. By (7), $M_t[Q, R, X]$ is computed correctly, and thus it contains $|C|$. This completes the proof of (9).

(10) *If \mathcal{C} contains $l \in \mathbb{N}$, then G contains a cycle C of length l .*

Consider the first time l has been added to \mathcal{C} . At that point, the algorithm has just added l to $M_t[Q, R, X]$ for some node t , set $Q = \{q_1, \dots, q_s\}$, set $X \subseteq A(t) \setminus Q$, and set $R = \{(q_1, q_2), \dots, (q_{s-1}, q_s), (q_s, q_1)\}$. Note that the corresponding path system $S_{Q,R,X}$ of length l is indeed an induced cycle C of length l with $C \cap \chi(t) = Q$. This completes the proof of (10).

Backtracking:

First, we extend the tables M_t so that for every $l \in M_t[Q, R, X]$, when we add l to $M_t[Q, R, X]$, we additionally store the current sets $Q_1, Q_2, R_1, R_2, X_1, X_2$ and the lengths l_1, l_2 queried in the tables $M_{t_1}[Q_1, R_1, X_1], M_{t_2}[Q_2, R_2, X_2]$ that were used to obtain a path system $S_{Q,R,X}$ of length l . This is only $O(1)$ overhead per entry.

Then, when we detect an induced cycle of a new length, we recursively look up all the vertices used to form this induced cycle. This requires $O(1)$ time per node, and as there are $O(n)$ nodes, the whole lookup takes time $O(n)$ per induced cycle. Since we only record at most c induced cycle lengths, the overall time spent for backtracking is thus also only $O(n)$. ■

9. FINDING THREE INDUCED CYCLES

In this section, we describe how to apply the algorithm from Theorem 1.2 to decide whether $\text{cl}(G) \geq 3$ in polynomial time. Before we provide this algorithm, we need to introduce a result from [3].

Theorem 9.1 (Berger, Seymour, Spirkl [3]). *Let G be a graph and let $u, v \in V(G)$ be two vertices. There exists an algorithm with running time $O(n^{18})$ that decides whether there exists a u - v path in G that is not a shortest u - v path. If such a path exists, the algorithm will output it.*

Now, we are ready to present our second algorithm.

Theorem 1.1. *Let G be a graph. There exists an algorithm with running time $O(|V(G)|^{22})$ that decides whether $\text{cl}(G) \geq 3$ and outputs a list of three induced cycles of distinct lengths if they exist.*

Proof. Suppose that G is $\{K_3, K_{2,2}\}$ -free. Let $k = f_{1.3}(c, 2)$ be the threshold from Theorem 1.3. Test whether $\text{tw}(G) \leq k - 1$ using Theorem 8.1. If so, then apply Theorem 1.2 and output the result. Otherwise, remove arbitrary vertices from G one by one until the treewidth of the resulting graph G' is at most $k - 1$. Add the last removed vertex back to G' to obtain

G'' . Observe that $\text{tw}(G'') = k$ and that by Theorem 1.3, G'' (and hence G) contains three cycles of distinct lengths. Apply Theorem 1.2 to G'' and output the result. The correctness follows since G'' is an induced subgraph of G . The running time of the algorithms from Theorems 8.1 and 1.2 is $O(n)$ and thus the total running time of this step is $O(n^2)$.

Now, suppose that G contains a K_3 . Thus, the shortest induced cycle in G has length three. We can find one cycle C^0 of length three by iterating over all triples of $V(G)$ in time $O(n^3)$. Then, for each edge $uv \in E(G)$, do the following:

- (1) Construct a graph G_{uv} from G by removing the edge uv and vertices in $N(u) \cap N(v)$.
- (2) Find a shortest u - v path P in G_{uv} , if one exists, by running breadth first search. If P exists, let C_{uv}^1 be the induced cycle of G with vertex set P .
- (3) Find a non-shortest u - v path N in G_{uv} , if one exists, using Theorem 9.1. If N exists, let C_{uv}^2 be the induced cycle of G with vertex set N .
- (4) If P above does not exist, let $L_{uv} = \emptyset$. If only P above exists, let $L_{uv} = \{C_{uv}^1\}$. If both P and N above exist, let $L_{uv} = \{C_{uv}^1, C_{uv}^2\}$.

Construct a set L of cycles by starting with $L = \emptyset$, examining each L_{uv} in turn, and adding the cycle $C \in L_{uv}$ to L if and only if L does not yet contain a cycle of length $|C|$. This can be done in time $O(n^2)$.

If $|L| < 2$, output “ $\text{cl}(G) < 3$ ”. If $|L| \geq 2$, choose two elements $C^1, C^2 \in L$, and output “ $\text{cl}(G) \geq 3$ ” and the list C^0, C^1, C^2 . By Theorem 9.1, the set L_{uv} can be constructed in time $O(n^{18})$ for each edge uv , and so the total running time is $O(n^{20})$.

We now prove correctness in the case when G contains a K_3 . Suppose first that the algorithm outputs “ $\text{cl}(G) \geq 3$ ”. Since $N_{G_{uv}}(u) \cap N_{G_{uv}}(v) = \emptyset$ for every $uv \in E(G)$, it follows that $|C| > 3$ for every $C \in L$. Thus, the lengths of the cycles C^0, C^1, C^2 are all distinct, and so the algorithm performs correctly.

