Detecting a long even hole

Linda Cook[1,2]
Princeton University, Princeton, NJ 08544, USA

Paul Seymour[2]
Princeton University, Princeton, NJ 08544, USA

August 28, 2019; revised October 29, 2021

[1]Current affiliation: Discrete Mathematics Group, Institute for Basic Science (IBS), Daejeon, Republic of Korea.

[2]This work was partially supported by AFOSR grant A9550-19-1-0187 and NSF grant DMS-1800053. Additionally, Linda Cook was partially supported by the Institute for Basic Science (IBS-R029-C1).

Abstract

For each integer $\ell \geq 4$, we give a polynomial-time algorithm to test whether a graph contains an induced cycle with length at least ℓ and even.

1 Introduction

All graphs in this paper are finite and have no loops or parallel edges. A *hole* in a graph is an induced subgraph which is a cycle of length at least four. The *length* of a path or cycle A is the number of edges in A and the *parity* of A is the parity of its length. For graphs G, H we will say that G *contains* H if some induced subgraph of G is isomorphic to H. We say G is *even-hole-free* if G does not contain an even hole. We denote by $|G|$ the number of vertices of a graph G. A graph algorithm is *polynomial-time* if its running time is at most polynomial in $|G|$.

This paper concerns detecting holes in a graph with length satisfying certain conditions. It is of course trivial to test for the existence of a hole of length at least ℓ, in polynomial time for each constant $\ell \geq 4$, as follows. We enumerate all induced paths P of length $\ell - 2$. For each choice of P, let its ends be x and y, let $P^* = V(P) \setminus \{x, y\}$, and let N be the set of vertices different from x, y that belong to or have a neighbour in P^*. Then we check whether x and y are in the same component of $G \setminus N$. This depends on ℓ being fixed; if ℓ is part of the input, then the problem is NP-complete, because it contains the Hamilton cycle problem with the following reduction. (Let G' be the graph obtained from the input graph G by subdividing every edge once. Then, G' contains a hole of length at least $\ell = 2n$ if and only if G contains a Hamilton cycle.)

But the problem is much less trivial if we impose restrictions on the parity of the hole length, or more generally on its residue class modulo some fixed number. Sepehr Hajebi [16] provided a proof in private communication that if ℓ is part of the input, then detecting holes of length at least ℓ in a specific residue class is W[1]-hard, and thus not fixed-parameter tractable unless the central conjecture of parameterized complexity theory (that "FPT $\neq$ W[1]") is false. More exactly, for all integers m, r with $m \geq 2$ and $0 \leq r < m$, if there is an algorithm that, with input G, ℓ, determines in time $\mathcal{O}(f(\ell)p(|G|))$ whether G contains a hole C of length at least ℓ and with $|E(C)| \cong r \mod m$, where f is some computable function and p is a polynomial, then FPT = W[1]. But this is different from what we are doing in this paper: we are working with ℓ fixed, and Hajebi wants ℓ part of the input.

Here is an overview of positive results about algorithms to detect even and odd holes, with odd holes first:

- In 2005, Chudnovsky, Cornuéjols, Liu, Seymour and Vušković [7] gave an $\mathcal{O}(|G|^9)$-time algorithm to test whether a graph G or its complement has an odd hole.
- In 2019, Chudnovsky, Scott, Seymour, and Spirkl [8] gave an algorithm to detect an odd hole in G in time $\mathcal{O}(|G|^9)$; and Lai, Lu and Thorup [15] improved this running time to $\mathcal{O}(|G|^8)$.
- Also in 2019, Chudnovsky, Scott and Seymour [4] gave a $\mathcal{O}(|G|^{20\ell+40})$-time algorithm to test whether G contains an odd hole of length at least ℓ, where ℓ is any fixed number.
- In 2021, Chudnovsky, Scott and Seymour [9] gave an algorithm that finds a shortest odd hole in G (if there is one) in time $\mathcal{O}(|G|^{14})$.

For even holes the story is a little different:

- In 2002, Conforti, Cornuéjols, Kapoor and Vušković [12] gave an approximately $\mathcal{O}(|G|^{40})$-time algorithm to test whether a graph contains an even hole, by using a structure theorem about even-hole-free graphs from an earlier paper [11].

- In 2005, Chudnovsky, Kawarabayashi, and Seymour [3] provided a simpler algorithm that searches for even holes directly in time $\mathcal{O}(|G|^{31})$ time.

- In 2015, Chang and Lu [1] gave an $\mathcal{O}(|G|^{11})$-time algorithm to determine whether a graph contains an even hole; and Lai, Lu and Thorup [15] improved this running time to $\mathcal{O}(|G|^{9})$ in 2020.

- In 2020, Hou-Teng Cheong and Hsueh-I Lu [2] pointed out that the algorithm of [3], designed to test for an even hole, actually outputs a shortest even hole if there is one.

One analogue of what has been done for odd holes is still open for even holes, namely the problem of detecting long even holes. That is what is solved in this paper.

We remark that even versus odd has been almost the entire focus of previous research, but what about holes of length a multiple of three, can we detect them in polynomial time? Triangle-free graphs with no holes of length a multiple of three have some very interesting properties [10], but we currently have no idea how to recognize such graphs.

Since we are looking for even holes of length at least ℓ, we might as well assume that ℓ is even. Our main result is the following:

Theorem 1.1. *For each even integer $\ell \geq 4$, there is an algorithm with the following specifications:*

Input: *A graph G.*

Output: *Decides whether G has an even hole of length at least ℓ.*

Running time: $\mathcal{O}(|G|^{9\ell+1})$.

Our algorithm combines approaches described in [3] and [4]. The algorithm uses a technique called "cleaning", as do the algorithms of [3], [4] and many other algorithms to detect induced subgraphs.

Here is an outline of the method. Every even hole has length at least 4, so when $\ell = 4$ we may apply any of the known even holes detection algorithms. Thus, we might as well assume that $\ell \geq 6$; and throughout the paper $\ell \geq 6$ is a fixed even integer, and a *long* hole or path is a hole or path of length at least ℓ. A *shortest* long even hole is a long even hole of minimum length. If C is a hole in G, a vertex v of $V(G) \setminus V(C)$ is *C-major* if there is no three-vertex path of C containing all neighbours of v in $V(C)$. A hole C is *clean* if it has no C-major vertex.

- First, we test for the presence in the input graph G of certain kinds of induced subgraphs ("short" long even holes, "long jewels of bounded order", "long thetas", "long ban-the-bombs", "long near-prisms") that are detectable in polynomial time (sometimes, under the assumption that earlier graphs in this list are not present) and whose presence would imply that G contains a long even hole. We call these kinds of subgraphs "easily-detectable configurations." We may assume these tests are unsuccessful.

- Second, we generate a *cleaning list*, a list of polynomially many subsets of $V(G)$ such that if C is a shortest long even hole in G, then for some set X in the list, X contains every C-major vertex and no vertex of C. This process depends on the absence of easily-detectable configurations.

- Third, for every X in our cleaning list we check whether $G \setminus X$ contains a clean shortest long even hole. This also depends on the absence of easily-detectable configurations. We detect a clean shortest long even hole C by guessing three evenly spaced vertices along C and taking shortest paths between them.

We are calling long near-prisms easily-detectable configurations, but "easily" might be a misnomer, because this is by far the computationally most expensive step of the algorithm, and the bulk of what is novel in the paper. For a general graph G, deciding whether G contains a long near-prism is NP-complete; Maffray and Trotignon's proof [14] that deciding whether G contains a prism is NP-complete can easily be adjusted to prove that deciding whether G contains a long near-prism is NP-complete. Fortunately it really is easy to detect the other "easily detectable" configurations, so we can assume there are none; and in such graphs we can detect the presence of long near-prisms in polynomial time.

The approach of determining whether G contains an even hole by first testing whether G contains a theta or a prism was outlined in [3]. Moreover, Chudnovsky and Kapadia gave an algorithm to decide whether G contains a theta or a prism in [6]. Their algorithm does not translate directly to long theta and long near-prism detection, but we were able to use a similar algorithmic structure for our purposes.

Note, this paper's algorithm for long even hole detection and the long odd holes detection algorithm of [4] do not output the *shortest* long even (odd) hole if there is one. They simply output whether or not the graph contains a long even (odd) hole. We expect that modifying our long even holes algorithm to find the length of the shortest long even hole if there is one would not be trivial. Our algorithm will terminate if the input graph contains an easily detectable configuration. Easily detectable configurations are guaranteed to contain long even holes, but they do not necessarily contain a shortest long even hole. If the input graph does not contain any easily-detectable configuration, the algorithm actually finds the shortest long even hole in the input graph.

2 The easily-detectable configurations

The *interior* P^* of a path P is the set of vertices of P that are not ends of P. Thus $P^* = \emptyset$ for a path P of length at most one. If $X, Y \subseteq V(G)$, we say X is *anticomplete* to Y if $X \cap Y = \emptyset$ and no vertex in X is adjacent to a vertex in Y. We begin with a test for what we called "short" long even holes:

Theorem 2.1. *For each integer $k \geq \ell$, there is an algorithm with the following specifications:*

Input: *A graph G.*

Output: *Decides whether G has a long even hole of length at most k.*

Running time: $\mathcal{O}(|G|^k)$.

Proof. We enumerate all vertex sets of size $\ell, \ell+1, \ldots, k$ and for each one, check whether it induces an even hole. Since k is a constant, checking each set takes $\mathcal{O}(1)$. □

We need the following easily-detectable configuration of [4] (slightly modified). Let $u, v \in V(G)$ and let Q_1, Q_2 be induced paths between u, v of different parity. Let P be an induced path between

u, v of length at least $\ell - \min(|E(Q_1)|, |E(Q_2)|)$, such that P^* is anticomplete to $Q_1^* \cup Q_2^*$. We say the subgraph induced on $V(P \cup Q_1 \cup Q_2)$ is a *long jewel of order* $\max(|V(Q_1)|, |V(Q_2)|)$ *formed by* Q_1, Q_2, P. Any graph containing a long jewel has a long even hole, since the holes $P \cup Q_1$ and $P \cup Q_2$ are both long holes and one of them is even.

We need a slight extension of Theorem 2.2 of [4]:

Theorem 2.2. *For each integer $k \geq \ell$, there is an algorithm with the following specifications:*

Input: *A graph G.*

Output: *Decides whether G has a long jewel of order at most k.*

Running time: $\mathcal{O}(|G|^n)$ *where* $n = k + 1 + \max(k, \ell - 1)$.

Proof. We enumerate all triples of induced paths Q_1, Q_2, R of G, such that:

- Q_1, Q_2 join the same pair of vertices, say u, v;
- one of Q_1, Q_2 is odd and the other is even, and each has at most k vertices;
- R has length $\ell - \min(|E(Q_1)|, |E(Q_2)|) - 2$ (or zero if this number is negative), and has one end u and the other some vertex w say;
- no vertex of $V(R) \setminus \{u\}$ equals or has a neighbour in $V(Q_1 \cup Q_2) \setminus \{u\}$.

For each such triple of paths, let X be the set of vertices of G that are different from and nonadjacent to each vertex of $V(Q_1 \cup Q_2 \cup R) \setminus \{v, w\}$. We test whether there is a path in $G[X \cup \{w, v\}]$ between w, v. If so we output that G contains a long jewel of order at most k. If no triple yields this outcome, we output that G has no such long jewel.

To see the correctness of the algorithm, certainly the output is correct if G contains no long jewel of order at most k. Suppose then it does, say formed by Q_1, Q_2, P. Let u, v be the ends of P, and let R be the subpath of P of length $\ell - \min(|E(Q_1)|, |E(Q_2)|) - 2$ (or zero if this number is negative) with one end u. When the algorithm tests the triple Q_1, Q_2, R, it will discover there is a path in $G[X \cup \{w, v\}]$ between w, v, because the remainder of P is such a path. Consequently the output is correct.

The running time is $O(|G|^2)$ for each triple of paths, and there are at most $|G|^z$ such triples where $z = k - 1 + \max(k, \ell - 1)$, so the running time is as claimed. This proves Theorem 2.2. $\square$

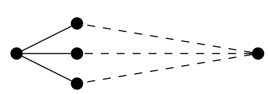

Figure 1: A theta (dashed lines mean paths of arbitrary positive length)

A *theta* is a graph consisting of two nonadjacent vertices u, v and three induced paths P_1, P_2, P_3 joining u, v with pairwise disjoint and anticomplete interiors, and we say P_1, P_2, P_3 *form* a theta. The union of any two of P_1, P_2, P_3 is a hole, and a *long theta* is a theta where all three holes are long. If G contains a long theta, then it contains a long even hole, because at least two of P_1, P_2, P_3 must have the same parity. To detect long thetas, we use the "three-in-a-tree" algorithm of [5], the following:

Theorem 2.3. *There is an algorithm with the following specifications:*

Input: *A graph G and three vertices v_1, v_2, v_3 of G.*

Output: *Decides whether there is an induced subgraph T of G with $v_1, v_2, v_3 \in V(T)$ such that T is a tree.*

Running time: $\mathcal{O}(|G|^4)$.

Chudnovsky and Seymour's algorithm in [5] to detect a theta in a graph G can be adjusted to detect a long theta, as follows:

Theorem 2.4. *There is an algorithm with the following specifications:*

Input: *A graph G.*

Output: *Decides whether G contains a long theta.*

Running time: $\mathcal{O}(|G|^{2\ell-1})$.

Proof. The algorithm is as follows. Say (temporarily) a *claw* is a graph that is the union of three paths Q_1, Q_2, Q_3, with a common end a and otherwise vertex-disjoint, of lengths k_1, k_2, k_3 respectively where $k_1, k_2, k_3 \geq 2$, and $k_1 + k_2, k_2 + k_3, k_3 + k_1 \geq \ell - 2$, and $k_1 + k_2 + k_3 \leq 2\ell - 6$. If three paths P_1, P_2, P_3 of G form a long theta, then $P_1 \cup P_2 \cup P_3$ includes a claw which is an induced subgraph of G. (To see this, if P_1, P_2, P_3 all have length at least $\ell/2$ take Q_1, Q_2, Q_3 all of length $\ell/2 - 1$, and if say P_3 has length less than $\ell/2$, take $Q_3 = P_3$ and Q_1, Q_2 of length $\ell - 2 - |E(P_3)|$.) Conversely, if three paths form a theta that includes a claw, then the theta is long.

Let B be a claw in G, and let q_1, q_2, q_3 be its three vertices of degree one in B. Let G' be the graph obtained from G by deleting all vertices different from q_1, q_2, q_3 that belong to or have a neighbour in $V(B) \setminus \{q_1, q_2, q_3\}$. Then B is an induced subgraph of a theta (and hence of a long theta) in G if and only if there is an induced tree T containing q_1, q_2, q_3 in G'.

So the algorithm is: enumerate all induced claws, and for each one, check if there is an induced tree as above. Since claws have at most $2\ell - 5$ vertices, there are only $\mathcal{O}(|G|^{2\ell-5})$ of them, so the running time is $\mathcal{O}(|G|^{2\ell-1})$. This proves Theorem 2.4. $\square$

Lai, Lu and Thorup [15] provide a faster algorithm for the three-in-a-tree problem. Using their $\mathcal{O}(|E(G)|(\log |G|)^2)$ algorithm we can reduce the running time for detecting a long theta to $\mathcal{O}(|G|^{2\ell-3}(\log |G|)^2)$, but this improvement does not affect the asymptotic running time of our long even holes detection algorithm.

For brevity, it is convenient to describe enumerating all subgraphs of a certain type as "guessing" subgraphs of that type. In this language the algorithm can be written as follows: We guess the paths Q_1, Q_2 and Q_3 and test whether q_1, q_2, q_3 are contained in some induced tree of G'.

We call a path P with ends x, y an *xy-path.* If P is a path, and $x, y \in V(P)$, we denote the subpath of P with ends x, y by x-P-y. A path with vertices $v_1, \dots, v_k$ in order is denoted by v_1-$\cdots$-v_k. If P, Q are paths with ends u, v and v, w respectively, and their union is a path with ends u, w, we denote this path by u-P-v-Q-w; and extend this notation for longer concatenations similarly.

Let us say a *ban-the-bomb* is a graph consisting of

- a cycle u-v_1-w-v_2-u of length four, and possibly the edge uw (but we insist that v_1, v_2 are nonadjacent); and one further vertex x adjacent to u, and nonadjacent to v_1, v_2, w; and
- for $i = 1, 2$, an induced xv_i-path P_i of length at least two, where P_i^* is anticomplete to $\{u, v_{3-i}, w\}$, and $V(P_1) \setminus \{x\}$ is anticomplete to $V(P_2) \setminus \{x\}$.

Thus it has three holes; and it is *long* if all three holes are long. It is easy to see that every graph containing a long ban-the-bomb has a long even hole.

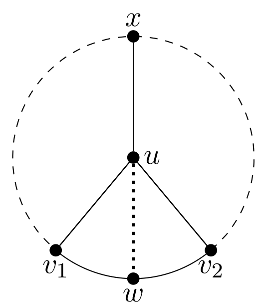

Figure 2: A ban-the-bomb. The dotted line is a possible edge.

If there is no long theta, we can also search for long ban-the-bombs using the three-in-a-tree algorithm, as follows.

Theorem 2.5. *There is an algorithm with the following specifications:*

Input: *A graph G with no long theta.*

Output: *Decides whether G contains a long ban-the-bomb.*

Running time: $\mathcal{O}(|G|^{2\ell+1})$.

Proof. Let us say a *bomb* is a graph consisting of a path R of length $2\ell - 6$, with middle three vertices v_1-w-v_2 in order and two more vertices u, x, where u is adjacent to v_1, v_2 and possibly to w, but to no other vertices of R, and x adjacent to u but to no vertex of R.

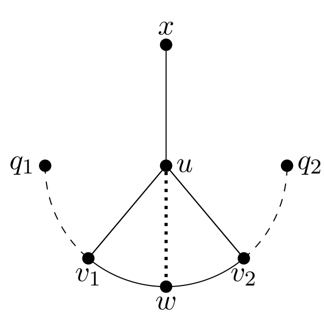

Figure 3: A bomb. The dashed lines are paths of length $2\ell - 6$, and the dotted line is a possible edge.

If there is a long ban-the-bomb in G, with vertices u, v_1, v_2, w, x and paths P_1, P_2 as in the definition, then P_1, P_2 both have length at least $\ell - 2$. For $i = 1, 2$ let Q_i be the subpath of P_i of length $\ell - 4$ with one end v_i, and let q_i be the other end of Q_i; then the subgraph induced on $V(Q_1 \cup Q_2) \cup \{u, w, x\}$ is a bomb. To search for long ban-the-bombs, we enumerate all induced subgraphs of G that are bombs. For each such induced bomb B, let L be its three vertices of degree one; check if there is an induced tree containing the vertices in L, in the graph obtained from G by deleting all vertices not in L that belong to or have neighbours in $V(B) \setminus L$. If so, output that G contains a long ban-the-bomb and stop. If no bomb has such a tree, output that there is no long ban-the-bomb.

This concludes the description of the algorithm. A bomb has $2\ell - 3$ vertices, so there are $\mathcal{O}(|G|^{2\ell-3})$ choices for the bomb, and the running time is $\mathcal{O}(|G|^{2\ell+1})$.

If a bomb is contained in a long ban-the-bomb, then such a tree exists, and the outcome is correct, but the converse is less clear. Suppose that for some bomb B there is a tree T as described in the algorithm. Let R be as in the definition of a bomb, with ends q_1, q_2 where q_1, v_1, w, v_2, q_2 are in order. There is a vertex $t \in V(T)$ and three paths T_1, T_2, T_3 of T (possibly of length zero) between t and q_1, q_2, x respectively, pairwise anticomplete except for t. For $i = 1, 2$, the hole t-T_i-q_i-R-v_i-x-T_3-t is long, since $T_i \cup T_3$ has length at least two; so if $t \neq x$, there is a long theta formed by the paths t-T_i-q_i-R-v_i-u for $i = 1, 2$, and the path t-T_3-x-u, a contradiction. Thus $t = x$, and we have a long ban-the-bomb. This proves correctness.

$\square$

A *triangle* is a graph consisting of three pairwise adjacent vertices. A *near-prism* is a graph consisting of two triangles with vertex sets $\{a_1, a_2, a_3\}$ and $\{b_1, b_2, b_3\}$, sharing at most one vertex, and three pairwise vertex-disjoint induced paths P_1, P_2, P_3, such that P_i has ends a_i and b_i for $i \in \{1, 2, 3\}$ and the only edges between $V(P_1)$, $V(P_2)$ and $V(P_3)$ are those of the triangles on $\{a_1, a_2, a_3\}$ and $\{b_1, b_2, b_3\}$. A near-prism is *long* if the subgraph induced on $V(P_i \cup P_j)$ is a long hole for all distinct $i, j \in \{1, 2, 3\}$. It is a *prism* if the two triangles are vertex-disjoint. We call P_1, P_2, P_3 the *constituent paths* of the near-prism. It is easy to see that every graph with a long near-prism has a long even hole.

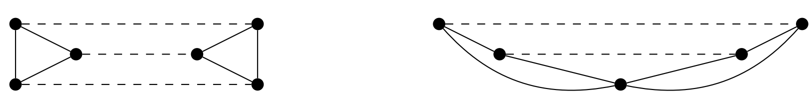

Figure 4: Near-prisms.

3 Detecting a clean lightest long near-prism

Our next goal is a poly-time algorithm to test whether G contains a long near-prism, for graphs G that contain none of the other easily-detectable configurations; and this and the next two sections are devoted to this. Let us say a graph G is a *prospect* if G contains no long even hole of length at most 2ℓ, no long jewel of order at most $\ell + 1$, no long theta and no long ban-the-bomb. We will show the following (the outline of this algorithm is like that of [6]):

Theorem 3.1. *There is an algorithm with the following specifications:*

Input: *A prospect G.*

Output: *Decides whether G contains a long near-prism.*

Running time: $\mathcal{O}(|G|^{9\ell+1})$.

We need:

Lemma 3.2. *Let G be a prospect, and let K be a long near-prism in G, with constituent paths P_1, P_2, P_3. Then at least two of P_1, P_2, P_3 have length at least ℓ.*

Proof. Suppose that P_1, P_2 both have length less than ℓ. Then the long hole induced on $V(P_1 \cup P_2)$ is odd, since its length is between ℓ and 2ℓ, and G is a prospect; and so the paths a_3-a_1-P_1-b_1-b_3 and a_3-a_2-P_2-b_2-b_3 have different parity. Hence these two paths with P_3 form a long jewel of order at most $\ell + 1$, a contradiction. This proves Lemma 3.2.

□

For a graph G and $x, y \in V(G)$, we call the length of a shortest xy-path in G the *G-distance* between x and y and denote it by $d_G(x, y)$. Let us say a *frame* F is a graph with the following properties:

- F is the union of two triangles with vertex sets A, B with at most one vertex in common, and three graphs F_1, F_2, F_3 that are pairwise vertex-disjoint; and each of F_1, F_2, F_3 has exactly one vertex in A and one in B;
- for $1 \le i \le 3$, F_i is either a path with one end in A and the other in B of length at most $\ell - 1$, or the disjoint union of two paths, both of length exactly $\ell/2 - 1$, one with an end in A and the other with an end in B (it follows that if the triangles share a vertex then one of the F_i is a path of length zero); and
- at most one of F_1, F_2, F_3 is a path.