Now, suppose that the algorithm outputs “ $\text{cl}(G) < 3$ ”. It follows that $|L| < 2$. Assume for a contradiction that $\text{cl}(G) \geq 3$. Then, there exist distinct $l_1, l_2 > 3$ and induced cycles C_1, C_2 in G such that $|C_i| = l_i$. For $i \in \mathbb{N}_2$, let $u_i v_i$ be an edge of C_i , and let P_i be the path obtained from C_i by removing the edge $u_i v_i$. Then, P_i is a path from u_i to v_i in $G_{u_i v_i}$. If P_i is not the shortest path from u_i to v_i in $G_{u_i v_i}$, then $|L_{u_i v_i}| = 2$, contrary to the fact that $|L| < 2$. Thus, for $i \in \mathbb{N}_2$, P_i is a shortest path from u_i to v_i in $G_{u_i v_i}$. Therefore, $|C_{u_1 v_1}^1| = |C_1|$ and $|C_{u_2 v_2}^1| = |C_2|$, and thus, L contains cycles of lengths l_1 and l_2 , again contrary to the fact that $|L| < 2$. This proves the algorithm works correctly when G contains a K_3 .

Therefore, we may assume that G is K_3 -free but contains a $K_{2,2}$. Thus, the shortest induced cycle in G has length four. We can find one cycle C^0 of length four by iterating over all 4-sets of $V(G)$ in time $O(n^4)$. Then, for each path $Q = v_1 v_2 v_3 v_4$ in G , do the following:

- (1) Construct a graph G_Q from G by removing $N(v_2) \setminus \{v_1\}$ and $N(v_3) \setminus \{v_4\}$.
- (2) Find a shortest v_1 - v_4 path P in G_Q , if one exists, by running breadth first search. If P exists, let C_Q^1 be the induced cycle of G with vertex set $P \cup \{v_2, v_3\}$.
- (3) Find a non-shortest induced v_1 - v_4 path N in G_Q , if one exists, using Theorem 9.1. If N exists, let C_Q^2 be the induced cycle of G with vertex set $N \cup \{v_2, v_3\}$.
- (4) If P above does not exist, let $L_Q = \emptyset$. If only P above exists, let $L_Q = \{C_Q^1\}$. If both P and N above exist, let $L_Q = \{C_Q^1, C_Q^2\}$.

Construct a set L of cycles by starting with $L = \emptyset$, examining each L_Q in turn, and adding the cycle $C \in L_Q$ to L if and only if L does not yet contain a cycle of length $|C|$. This can be done in time $O(n^4)$.

If $|L| < 2$, output “ $\text{cl}(G) < 3$ ”. If $|L| \geq 2$, choose two elements $C^1, C^2 \in L$, and output “ $\text{cl}(G) \geq 3$ ” and the list C^0, C^1, C^2 . By Theorem 9.1, the set L_Q can be constructed in time $O(n^{18})$ for each 4-path Q , and so the total running time is $O(n^{22})$.

Observe that deleting $N(v_2) \setminus \{v_1\}$ and $N(v_3) \setminus \{v_4\}$ ensures that each cycle $C \in L$ has length at least five since Q has length three, and P and N have length at least two. Then, the correctness follows by the same argument as in the case of K_3 . This concludes the proof. ■

10. ACKNOWLEDGMENTS

The authors thank Paul Seymour and Nicolas Trotignon for suggesting the problem to them and for several helpful discussions.

REFERENCES

- [1] Tara Abrishami, Bogdan Alecu, Maria Chudnovsky, Sepehr Hajebi, and Sophie Spirkl. Induced subgraphs and tree decompositions X. Towards logarithmic treewidth for even-hole-free graphs. Manuscript available at <https://arxiv.org/abs/2307.13684>, 2023.
- [2] Tara Abrishami, Bogdan Alecu, Maria Chudnovsky, Sepehr Hajebi, and Sophie Spirkl. Induced subgraphs and tree decompositions VII. Basic obstructions in H-free graphs. *Journal of Combinatorial Theory, Series B*, 164:443–472, 2024.
- [3] Eli Berger, Paul Seymour, and Sophie Spirkl. Finding an induced path that is not a shortest path. *Discrete Mathematics*, 344(7):112398, 2021.
- [4] Hans L. Bodlaender. A linear-time algorithm for finding tree-decompositions of small treewidth. *SIAM Journal on Computing*, 25(6):1305–1317, 1996.
- [5] Hans L. Bodlaender and Ton Kloks. Efficient and Constructive Algorithms for the Pathwidth and Treewidth of graphs. *Journal of Algorithms*, 21(2):358–402, 1996.
- [6] Maria Chudnovsky, Sepehr Hajebi, and Sophie Spirkl. Induced subgraphs and tree decompositions XVII. Anticomplete sets of large of treewidth. Manuscript available at <https://arxiv.org/abs/2411.11842>, 2024.
- [7] Maria Chudnovsky, Sepehr Hajebi, and Sophie Spirkl. Induced subgraphs and tree decompositions XVI. Complete bipartite induced minors. *Journal of Combinatorial Theory, Series B*, 176:287–318, 2026.
- [8] Reinhard Diestel. *Graph theory*, volume 173 of *Graduate Texts in Mathematics*. Springer, Berlin, Fifth edition, 2017.
- [9] P. Erdős and G. Szekeres. A combinatorial problem in geometry. *Compositio Mathematica*, 2:463–470, 1935. Available at Numdam.
- [10] Paul Erdős and Leo Moser. On the representation of directed graphs as unions of orderings. *Publicationes Mathematicae Debrecen*, 9:125–132, 1964.
- [11] Frank P. Ramsey. On a problem of formal logic. *Proceedings of the London Mathematical Society*, 30(4):264–286, 1929.