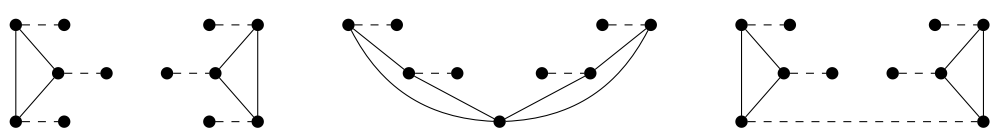

Figure 5: Frames

We call A, B the *bases* of the frame. A *frame in G* means an induced subgraph of G that is a frame. The *ends* of a frame F are its vertices of degree one, and the set of vertices of F that are not ends of F is denoted by F^*.

If K is a near-prism in a prospect G, then K is long if and only if K contains a frame, by Lemma 3.2. (Indeed, every long near-prism contains a unique frame, which we denote by F_K.) Thus if at some stage we have a frame F in the input prospect G, and we find a near-prism K of G containing F, then we know that K is long without having to check the lengths of the missing parts of its constituent paths; and conversely, if there is a long near-prism K in G, and we examine all frames in G and test, for each one, whether it is contained in a long near-prism, then eventually we will test F_K and report success. (Enumerating all frames can be done in polynomial time, since frames have a bounded number of vertices; the more difficult issue is to handle a given frame in polynomial time.)

That is our basic method, to try all frames and see if they can be extended to long near-prisms. But it is helpful to have a little more information about the long near-prism we are looking for than just its frame. For instance, for the first frame in figure 5, we would like to know which vertex of the left triangle corresponds to which one of the right. Let us say an *ordered frame* $\mathcal{F}$ consists of a frame F together with a linear ordering of both of its bases. Let K be a long near-prism and let F_K be its frame. Let K have constituent paths P_1, P_2, P_3, where $|E(P_1)| \le |E(P_2)| \le |E(P_3)|$, and for $1 \le i \le 3$ let P_i have ends a_i, b_i. Then these six (or possibly, five) vertices belong to the two bases of F_K, and we would like to know this labelling. We define $\mathcal{F}_K$ to be the ordered frame consisting of F_K and the orderings $a_1 < a_2 < a_3$ and $b_1 < b_2 < b_3$, and call it an *ordered frame of* K. (It is not quite unique, because two of P_1, P_2, P_3 might have the same length.)

If $\mathcal{F}$ is an ordered frame of a long near-prism K, with bases $\{a_1, a_2, a_3\}, \{b_1, b_2, b_3\}$ and constituent paths P_1, P_2, P_3, where P_i has ends a_i, b_i for $i = 1, 2, 3$, we say that P_1, P_2, P_3 are *numbered according to* $\mathcal{F}$ if the orderings of $\mathcal{F}$ are $a_1 < a_2 < a_3$ and $b_1 < b_2 < b_3$.

If K is a long near-prism in G, we call a vertex $q \in V(G) \setminus V(K)$ K*-major* if there is no three-vertex path P of K containing all neighbours of q in $V(K)$. Note, it does not matter for the sake of the proof, but P might contain vertices of the bases. We say K is *clean* if there are no K-major vertices. We call a long near-prism K' *shorter* than a long near-prism K if $|V(K')| < |V(K)|$, and thereby define a *shortest* long near-prism.

We will test for long near-prisms as follows. Shortest long near-prisms have special properties that make them easier to detect than general long near-prisms, so we will hunt for a near-prism with these special properties. Sometimes it is convenient to pin down the target even further: we will hunt for the "lightest" long near-prism, the lexicographically earliest of all shortest long near-prisms.

To do this we will first guess its ordered frame: so now we need a poly-time algorithm that, given an ordered frame, will test whether there is a lightest long near-prism with this ordered frame. This comes in two phases:

- Given an ordered frame, we generate a "cleaning list" of polynomially many sets of vertices, such that for every shortest long near-prism K of G with the given ordered frame, there exists X in the list such that K is clean in $G \setminus X$; that is, X is disjoint from $V(K)$, and X contains all K-major vertices. This is explained in section 5.
- For each X in this cleaning list, we search for a clean lightest long near-prism with the given ordered frame in $G \setminus X$. This algorithm is explained in the remainder of this section.

A long near-prism K in G is *tidy* if F_K^* is anticomplete to $V(G) \setminus V(K)$. If we have a frame F in G, and we are trying to test if there is a long near-prism K with $F_K = F$, we might as well delete all vertices of G not in $V(F)$ that have a neighbour in F^*, because no such vertex belongs to $V(K)$. If G' is the graph that remains, and the long near-prism we are looking for exists, then it is tidy in G'.

Lemma 3.3. *Let G be a prospect, and let K be a tidy shortest long near-prism in G, with constituent paths P_1, P_2, P_3. For all distinct $i, j \in \{1, 2, 3\}$, there is no induced path Q of G with one end in $V(P_i)$ and the other in $V(P_j)$, such that*

- *$V(Q)$ is anticomplete to $V(P_k)$ where $k \in \{1, 2, 3\} \setminus \{i, j\}$;*

- *no vertex of Q^* is K-major; and*
- $2|E(Q)| \le 1 + \min(|E(P_i)|, |E(P_j)|)$.

Proof. Suppose that there is such a path Q for some shortest long near-prism K in G, and choose Q, K with minimal union. Let the vertices of Q be q_0-q_1-$\cdots$-q_t-q_{t+1} where $q_0 \in V(P_i)$ and $q_{t+1} \in V(P_j)$. From the minimality of $Q \cup K$, none of $q_2, \ldots, q_{t-1}$ has a neighbour in $V(K)$. Since $q_1, \ldots, q_t$ are not K-major, it follows that $t \ge 2$, and there is a three-vertex path of K that contains all neighbours of q_t in $V(K)$; and since K is tidy, all neighbours of q_t in $V(K)$ belong to $V(P_j)$, and so there is a minimal subpath R_j of P_j containing all neighbours of q_t in $V(K)$. Thus R_j has length zero, one or two, and we will treat these cases separately. Let the ends of R_j be u_j, v_j, where a_j, u_j, v_j, b_j are in order in P_j (in the usual notation). Since K is tidy, the paths a_j-P_j-u_j and v_j-P_j-b_j both have length at least $\ell/2 - 1$. Similarly, there is a minimal subpath R_i of P_i containing all neighbours of q_i in $V(K)$, of length at most two, with ends u_i, v_i say. Let $k \in \{1, 2, 3\} \setminus \{i, j\}$.

(1) *R_j does not have length one.*

Suppose it does. Let S be the induced path from q_1 to a_i with interior in $V(P_i)$; then there is a prism with bases $\{a_1, a_2, a_3\}$, $\{q_t, u_j, v_j\}$ and constituent paths

$$a_i\text{-}S\text{-}q_1\text{-}\cdots\text{-}q_t,$$

$$a_j\text{-}P_j\text{-}u_j,$$

$$a_k\text{-}P_k\text{-}b_k\text{-}b_j\text{-}P_j\text{-}v_j.$$

All of its holes are long, since $2(\ell/2-1)+4 \ge \ell$, and so it is a long prism, and hence not shorter than K. Consequently $|E(S)|+t-1 \ge |E(P_i)|$. Similarly, let T be the induced path from q_i to b_i with interior in $V(P_i)$; then $|E(T)|+t-1 \ge |E(P_i)|$. Adding, we obtain that $|E(S)|+|E(T)|+(2t-2) \ge 2|E(P_i)|$. But $|E(S)|+|E(T)| \le |E(P_i)|+2$, and so $2t \ge |E(P_i)|$, contrary to the hypothesis, since $|E(Q)| = t+1$. This proves (1).

(2) *R_j does not have length zero.*

Suppose it does; so $u_j = v_j = q_{t+1}$. By (1) with P_i, P_j exchanged, it follows that either $u_i = v_i$ or u_i, v_i are nonadjacent. If $u_i = v_i$ there is a long theta with constituent paths Q and

$$q_0\text{-}P_i\text{-}a_i\text{-}a_j\text{-}P_j\text{-}q_{t+1}$$

$$q_0\text{-}P_i\text{-}b_i\text{-}b_j\text{-}P_j\text{-}q_{t+1},$$

a contradiction. If u_i, v_i are nonadjacent, there is a long theta with constituent paths

$$q_1\text{-}\cdots\text{-}q_t,$$

$$q_1\text{-}u_i\text{-}P_i\text{-}a_i\text{-}a_j\text{-}P_j\text{-}q_{t+1}$$

$$q_1\text{-}v_i\text{-}P_i\text{-}b_i\text{-}b_j\text{-}P_j\text{-}q_{t+1},$$

a contradiction. This proves (2).

From (1) and (2) we may assume that R_i, R_j both have length two. Let P'_j be the path obtained from P_j by replacing the subpath R_j by u_j-q_t-v_j. Then P_i, P'_j, P_k are the constituent paths of a shortest long near-prism K', also with frame F_K. From the minimality of $Q \cup K$, one of $q_1, \dots, q_{t-1}$ is K'-major. Since it is not K-major, it is adjacent to q_t, and hence must be q_{t-1}. Thus q_{t-1} has a neighbour in $V(K)$, and so $t = 2$ (because none of $q_2, \dots, q_{t-1}$ has a neighbour in $V(K)$). But then the subgraph induced on $V(P_i \cup P'_j) \cup \{q_1\}$ is a long ban-the-bomb, a contradiction. This proves Lemma 3.3. $\square$

If F is a frame with bases A, B, and v is an end of F, choose $u \in V(A \cup B)$ with minimum F-distance to v; we call v the *u-end* of K. For each $u \in V(A \cup B)$ there is at most one u-end of F, but there might be none. For a path P with ends a, b, we call $v \in V(P)$ a *midpoint* of P if $|d_P(v, a) - d_P(v, b)| \leq 1$.

We would like to assign weights to the edges of G, all very close to one and all different, such that no two different sets of edges X, Y have the same total weight, and if $|X| < |Y|$ then the total weight in X is less than that in Y. A convenient way to do this, and a way that is easy to handle algorithmically, is to take an arbitrary linear ordering of $E(G)$, say $E(G) = \{e_1, \dots, e_n\}$, and let edge e_i have weight $1 + 2^{-i}$ for each i; then the total weight in a set X is less than that in a set Y if and only if either $|X| < |Y|$, or $|X| = |Y|$ and X is lexicographically earlier than Y (the latter means that Y contains e_i where $i \in \{1, \dots, n\}$ is minimum with $e_i \in (X \setminus Y) \cup (Y \setminus X)$). So, let us take some linear order of $E(G)$, and for $X, Y \subseteq E(G)$, we say X is *lighter* than Y if either $|X| < |Y|$, or $|X| = |Y|$ and X is lexicographically earlier than Y. If G has a long near-prism, it has at least one shortest long near-prism, and exactly one of them is the lightest long near-prism; and we find that for algorithms it is better to hunt for the lightest long near-prism than just a shortest one. These weights have $O(|G|^2)$ bits, so doing arithmetic with them is a little time-consuming; but we can certainly find the lightest st-path in time $\mathcal{O}(|G|^3)$ (and if it mattered, we could do it faster).

Lemma 3.4. *Let G be a prospect and let K be a shortest long near-prism in G. Suppose that K is tidy, clean, and has an ordered frame $\mathcal{F}$. Let P_1, P_2, P_3 be the constituent paths of K numbered according to $\mathcal{F}$. Let $i \in \{1, 2, 3\}$ and let $v, w \in P_i^*$. Let S be the set of all vertices that are equal or adjacent to a vertex in $V(P_j)$ for any $j \in \{1, 2, 3\}$ satisfying $j < i$. Then, v-P_i-w is the lightest vw-path in $G \setminus S$.*

Proof. Let M be a lightest vw-path in $G \setminus S$ and suppose M is not the path v-P_i-w. We will analyse the properties of M in G. If M^* is anticomplete to every vertex in $V(K) \setminus V(v\text{-}P_i\text{-}w)$ then $K \setminus V(v\text{-}P_i\text{-}w) \cup M$ is a long near-prism and it is lighter than K, a contradiction. Hence, we may assume for some $j \in \{1, 2, 3\}$ with $j > i$, there is an edge with one end in M^* and the other end in $V(P_j)$. Let M' be a shortest subpath of M with one end equal to v or w and the other end adjacent to a vertex $q \in V(P_j)$ for some $j \in \{1, 2, 3\} \setminus \{i\}$. Then,

$$|E(M')| \leq \frac{1}{2}|E(M) \leq \frac{1}{2}|E(v\text{-}P_i\text{-}w)| \leq \frac{1}{2}|E(P_i)| \leq \frac{1}{2}|E(P_j)|$$

Since $V(M)$ contains no K-major vertices it follows that $V(M')$ is anticomplete to $V(P_k)$ for $k \in \{1, 2, 3\} \setminus \{i, j\}$. Hence, the path M'-q contradicts Lemma 3.3. $\square$

Theorem 3.5. *There is an algorithm with the following specifications:*

Input: *A prospect G, a linear order of $E(G)$, and an ordered frame $\mathcal{F}$ in G.*

Output: *Decides either that G contains a long near-prism with ordered frame $\mathcal{F}$, or that there is a no long near-prism that is the lightest among all long near-prisms, and has ordered frame $\mathcal{F}$, and is clean.*

Running time: $\mathcal{O}(|G|^3)$.

Proof. Here is the algorithm. Let the frame F of $\mathcal{F}$ have bases $\{a_1, a_2, a_3\}$ and $\{b_1, b_2, b_3\}$, where the linear orders of $\mathcal{F}$ are $a_1 < a_2 < a_3$ and $b_1 < b_2 < b_3$. For $1 \leq i \leq 3$, let s_i be the a_i-end of F and let t_i be the b_i-end of F, if they exist. (Certainly s_2, t_2, s_3, t_3 exist, but s_1, t_1 might not.) Let W_0 be the set of all vertices of G that are not ends of F, and belong to or have a neighbour in F^*, and let $G_0 = G \setminus W_0$.

Step 1: If s_1 is defined, compute the lightest s_1t_1-path M_1 in G_0 (if there is no such path, output "failure", that is, the desired near-prism does not exist, and stop). If s_1 is not defined, let M_1 be the null graph. In either case let W_1 be the set of vertices of G_0 that belong to or have a neighbour in $V(M_1)$, and let $G_1 = G_0 \setminus W_1$.

Step 2: Compute the lightest s_2t_2-path M_2 in G_1 (reporting failure if there is no such path). Let W_2 be the set of vertices of G_1 that belong to or have a neighbour in $V(M_2)$, and let $G_2 = G_1 \setminus W_2$.

Step 3: Compute the lightest s_3t_3-path M_3 in G_2 (reporting failure if there is no such path).

Step 4: Check whether $F \cup M_1 \cup M_2 \cup M_3$ is a long near-prism in G, and if so, output that fact and stop.

This concludes the description. For running time, we just have to find the sets W_0, W_1, W_2, which take time $\mathcal{O}(|G|^2)$, and solve three lightest-path problems, so the total running time is $\mathcal{O}(|G|^3)$.

To prove correctness: the positive output is clearly correct, but we need to check the negative output. Assume then that there is a long near-prism K that is the lightest among all near-prisms, and it has ordered frame $\mathcal{F}$, and is clean. Let its constituent paths be P_1, P_2, P_3, numbered according to $\mathcal{F}$, and hence with $|E(P_1)| \leq |E(P_2)| \leq |E(P_3)|$.

We claim that in step 1 above, the algorithm will compute some M_1, and if s_1, t_1 exist then M_1 is the path s_1-P_1-t_1. To see this, the claim is true if s_1, t_1 do not exist, so we assume they do. Then there is an s_1t_1-path in G_0, namely the path s_1-P_1-t_1, and so the algorithm will not report failure in step 1, and so computes the lightest s_1t_1-path M_1 in G_0. Then by Lemma 3.4, s_1-P_i-t_1 equals M_1. This proves our claim.

Similarly in step 2, the algorithm computes M_2. If s_2, t_2 exist then M_2 is the path s_2-P_2-t_2 by Lemma 3.4 because s_2-P_2-t_2 is lightest in G_1 (though not necessarily in G_0). And in step 3 the algorithm computes M_3; and $F \cup M_1 \cup M_2 \cup M_3$ is a long near-prism, and the output is correct. This proves Theorem 3.5. $\square$

4 Major vertices on near-prisms

In this section we prove some properties of K-major vertices, when K is a shortest long near-prism. If K is a long near-prism with constituent paths P_1, P_2, P_3, and each P_i has ends a_i, b_i as usual, and x is K-major with a neighbour in $V(P_i)$, we define $\alpha_i(x)$ to be the neighbour v of x in $V(P_i)$ such that the path v-P_i-a_i is minimal; and define $\beta_i(x)$ to be the neighbour v of x in $V(P_i)$ such that the path v-P_i-b_i is minimal. We begin with some lemmas:

Lemma 4.1. *Let K be a tidy shortest long near-prism in a graph G. If x is a K-major vertex, then x has neighbours in at least two constituent paths of K.*

Proof. In the usual notation, suppose that all neighbours of x in $V(K)$ are contained in $V(P_1)$, say; so $\alpha_1(x)$-P_1-$\beta_1(x)$ has length strictly greater than two. We obtain a near-prism K' shorter than K by replacing $\alpha_1(x)$-P_1-$\beta_1(x)$ in P_1 with the path $\alpha_1(x)$-x-$\beta_1(x)$. Since K' contains the same frame as K, it follows that K' is a long near-prism, a contradiction. This proves Lemma 4.1. □

Lemma 4.2. *Let K be a tidy shortest long near-prism in a graph G, with constituent paths P_1, P_2, P_3. For all distinct $i, j \in \{1, 2, 3\}$, if x is a K-major vertex with no neighbours in $V(P_j)$, then x either has exactly one neighbour in $V(P_i)$, or two nonadjacent neighbours in $V(P_i)$.*

Proof. Suppose that x has no neighbour in $V(P_3)$, and $\alpha_1(x), \beta_1(x)$ are distinct and adjacent, say. Then there is a long prism with bases $\{a_1, a_2, a_3\}$ and $\{x, \alpha_1(x), \beta_1(x)\}$ and constituent paths

$$a_1\text{-}P_1\text{-}\alpha_1(x)\text{-}x,$$

$$a_2\text{-}P_2\text{-}\alpha_2(x),$$

$$a_3\text{-}P_3\text{-}b_3\text{-}b_1\text{-}P_1\text{-}\beta_1(x),$$

and it is shorter than K, a contradiction. This proves Lemma 4.2.

□

Lemma 4.3. *Let G be a graph with no long theta, and let K be a tidy shortest long near-prism in G. If x is a K-major vertex, then x has three pairwise nonadjacent neighbours in $V(K)$.*

Proof. Suppose not. By Lemma 4.1, in the usual notation we may assume x has a neighbour in $V(P_1)$ and a neighbour in $V(P_2)$, and we may assume that x has no neighbours in $V(P_3)$. If x has exactly one neighbour in $V(P_1)$ and exactly one neighbour in $V(P_2)$, then $V(P_1 \cup P_2) \cup \{x\}$ induces a long theta. So by Lemma 4.2 we may assume that x has two nonadjacent neighbours in $V(P_1)$; but then the claim is true. This proves Lemma 4.3. □

Lemma 4.4. *Let G be a graph with no long theta, and let K be a tidy shortest long near-prism in G, with constituent paths P_1, P_2, P_3. Let x, y be nonadjacent K-major vertices. If $i, j \in \{1, 2, 3\}$, and x has no neighbours in $V(P_i)$ and y has no neighbours in $V(P_j)$ then $i = j$.*

Proof. Suppose that x has no neighbours in $V(P_3)$, and y has no neighbours in $V(P_1)$, say. Then x has neighbours in $V(P_1)$ and in $V(P_2)$, and y has neighbours in $V(P_2)$ and $V(P_3)$ by Lemma 4.1. Let M be an induced xy-path with interior in $V(P_2)$. By Lemma 4.2, $\alpha_1(x), \beta_1(x)$ are either equal or nonadjacent, and $\alpha_3(y), \beta_3(y)$ are either equal or nonadjacent. Thus there are four cases, but in each case there is a long theta induced on the union of the vertex sets of the paths $\alpha_1(x)$-P_1-a_1, $\beta_1(x)$-P_1-b_1, $\alpha_3(y)$-P_3-a_3, $\beta_3(y)$-P_3-b_3 and M, a contradiction. This proves Lemma 4.4.

□

We need some more definitions. Let K be a tidy shortest long near-prism in G. For $v \in V(K) \setminus F_K^*$ and integers $m, n \geq 0$, we define the path $K_n^m(v)$ as follows. In the usual notation, let $v \in V(P_i)$ say. Let M be the maximal subpath of the path v-P_i-a_i that has one end v and has length at most m, and has no internal vertex in F_K^*. (Thus, M is permitted to have an end in F_K^*, but no more.) Let N be the maximal subpath of the path v-P_i-b_i that has one end v and has length at most n, and has no internal vertex in F_K^*; and let $K_n^m(v) = M \cup N$.

Also, if x is K-major, then for $1 \leq i \leq 3$, if x has a neighbour in $V(P_i)$ let $A_i(x)$ be the vertex set of the path $\alpha_i(x)$-P_i-a_i, and if x has no such neighbour let $A_i(x) = V(P_i)$. For $i, j \in \{1,2,3\}$, let $A_{i,j}(x) = A_i(x) \cup A_j(x)$, and let $A_{1,2,3}(x) = A_1(x) \cup A_2(x) \cup A_3(x)$. If x, y are K-major, we say that y is *distant* from x if

- x, y are nonadjacent, and y has a neighbour in $A_{1,2,3}(x)$;
- for $1 \leq i \leq 3$, if x has a neighbour in $V(P_i)$, then y has no neighbour in $V(K_1^{\ell-2}(\alpha_i(x)))$; and
- for $1 \leq i \leq 3$, if x has no neighbour in $V(P_i)$, then for some $j \in \{1,2,3\} \setminus \{i\}$, y has no neighbour in $V(K_{\ell-3}^0(\beta_j(x)))$.

We need to prove some properties of distant pairs.

Lemma 4.5. *Let G be a graph with no long theta, and let K be a tidy shortest long near-prism in G, with constituent paths P_1, P_2, P_3. Let x, y be K-major, where y is distant from x. Then y has exactly two neighbours in $A_{1,2,3}(x)$ and they are adjacent.*

Proof. We begin with:

(1) *If x has a neighbour in each of $V(P_1), V(P_2), V(P_3)$ then the theorem holds.*

Suppose that x has a neighbour in each of $V(P_1), V(P_2), V(P_3)$. If y has a neighbour in each of $A_1(x)$, $A_2(x)$, $A_3(x)$, there is a long theta formed by three xy-paths with interiors in $A_1(x), A_2(x), A_3(x)$, a contradiction. So we may assume that y has no neighbour in $A_3(x)$. Suppose that y also has no neighbour in $A_2(x)$. By Lemma 4.1, y has a neighbour in one of $V(P_2), V(P_3)$, say $V(P_2)$. Let M be an induced xy-path with interior in $V(P_2)$. If y has a unique neighbour in $A_1(x)$, there is a long theta induced on $A_1(x) \cup A_3(x) \cup V(M)$. If y has two nonadjacent neighbours in $A_1(x)$, there is an induced $\alpha_1(x)a_1$-path R with interior in $A_1(x) \cup \{y\}$ containing y, and then there is a long theta induced on $V(R) \cup A_3(x) \cup V(M)$, a contradiction. So y has exactly two adjacent neighbours in $A_1(x)$ and the theorem holds.

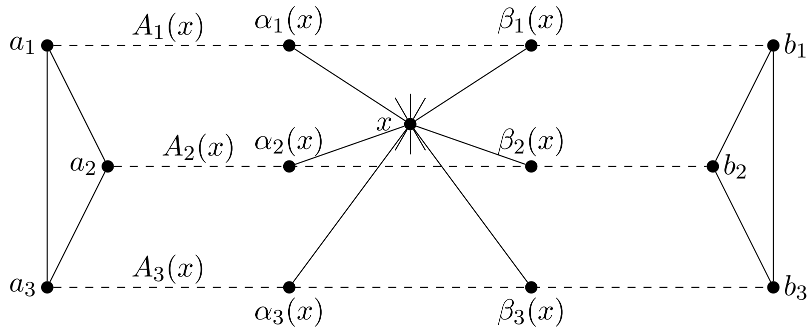

Figure 6: x has a neighbour in each of P_1, P_2, P_3 (possibly $\alpha_i(x) = \beta_i(x)$).

So we may assume that y has a neighbour in $A_1(x)$ and in $A_2(x)$, and not in $A_3(x)$. If y has two nonadjacent neighbours in $A_1(x)$, or two nonadjacent neighbours in $A_2(x)$, there is a long theta formed by three xy-paths all with interior in $A_{1,2,3}(x)$, a contradiction. So by Lemma 4.3, there exists $i \in \{1, 2, 3\}$ such that y has a neighbour in $V(P_i) \setminus A_i(x)$. This neighbour is nonadjacent to $\alpha_1(x), \alpha_2(x)$, obviously if $i = 3$ and from the definition of "distant" if $i \in \{1, 2\}$. Hence there is an induced xy-path R with interior in $(V(P_i) \setminus A_i(x)) \cup V(P_3)$, containing no neighbour of $\alpha_1(x)$ or $\alpha_2(x)$. But then there is a long theta formed by the path R and two xy-paths with interiors in $A_1(x), A_2(x)$ respectively, a contradiction. This proves (1).

We may therefore assume that x has no neighbour in $V(P_1)$, and hence $A_1(x) = V(P_1)$. By Lemma 4.3, x has two nonadjacent neighbours in one of $V(P_2), V(P_3)$, say in $V(P_j)$ where $j \in \{2, 3\}$; thus $\alpha_j(x), \beta_j(x)$ are distinct and nonadjacent. By Lemma 4.4, y has a neighbour in $V(P_i)$ for $i = 2, 3$.

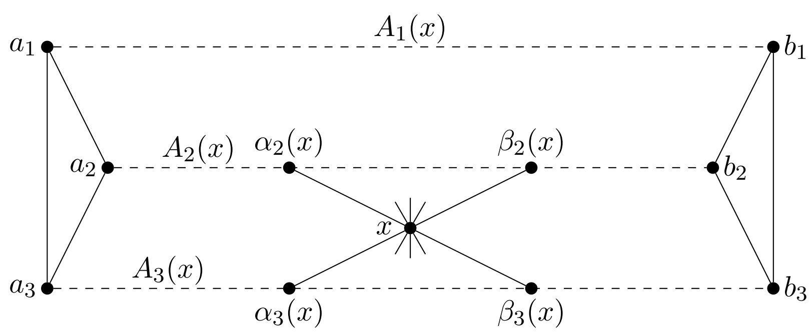

Figure 7: x has no neighbour in $V(P_1)$.

(2) *y does not have two nonadjacent neighbours in $A_2(x) \cup A_3(x)$.*

Suppose that it does; then there are two long xy-paths R_1, R_2, with $R_1^*, R_2^* \subseteq A_2(x) \cup A_3(x)$ and with R_1^* anticomplete to R_2^*. If y has a neighbour in $V(P_i) \setminus A_i(x)$ for some $i \in \{2, 3\}$, this neighbour is nonadjacent to $\alpha_i(x)$ from the definition of "distant"; so if y has a neighbour in $V(P_i) \setminus A_i(x)$ for some $i \in \{2, 3\}$, or a neighbour in $V(P_1)$, there is an xy-path with interior in $V(K) \setminus (A_2(x) \cup A_3(x))$ with interior anticomplete to R_1^*, R_2^*, and these three paths form a long theta, a contradiction. So every neighbour of y in $V(K)$ belongs to $A_2(x) \cup A_3(x)$. By Lemma 4.1, y has a neighbour in $A_2(x)$ and one in $A_3(x)$, and by Lemma 4.3 we may assume it has two nonadjacent neighbours in $V(P_2)$; but then there is a long theta formed by the paths

$$y\text{-}\beta_2(y)\text{-}P_2\text{-}\alpha_2(x)\text{-}x,$$

$$y\text{-}\beta_3(y)\text{-}P_3\text{-}\alpha_3(x)\text{-}x$$

$$y\text{-}\alpha_2(y)\text{-}P_2\text{-}a_2\text{-}a_1\text{-}P_1\text{-}b_1\text{-}b_j\text{-}P_j\text{-}\beta_j(x)\text{-}x,$$

a contradiction. This proves (2).

(3) *We may assume that y has a neighbour in $V(P_1)$.*

Suppose that y has no neighbour in $V(P_1)$. By definition of distant, y has at least one neighbour in $A_{1,2,3}(x)$. By (2) there is a two-vertex path of K containing all neighbours of y in $A_{1,2,3}(x)$. We may therefore assume that y has a unique neighbour $v \in A_{1,2,3}(x)$, because otherwise the theorem holds, and we may assume that $v \in A_2(x)$. Both x, y have a neighbour in $V(P_2 \cup P_3)$ that is not in $A_{2,3}(x)$ and has no neighbour in this set; and so there is an xy-path R with interior anticomplete to $A_{2,3}(x)$. But then there is a long theta formed by the paths

$$v\text{-}P_2\text{-}\alpha_2(x)\text{-}x,$$

$$v\text{-}y\text{-}R\text{-}x$$

$$v\text{-}P_2\text{-}a_2\text{-}a_3\text{-}P_3\text{-}\alpha_3(x)\text{-}x$$

a contradiction. This proves (3).

(4) *y has no neighbour in $A_2(x) \cup A_3(x)$.*

By (2), y has at most two neighbours in $A_2(x) \cup A_3(x)$. If y has two neighbours u, v in $A_2(x) \cup A_3(x)$, then they are adjacent and we may assume they belong to $A_2(x)$, and $a_2, u, v, \alpha_2(x)$ are in order in P_2. But then there is a long prism with bases $\{a_1, a_2, a_2\}$ and $\{y, u, v\}$ and constituent paths

$$y\text{-}\alpha_1(y)\text{-}P_1\text{-}a_1,$$

$$u\text{-}P_2\text{-}a_2$$

$$v\text{-}P_2\text{-}\alpha_2(x)\text{-}x\text{-}\alpha_3(x)\text{-}P_3\text{-}a_3,$$

and it is shorter than K, a contradiction. Thus y has at most one neighbour in $A_2(x) \cup A_3(x)$. If there is such a neighbour, say $v \in A_2(x)$, let M be an induced xy-path with interior in $V(\beta_1(y)\text{-}P_1\text{-}b_1\text{-}b_j\text{-}P_j\text{-}\beta_j(x))$; then there is a long theta formed by the paths

$$v\text{-}P_2\text{-}\alpha_2(x)\text{-}x,$$

$$v\text{-}P_2\text{-}a_2\text{-}a_3\text{-}P_3\text{-}\alpha_3(x)\text{-}x,$$

$$v\text{-}y\text{-}M\text{-}x,$$

a contradiction. This proves (4).

From the definition of "distant", we may assume that y has no neighbour in $V(K^0_{\ell-3}(\beta_2(x)))$. Since y has a neighbour in $V(P_1)$, there is an xy-path R_1 with interior in the vertex set of $\beta_1(y)$-P_1-b_1-b_2-P_2-$\beta_2(x)$ and $\beta_2(x) \in R_1^*$. It follows that R_1 is long.

By Lemma 4.4, y has a neighbour in $V(P_3)$ not in $A_3(x)$ and not adjacent to $\alpha_3(x)$; let R_3 be an induced xy-path with interior in $V(P_3)$, chosen with interior anticomplete to $\alpha_3(x)$ if $j = 3$. Let R_2 be the path y-$\alpha_1(y)$-P_1-a_1-a_j-P_j-$\alpha_j(x)$. If $\alpha_1(y)$ is distinct from and nonadjacent to $\beta_1(y)$, the three paths R_1, R_2, R_3 form a long theta, a contradiction. If $\alpha_1(y) = \beta_1(y)$, then the three paths $R_1 \setminus \{y\}$, $R_2 \setminus \{y\}$, $\alpha_1(y)$-y-R_3-x form a long theta, a contradiction. Thus $\alpha_1(y), \beta_1(y)$ are distinct and adjacent. This proves Lemma 4.5.

□

Lemma 4.6. *Let G be a graph with no long theta, and let K be a tidy shortest long near-prism in G, with constituent paths P_1, P_2, P_3. Let x, y, z be K-major, such that y, z are both distant from x and y, z are nonadjacent. For all distinct $i, j, k \in \{1, 2, 3\}$, either there is no yz-path of length at least $\ell - 2$ with interior in $A_{i,j}(x)$, or there is no yz-path of length at least $\ell - 2$ with interior in $V(P_k)$.*

Proof. Suppose that M_1 is a yz-path of length at least $\ell - 2$ with interior in $A_{i,j}(x)$, and M_2 is a yz-path of length at least $\ell - 2$ with interior in $V(P_k)$. By Lemma 4.5, y, z each have exactly two neighbours in $A_{i,j}(x)$ and they are adjacent. By Lemma 4.2, y, z each have a third neighbour in $V(P_i \cup P_j)$, and this neighbour does not belong to $A_{i,j}(x)$ and has no neighbour in $A_{i,j}(x)$, since (x, y) and (x, z) are distant. Consequently there is an induced yz-path M_3 with interior in $V(P_i \cup P_j)$ and anticomplete to $M_1^* \cup M_2^*$; and M_1, M_2, M_3 form a long theta, a contradiction. This proves Lemma 4.6.

□

Lemma 4.7. *Let G be a graph with no long theta, and let K be a tidy shortest long near-prism in G, with constituent paths P_1, P_2, P_3. Let x, y, z be K-major, such that y, z are both distant from x. If there exist $i, j \in \{1, 2, 3\}$ such that y has no neighbour in $V(P_i)$ and z has no neighbour in $V(P_j)$ then $i = j$.*

Proof. Suppose that y has no neighbour in $V(P_i)$ and z has no neighbour in $V(P_j)$, and $i \neq j$. By Lemma 4.4, y, z are adjacent; and also by Lemma 4.4 (applied to x, y and to x, z), x has a neighbour in each of $V(P_1), V(P_2), V(P_3)$. From Lemma 4.5, y, z each have exactly two neighbours in $A_{1,2,3}(x)$ and they are adjacent, and from the symmetry we may assume that y, z have no neighbour in $A_3(x)$. Let R be the path $\alpha_1(x)$-P_1-a_1-a_2-P_2-$\alpha_2(x)$, and let the neighbours of y in $V(R)$ be y_1, y_2, where $\alpha_1(x), y_1, y_2, \alpha_2(x)$ are in order in R. Define z_1, z_2 similarly. We may assume that $\alpha_1(x), y_1, z_2, \alpha_2(x)$ are distinct and in order in R. If the path y_2-R-z_1 has length at least $\ell - 3$, then there is a long prism with bases $\{y, y_1, y_2\}$, $\{z, z_1, z_2\}$, and constituent paths

$$y\text{-}z,$$

$$y_2\text{-}R\text{-}z_1,$$

$$y_1\text{-}R\text{-}\alpha_1(x)\text{-}x\text{-}\alpha_2(x)\text{-}R\text{-}z_2$$

and it is shorter than K, a contradiction. Thus y_2-R-z_1 has length at most $\ell-4$ and so $\{y_1,y_2,z_1,z_2\}$ is a subset of one of $A_1(x), A_2(x)$; and we may assume that $\{y_1,y_2,z_1,z_2\} \subseteq A_1(x)$. So y,z have no neighbours in $A_2(x)$ and no neighbours in $A_3(x)$, restoring the symmetry between P_2,P_3; and therefore we may assume that $i=3$ and $j=2$, that is, y has no neighbour in $V(P_3)$ and z has no neighbour in $V(P_2)$. By Lemma 4.1, y has a neighbour in $V(P_2)$ and z has a neighbour in $V(P_3)$.

If $y_1=z_1$ and hence $y_2=z_2$, there is a long prism with bases $\{a_1,a_2,a_3\}$, $\{y,z,y_2\}$ and constituent paths

$$y_2\text{-}P_1\text{-}a_1,$$

$$y\text{-}\alpha_2(y)\text{-}P_2\text{-}a_2$$

$$z\text{-}\alpha_3(z)\text{-}P_3\text{-}a_3,$$

and it is shorter than K, a contradiction. So $y_1 \neq z_1$, and therefore y_1,z_2 are nonadjacent. Then there is a long theta with constituent paths

$$z\text{-}y\text{-}y_1\text{-}P_1\text{-}\alpha_1(x)\text{-}x,$$

$$z\text{-}z_2\text{-}R\text{-}\alpha_2(x)\text{-}x,$$

and an induced xz-path with interior in $V(P_3)$, a contradiction. This proves Lemma 4.7.

□

5 Cleaning lightest long near-prisms

In this section we will complete the proof of Theorem 3.1, by showing how to compute a cleaning list for lightest long near-prisms.

Let $\mathcal{Q}$ be a set of paths of G, pairwise anticomplete. We define $V(\mathcal{Q})$ to be the union of the vertex sets of the members of $\mathcal{Q}$, and $\mathcal{Q}^*$ to be the union of the interiors of the member of $\mathcal{Q}$, and the *cost* of $\mathcal{Q}$ to be the cardinality of $V(\mathcal{Q})$.

Let K be a shortest long near-prism, with an ordered frame $\mathcal{F}$, and with constituent paths P_1,P_2,P_3, numbered according to $\mathcal{F}$. A K-major vertex x is $(K,\mathcal{F})$-*extremal* if either

- there is a K-major vertex with no neighbour in $V(P_1)$, and x is chosen with no neighbour in $V(P_1)$ and with $A_2(x)$ maximal; or
- every K-major vertex has a neighbour in $V(P_1)$, and x is chosen with $A_1(x)$ maximal.

Thus if x is $(K,\mathcal{F})$-extremal, and has a neighbour in $V(P_1)$, then every K-major vertex has a neighbour in $A_1(x)$; and otherwise $A_1(x)=V(P_1)$, and every K-major vertex has a neighbour in $V(P_1)\cup A_2(x)$. A $(K,\mathcal{F})$-*contrivance* consists of a quintuple $(x,y,\alpha,h,\mathcal{Q})$, where x,y are K-major (possibly $y=x$), and x is $(K,\mathcal{F})$-extremal, and $\mathcal{Q}$ is a set of paths of K, pairwise anticomplete, and $\alpha \in \mathcal{Q}^*$, and $h \in \{1,2\}$, such that:

- every K-major vertex is either adjacent to one of x,y or has a neighbour in $\mathcal{Q}^*$;
- if x has a neighbour in $V(P_1)$ then $h=1$ and $\alpha=\alpha_1(x)$, and otherwise $h=2$ and $\alpha=\alpha_2(x)$; and

- every neighbour of x or y in $A_{1,2}(x)$ belongs to $\mathcal{Q}^*$.

Its *cost* is the cost of $\mathcal{Q}$. From Lemma 4.6 we have:

Lemma 5.1. *Let G be a prospect, and let K be a tidy shortest long near-prism in G with an ordered frame $\mathcal{F}$, and with a K-major vertex. Then there is a $(K, \mathcal{F})$-contrivance with cost at most $6\ell - 4$.*

Proof. Let P_1, P_2, P_3 be the constituent paths of K. Choose x $(K, \mathcal{F})$-extremal, and let S be the set of all K-major vertices that are distant from x.

If x has a neighbour in $V(P_1)$ let $h = 1$ and $\alpha = \alpha_1(x)$, and otherwise let $h = 2$ and $\alpha = \alpha_2(x)$. If x has a neighbour in $V(P_i)$ for $i = 1, 2, 3$, let Q_i be the path $K_2^{\ell-1}(\alpha_i(x))$ for $i = 1, 2, 3$. If x has neighbours in $V(P_i), V(P_j)$ and not in $V(P_k)$, where $\{i, j, k\} = \{1, 2, 3\}$ and $i < j$, let Q_1 be the path $K_2^{\ell-1}(\alpha_i(x))$, let Q_2 be the path $K_2^{\ell-1}(\alpha_j(x))$, and let Q_3 be the path $K_{\ell-2}^1(\beta_i(x))$. Every K-major vertex has a neighbour in $A_{1,2,3}(x)$, since x is $(K, \mathcal{F})$-extremal; and so every K-major vertex nonadjacent to x either belongs to S or has a neighbour in one of Q_1^*, Q_2^*, Q_3^*, from the definition of "distant". If $S = \emptyset$, let $\mathcal{Q}$ be the set of components of the graph induced on the union of the vertex sets of Q_1, Q_2, Q_3; then $(x, x, \alpha, h, \mathcal{Q})$ is a $(K, \mathcal{F})$-contrivance satisfying the theorem, so we may assume that $S \neq \emptyset$.

If every vertex in S has a neighbour in $V(P_3)$, let $k = 3$, and otherwise let $k = 2$; then by Lemma 4.7, every vertex in S has a neighbour in $V(P_k)$. Choose $y \in S$ with $A_k(y)$ maximal, let Q_4 be the path $K_1^{\ell-4}(\alpha_k(y))$. By Lemma 4.5, y has exactly two neighbours in A_{12} and they are adjacent. Let Q_5 be a path of K of length $2\ell - 7$ such that the two neighbours of y in $A_{1,2}(x)$ are the two middle vertices of Q_5.

(1) *Every vertex in S nonadjacent to y has a neighbour in $Q_4^* \cup Q_5^*$.*

Let $z \in S$ be nonadjacent to y, and suppose it has no neighbour in $Q_4^* \cup Q_5^*$. From the choice of x, it follows that y, z both have a neighbour in $A_{1,2}(x)$, and so there is a yz-path M_1 with interior in $A_{1,2}(x)$; and M_1 has length at least $\ell - 2$ since z has no neighbour in Q_5^*. By Lemma 4.6, there is no yz-path of length at least $\ell - 2$ with interior in $V(P_3)$. Suppose that $k = 3$; then from the choice of y, there is a yz-path with interior in $A_3(y)$, which has length at least $\ell - 2$ since z has no neighbour in Q_4^*, a contradiction.

So $k = 2$, and therefore some vertex in $t \in S$ has no neighbour in $V(P_3)$. It follows from Lemmas 4.4 and 4.1 applied to t, x that x has a neighbour in $V(P_1)$ and a neighbour in $V(P_2)$. By applying Lemma 4.7 and Lemma 4.1 to t and z we obtain that z has a neighbour in $V(P_1)$ and in $V(P_2)$. It follows that z has a neighbour in $A_2(y)$ from the choice of y. Hence there is a yz-path M_2 with interior in $A_2(y)$, which has length at least $\ell - 2$ since z has no neighbour in Q_4^*. But x has a neighbour in $V(P_1)$, and therefore y, z both have neighbours in $A_1(x)$ since x is $(K, \mathcal{F})$-extremal, and so y, z have no neighbours in $A_2(x)$ by Lemma 4.5; and it follows that M_1 has interior in $V(P_1)$. This contradicts Lemma 4.6, (taking $i = 1$, $j = 3$ and $k = 2$). This proves (1).

Let $\mathcal{Q}$ be the set of components of the graph induced on the union of the vertex sets of $Q_1, \ldots, Q_5$; then $(x, y, \alpha, h, \mathcal{Q})$ is a $(K, \mathcal{F})$-contrivance satisfying the theorem. This proves Lemma 5.1.

□

If K is a tidy long near-prism, and $\mathcal{F}$ is an ordered frame for K, and the constituent paths of K are P_1, P_2, P_3 numbered according to $\mathcal{F}$, and x is K-major, let $L(x) = A_1(x)$ if x has a neighbour in $V(P_1)$, and $L(x) = V(P_1) \cup A_2(x)$ otherwise. If K is a tidy, lightest long near-prism, then a knowledge of the ordered frame $\mathcal{F}$ and of a $(K, \mathcal{F})$-contrivance $(x, y, \alpha, h, \mathcal{Q})$ allows us to reconstruct $L(x)$, as the next result shows:

Lemma 5.2. *Let G be a prospect, let K be a tidy lightest long near-prism in G, let $\mathcal{F}$ be an ordered frame of K, with frame F, and let $(x, y, \alpha, h, \mathcal{Q})$ be a $(K, \mathcal{F})$-contrivance. Let P_1, P_2, P_3 be the constituent paths of K, numbered according to $\mathcal{F}$, where P_i has ends a_i, b_i as usual. Let s_i, t_i be the a_i-end and b_i-end of F respectively, if they exist. Let Z_1 be the set of all vertices of G not in $V(\mathcal{Q})$ but with a neighbour in $\mathcal{Q}^*$, and let Z_2 be the set of all vertices adjacent to x or y that are not in $V(F)$ or in $V(\mathcal{Q})$. Let $G_1 = G \setminus (Z_1 \cup Z_2)$.*

- *If $h = 1$ (and therefore x has a neighbour in $V(P_1)$, and $\alpha = \alpha_1(x)$, and s_1 is defined), then s_1-P_1-α is the lightest $s_1\alpha$-path in G_1.*
- *Assume that $h = 2$ (and so x has no neighbour in $V(P_1)$, and $\alpha = \alpha_2(x)$, and s_2 is defined). If s_1 is not defined, then P_1 is the a_1b_1-path in $F \setminus \{a_2, a_3, b_2, b_3\}$, and s_2-P_2-$\alpha_2(x)$ is the lightest $s_2\alpha$-path in G_1. If s_1 is defined, then s_1-P_1-t_1 is the lightest s_1t_1-path in G_1, and s_2-P_2-$\alpha_2(x)$ is the lightest $s_2\alpha$-path in G_2, where G_2 is obtained from G_1 by deleting all vertices that belong to or have a neighbour in $V(s_1$-P_1-$t_1)$.*

Proof. To prove the first bullet, we assume that $h = 1$, and so x has a neighbour in $V(P_1)$, and therefore s_1, t_1 are defined, and $\alpha = \alpha_1(x)$. The path s_1-P_1-$\alpha_1(x)$ is the lightest $s_1\alpha$-path in G_1 by Lemma 3.4. This proves the first bullet.

For the second bullet, we assume that $h = 2$, and so x has no neighbour in $V(P_1)$, and therefore x has a neighbour in $V(P_2)$ by Lemma 4.1; and so s_2, t_2 are defined and $\alpha = \alpha_2(x)$. If s_1 is not defined, then P_1 is a path of F as claimed, and s_2-P_2-$\alpha_2(x)$ is the lightest $s_2\alpha$-path in G_1 by Lemma 3.4. So we assume that s_1, t_1 are defined. Then, by Lemma 3.4, s_1-P_1-t_1 is the lightest s_1t_1-path in G_1. Similarly s_2-P_2-$\alpha_2(x)$ is a lightest $s_2\alpha_2(x)$-path in G_2 (though not necessarily in G_1) by Lemma 3.4. This proves the second bullet and so proves Lemma 5.2.

□

Thus, if there is a lightest long near-prism K, with a given ordered frame $\mathcal{F}$ and a given $(K, \mathcal{F})$-contrivance $(x, y, \alpha, h, \mathcal{Q})$, we can reconstruct $L(x)$ algorithmically, using the construction of Lemma 5.2, in time $\mathcal{O}(|G|^3)$. More exactly, if $h = 1$, then the first bullet of Lemma 5.2 gives a method to compute $A_1(x) = L(x)$. If $h = 2$, we first compute P_1 using the method of the second bullet of Lemma 5.2; then compute G_2; and then compute $A_2(x)$, again using the method of the second bullet of Lemma 5.2. In summary:

Lemma 5.3. *There is an algorithm with the following specifications:*

Input: *A prospect G, a linear order of the edges of G, an ordered frame $\mathcal{F}$ in G, and a quintuple $(x, y, \alpha, h, \mathcal{Q})$ where $x, y, \alpha \in V(G)$ and $\mathcal{Q}$ is a set of pairwise anticomplete induced paths of G.*

Output: *A subgraph L of G, such that if there is a long near-prism in G, and the lightest long near-prism K is tidy and has ordered frame $\mathcal{F}$ and $(x, y, \alpha, h, \mathcal{Q})$ is a $(K, \mathcal{F})$-contrivance, then $L = L(x)$.*

Running time: $\mathcal{O}(|G|^3)$.

The good thing about having reconstructed $L(x)$ is that every K-major vertex has a neighbour in $L(x)$, either in the interior of the path $G[L(x)]$ or in $\mathcal{Q}^*$; and no vertices not in $V(K) \setminus L(x)$ have such a neighbour, so now we can clean the K-major vertices. More exactly, let Z_4 be the set of all vertices of G that are not in $V(F) \cup V(\mathcal{Q}) \cup V(L(x))$ and have a neighbour either in $\mathcal{Q}^*$ or in the interior of a path of $G[L(x)]$; then $Z_4 \cup V(K) = \emptyset$ and every K-major vertex belongs to Z_4. We obtain:

Lemma 5.4. *There is an algorithm with the following specifications:*

Input: *A prospect G, a linear order of the edges of G, an ordered frame $\mathcal{F}$ in G, and a quintuple $(x, y, \alpha, h, \mathcal{Q})$ where $x, y, \alpha \in V(G)$ and $h \in \{1, 2\}$, and $\mathcal{Q}$ is a set of pairwise anticomplete induced paths of G.*

Output: *A subset $X \subseteq V(G)$, such that if there is a long near-prism in G, and the lightest long near-prism K is tidy and has ordered frame $\mathcal{F}$ and $(x, y, \alpha, h, \mathcal{Q})$ is a $(K, \mathcal{F})$-contrivance, then X contains all K-major vertices and is disjoint from $V(K)$.*

Running time: $\mathcal{O}(|G|^3)$.

We can now prove Theorem 3.1, which we restate:

Theorem 5.5. *There is an algorithm with the following specifications:*

Input: *A prospect G.*

Output: *Decides whether G contains a long near-prism.*

Running time: $\mathcal{O}(|G|^{9\ell+1})$.

Proof. Fix a linear order of the edges of G. Enumerate all ordered frames $\mathcal{F}$ in G. For each one, let $\mathcal{F}$ have frame F, and compute G_1, the graph obtained from G by deleting all vertices not in F^* but with a neighbour in F^*, except the ends of F. Compute the linear order of $E(G_1)$ induced from the given linear order of $E(G)$. Compute all quintuples $(x, y, \alpha, h, \mathcal{Q})$ where $x, y \in V(G_1)$, and $\alpha \in \mathcal{Q}^*$, and $h \in \{1, 2\}$, and $\mathcal{Q}$ is a set of pairwise anticomplete induced paths of G_1 with cost at most $6\ell - 4$. Apply the algorithm of Lemma 5.4 to G_1, the linear order of $E(G_1)$, $\mathcal{F}$ and $(x, y, \alpha, h, \mathcal{Q})$, to obtain a set $X \subseteq V(G_1)$. Apply the algorithm of Theorem 3.5 to $G_1 \setminus X$, the induced linear order of its edge set, and the given frame. If this tells us that G_1 has a long near-prism, output this and stop. If after examining all choices of $(x, y, \alpha, h, \mathcal{Q})$ we have not found a long near-prism, move to the next ordered frame; and if after examining all ordered frames we have not found a long near-prism, report that there is none.

There are only at most 3ℓ vertices in a frame, and so only $\mathcal{O}(|G|^{3\ell})$ different ordered frames to examine. For each one, there are only $\mathcal{O}(|G|^{6\ell-2})$ different quintuples $(x, y, \alpha, h, \mathcal{Q})$ to check, since $\mathcal{Q}$ has cost at most $6\ell - 4$ and there are only at most $6\ell - 4$ choices for α. For each choice of the quintuple, applying the algorithm of Lemma 5.4 takes time $\mathcal{O}(|G|^3)$, and then applying the algorithm of Theorem 3.5 takes time $\mathcal{O}(|G|^3)$. So the total running time is $\mathcal{O}(|G|^{9\ell+1})$.

For correctness, certainly if the algorithm reports a long near-prism then this is correct. To check the converse, suppose that G contains a long near-prism, and let K be the lightest long near-prism. Let $\mathcal{F}$ be an ordered frame for K. Since K has a tidy frame in G_1, Lemma 5.1 implies that there is a

$(K,\mathcal{F})$-contrivance $(x,y,\alpha,h,\mathcal{Q})$ in G_1, where $\mathcal{Q}$ has cost at most $6\ell-2$. When the algorithm checks this ordered frame and this quintuple, the algorithm of Lemma 5.4 outputs a set X that contains all K-major vertices and does not intersect $V(K)$; so K is clean in $G_1 \setminus X$. The algorithm of Theorem 3.5, applied to $G_1 \setminus X$ cannot output that there is no long near-prism that is the lightest among all long near-prisms, and has ordered frame $\mathcal{F}$, and is clean, because there is one. Thus it will output that G_1 contains a long near-prism. This proves correctness, and so proves Theorem 5.5.

$\square$

6 Detecting a clean lightest long even hole

Let us say a graph G is a *candidate* if it contains no long even hole of length at most 2ℓ, no long jewel of order at most $\ell+1$, no long theta, no long ban-the-bomb, and no long near-prism. Thus, candidates are prospects.

Let C be a hole in a graph G. We recall that a vertex $x \in V(G) \setminus V(C)$ is *C-major* if no three-vertex path of C contains all the neighbours of x in $V(C)$, and C is *clean* if there is no C-major vertex. In this section we provide an algorithm to detect a clean lightest long even hole in a candidate if there is one. We begin with:

Lemma 6.1. *Let G be a candidate, and let C be a shortest long even hole in G, and let x be C-major. Then x has three pairwise nonadjacent neighbours in $V(C)$, and for every three-vertex path Q of C, x has at least two neighbours in $V(C) \setminus V(Q)$.*

Proof. Since G is a candidate it follows that C has length at least $2\ell+2$. If x has at least five neighbours in $V(C)$ then both claims are true, so we assume that x has at most four neighbours in $V(C)$, say $v_1,\ldots,v_k$ in order, where $2 \le k \le 4$. If $k=2$ let P_1,P_2 be the two v_1v_2-paths of C, and if $k \in \{3,4\}$ let P_i be the v_iv_{i+1}-path of C not containing v_{i+2} for $1 \le i \le k$ (reading subscripts modulo k).

(1) *For $1 \le i \le k$, if P_i has length at least $\ell-2$ then P_i is odd, and the path $C \setminus P_i^*$ has length at least $\ell+2$.*

If P_i has length at least $\ell-2$, then (reading subscripts modulo k) the hole x-v_i-P_i-v_{i+1}-x is long and shorter than C, and therefore odd, and so P_i is odd. Consequently $C \setminus P_i^*$ is also odd, since C is even; and hence the paths $C \setminus P_i^*$, v_1-x-v_2 and v_1-P_i-v_2 form a long jewel, which therefore has order at least $\ell+2$, that is, $C \setminus P_i^*$ has length at least $\ell+1$. This proves (1).

Let P_1 be the longest of $P_1,\ldots,P_k$. If $k=2$, then P_1 is long, and so (1) implies that the paths P_1, P_2 and v_1-x-v_2 form a long theta, a contradiction, so $k \ge 3$. Suppose that $k=3$. If P_2,P_3 both have length at least three then both claims are true, so we may assume that P_2 has length at most two. So P_1 is long, and hence so is P_3, by (1), and therefore they are both odd, by (1) again. Thus P_2 is even, and so has length two, and hence $G[V(C) \cup \{x\}]$ is a long ban-the-bomb, a contradiction. This proves that $k=4$.

If two of P_2,P_3,P_4 have length at least two, then both claims are true; so we may assume that P_2 has length one, and one of P_3,P_4 has length one, and therefore P_1 is long. Now there are two cases. If P_3 has length one then P_4 is long, by (1), and so $G[V(C) \cup \{x\}]$ is a long ban-the-bomb, a contradiction; and if P_4 has length one then P_3 is long, by (1), and $G[V(C) \cup \{x\}]$ is a long near-prism, a contradiction. This proves Lemma 6.1.

$\square$

Let C be a shortest long even hole. For u, v distinct and nonadjacent vertices in $V(C)$ we call a uv-path Q a *shortcut* if $V(Q)$ contains no C-major vertices and Q has length less than $d_C(u, v)$. We begin by proving the following.

Theorem 6.2. *Let G be a candidate and let C be a shortest long even hole in G. Then C has no shortcut.*

Proof. Suppose that G has a shortest long even hole C with a shortcut Q. Thus $|E(C)| \geq 2\ell + 2$, since G is a candidate. Choose C, Q to minimize $|E(Q)|$, and subject to that, to maximize $d_C(u, v)$, where u, v are the ends of Q. It follows that $Q^* \cap V(C) = \emptyset$. Let Q have vertices u-q_1-q_2-$\cdots$-q_k-v in order. It follows that Q has length $k + 1$, and so $d_C(u, v) \geq k + 2$. Consequently $k > 1$, since Q contains no C-major vertices.

(1) *The set of neighbours of q_1 in $V(C)$ is a clique, and the same holds for q_k, and q_1, q_k have no common neighbour in $V(C)$.*

Suppose that q_1 has two nonadjacent neighbours in $V(C)$, say x, y. Since q_1 is not C-major, there is a vertex z of C such that x-z-y is a path of C, and every neighbour of q_1 in $V(C)$ is one of x, y, z. Let C' be the hole induced on $(V(C) \setminus \{z\}) \cup \{q_1\}$. Then C' has the same length as C, and so is a shortest even hole, and $d_{C'}(q_1, v) = d_C(z, v) \geq d_C(u, v) - 1$. Let $Q' = Q \setminus \{u\}$. From the choice of C, Q it follows that Q' is not a shortcut for C', and so some vertex of Q' is C'-major, and hence is adjacent to q_1. Consequently q_2 is C'-major, and yet all its neighbours in $V(C')$ except q_1 lie in a three-vertex path of C and hence of C', contrary to Lemma 6.1. This proves the first assertion of (1). The second follows since $d_C(u, v) \geq k + 2 \geq 4$. This proves (1).

(2) *If $1 \leq i \leq k$ and q_i is adjacent to $w \in V(C) \setminus \{u, v\}$, then $d_C(u, w) = |E(R)|$, where R is the uw-path of $C \setminus \{v\}$. The same holds with u, v exchanged.*

Suppose not; then the shorter of the two uw-paths of C strictly includes one of the uv-paths of C, and so has length more than $d_C(u, v)$, contradicting the choice of C, Q. This proves (2).

(3) *One of $q_2, \ldots, q_{k-1}$ has a neighbour in $V(C)$.*

Suppose not. By (1), q_1 either has one, or two adjacent, neighbours in $V(C)$, and the same for q_k. There are two minimal paths R_1, R_2 of C with one end adjacent to q_1 and the other to q_k, and since the sum of their lengths is at least $|E(C)| - 2 \geq 2\ell$, we may assume that R_2 is long. Let the ends of R_2 be u', v' where u' is adjacent to q_1 and v' to q_k. Let S be the $u'v'$-path of C different from R_2, and let Q' be the path u'-q_1-Q-q_k-v'. Now Q' has the same length as Q, and therefore less than $d_C(u, v) \leq |E(S)|$. Consequently the hole $Q' \cup R_2$ has length less than C, and it is long and therefore odd. So Q', S have different parity. If R_1 is not long, then S has length at most $\ell + 1$, and so does Q, and hence the paths S, Q', R_2 form a long jewel of order at most $\ell + 1$, a contradiction. So R_1 is long.

Not both q_1, q_k have a unique neighbour in $V(C)$, since G contains no long theta, and they do not both have two adjacent neighbours, since G contains no long near-prism. Thus we may assume that q_1 has two adjacent neighbours x, u' in $V(C)$, and q_k has exactly one (namely $v = v'$). Since Q', S have different parity, it follows that the hole x-q_1-Q-v-R_1-x is long, even, and shorter than C, a contradiction. This proves (3).

Let L_1 be a uv-path of C such that one of $q_2, \ldots, q_{k-1}$ has a neighbour in $V(L_1)$, and let L_2 be the other uv-path of C.

(4) *Q^* is anticomplete to L_2^*.*

Choose $i \in \{2, \ldots, k-1\}$ such that q_i has a neighbour w_1 in $V(L_1)$, and suppose that there exists $j \in \{1, \ldots, k\}$ such that q_j has a neighbour w_2 in $V(L_2)$. By exchanging u, v if necessary, we may assume that $i \leq j$. See figure 8.

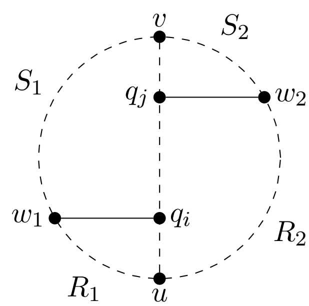

Figure 8: For step (4).

From the choice of C, Q it follows that $w_1, w_2 \neq u, v$. For $i = 1, 2$, let R_i be the uw_i-path of $C \setminus \{v\}$, and let S_i be the vw_i-path of $C \setminus \{u\}$. Let R_i, S_i have length r_i, s_i for $i = 1, 2$. Each of the paths u-q_1-q_i-w_1, w_1-q_i-$\cdots$-q_j-w_2, w_2-q_j-q_k-v is strictly shorter than Q (because $2 \leq i \leq k-1$), and hence none of them is a shortcut. From (2), it follows that $r_1 = d_C(u, w_1) \leq i + 1$, and $d_C(w_1, w_2) \leq j - i + 2$, and $s_2 \leq k - j + 2$. Adding, we deduce that

$$r_1 + d_C(w_1, w_2) + s_2 \leq (i+1) + (j-i+2) + (k-j+2) = k + 5 = |E(Q)| + 4.$$

But $d_C(w_1, w_2) = \min(r_1 + r_2, s_1 + s_2)$. Suppose that $d_C(w_1, w_2) = r_1 + r_2$. It follows that $r_1 + (r_1 + r_2) + s_2 \leq |E(Q)| + 4$, but $r_2 + s_2 > |E(Q)|$ since Q is a shortcut, and so $r_1 \leq 1$, and therefore $d_C(w_1, v) \geq d_C(u, v) - 1$. But $i > 1$, and so w_1-q_i-$\cdots$-q_k-v is a shortcut for C, contradicting the choice of C, Q. Thus $d_C(w_1, w_2) = s_1 + s_2 < r_1 + r_2$.

Hence $r_1 + (s_1 + s_2) + s_2 \leq |E(Q)| + 4$. But $r_1 + s_1 \geq d_C(u, v) > |E(Q)|$, and so $s_2 = 1$. Since u-Q-q_j-w_2 is not a shortcut for C that is shorter than Q, it follows that $j = k$. Since $r_1 \leq i + 1$ and

$$s_1 + 1 = d_C(w_1, w_2) \leq j - i + 2 = k - i + 2,$$

we deduce (adding) that $r_1 + s_1 \leq k + 2$. But $r_1 + s_1 > |E(Q)| = k + 1$, and so equality holds; that is, $r_1 = i + 1$ and $s_1 = k - i + 1$, and $r_1 + s_1 = |E(Q)| + 1$. Hence $r_1 + s_1 \leq r_2 + s_2$; and since $d_C(u, w_2) \le d_C(u, v)$ (from the choice of C, Q, since otherwise u-Q-q_k-w_2 would be a shortcut for C contrary to the choice of u, v), it follows that $r_2 + s_2 = r_1 + s_1 = |E(C)|/2$. Moreover, we showed that $q_j = q_k$ and w_2 is adjacent to v, on the assumption that $i \le j$; and it follows from the symmetry that the only edges between Q^* and L_2^* are the edge $q_k w_2$ and possibly an edge from q_1 to the neighbour (w_3) say of u in L_2, say w_3. If the latter edge does not exist, then u-Q-q_k-w_2-L_2-u is an even hole, of length $|E(C)| - 2$, a contradiction; so q_1 is adjacent to w_3. We already showed that $d_C(w_1, w_2) = s_1 + 1$, and it follows by the same argument with u, v exchanged that $d_C(w_1, w_3) = r_1 + 1 = i + 2$, and so the path w_3-q_1-$\cdots$-q_i-w_1 is a shortcut for C, a contradiction. This proves (4).

(5) $|E(L_1)| = |E(Q)| + 1 \le |E(L_2)|$.

Choose $i \in \{2, \ldots, k-1\}$ such that q_i has a neighbour $w \in L_1^*$. Since u-Q-q_i-w and w-q_i-$\cdots$-Q-v are not shortcuts for C, it follows from (2) that the sum of their lengths is at least $|E(L_1)|$, and so $|E(L_1)| \le |E(Q)| + 2$. Since one of L_1, L_2 is long (because $|E(C)| \ge 2\ell$), it follows that the hole $Q \cup L_2$ is long, and shorter than C, and therefore odd; and so Q, L_2 have opposite parity. Since L_1, L_2 have the same parity, and $|E(L_1)| > |E(Q)| \ge |E(L_1)| - 2$, we deduce that $|E(L_1)| = |E(Q)| + 1$. Since $|E(L_2)| > |E(Q)| = |E(L_1)| - 1$ it follows that $|E(L_1)| \le |E(L_2)|$. This proves (5).

By (5), we may number the vertices of L_1 as u-c_1-$\cdots$-c_{k+1}-v in order.

(6) *For $1 \le i \le k$, if q_i is adjacent to c_j where $1 \le j \le k+1$, then $j \in \{i, i+1\}$.*

If $i \in \{1, k\}$ this is true since q_1, q_k are not C-major, so we may assume that $2 \le i \le k-1$. The path u-Q-q_i-c_j has length $i + 1$, shorter than Q, and so is not a shortcut; and hence by (2), $j = d_C(u, c_j) \le i + 1$. Since c_j-q_i-Q-v is not a shortcut, it follows that $k + 2 - j = d_C(v, c_j) \le k + 2 - i$, and so $i \le j$. This proves (6).

By (3), (4) and (6), there exists $i \in \{2, \ldots, k-1\}$ such that q_i is adjacent to one of c_i, c_{i+1}, and by exchanging u, v if necessary, we may assume that q_i is adjacent to c_i. By (6),

$$u\text{-}c_1\text{-}\cdots\text{-}c_i\text{-}q_i\text{-}Q\text{-}v\text{-}L_2\text{-}u$$

is a hole C' say. Since the paths c_i-c_{i+1}-$\cdots$-c_{k+1}-v and c_i-q_i-$\cdots$-q_k-v have the same length, it follows that C' has the same length as C, and so is a shortest long even hole. From the choice of C, Q, the path u-q_1-$\cdots$-q_i is not a shortcut for C'. But its length is $i < d_{C'}(u, q_i)$, and so one of its vertices is C'-major. Hence there exists $h \in \{1, \ldots, i-1\}$ such that q_h is C'-major and not C-major, and so q_h has a neighbour in $\{q_i, \ldots, q_k\}$. But q_h is nonadjacent to $\{q_{i+1}, \ldots, q_k\}$, and therefore $h = i - 1$, so q_{i-1} is C'-major. By Lemma 6.1, at least two neighbours of q_{i-1} in $V(C)$ are not in $\{c_{i-1}, c_i, q_i\}$, contrary to (4) and (6). This proves Theorem 6.2.

□

We will also need:

Theorem 6.3. *Let C be a clean shortest long even hole in a candidate G. Let u, v be distinct, nonadjacent vertices in $V(C)$ with $d_C(u, v) \le |E(C)|/2 - 2$, and let L_1, L_2 be the two uv-paths of C*

where $|E(L_1)| \leq |E(C)|/2-2$. Then $P \cup L_2$ is a shortest long even hole for every shortest uv-path P in G.

Proof. Let P be a shortest uv-path in G, with vertices u-p_1-$\cdots$-p_k-v. Since C is clean, it follows from Theorem 6.2 that P has the same length as L_1. Suppose that for some $i \in \{1,\ldots,k\}$, p_i is equal or adjacent to some $w \in L_2^*$. By Theorem 6.2, the path u-p_1-$\cdots$-p_i-w (or u-p_1-$\cdots$-p_i if $w = p_i$) is not a shortcut for C, and so $i+1 \geq d_C(u,w)$. Since $i+1 \leq k+1 = |E(L_1)|$ it follows that the shorter uw-path of C is a subpath of L_2, and hence $i+1 \geq d_C(u,w) = d_{L_2}(u,w)$. Similarly $k-i+2 \geq d_{L_2}(w,v)$. Consequently

$$|E(P)|+2 = k+3 \geq d_{L_2}(u,w)+d_{L_2}(w,v) \geq d_{L_2}(u,v) = |E(L_2)| \geq |E(L_1)|+4,$$

a contradiction. This proves Theorem 6.3.

□

This can be strengthened: it is shown in [13] that

Theorem 6.4. *Let C be a clean shortest long even hole in a candidate G. Let u,v be distinct, nonadjacent vertices in $V(C)$, and let L_1, L_2 be the two uv-paths of C where $|E(L_1)| \leq |E(L_2)|$. Then for every shortest uv-path P in G, either $P \cup L_2$ is a clean shortest long even hole in G, or $|E(L_1)| = |E(L_2)|$ and $P \cup L_1$ is a clean shortest long even hole in G.*

We will not need this stronger form, however, so we omit it here.

Let us fix a linear order of the edges of G; then we can search for a lightest long even hole, instead of just a shortest one, and it is easier to find if it exists. For instance, from Theorem 6.3 we obtain

Theorem 6.5. *Let C be a lightest long even hole in a candidate G. Let u,v be distinct, nonadjacent vertices in $V(C)$ with $d_C(u,v) \leq |E(C)|/2-2$, and let L_1, L_2 be the two uv-paths of C where $|E(L_1)| \leq |E(C)|/2-2$. Then L_1 is the lightest uv-path in G that contains no C-major vertices.*

Proof. Let P be the lightest uv-path in G that contains no C-major vertices, and let G' be the graph obtained from G by deleting all C-major vertices. Thus P is the lightest uv-path in G'. But C is clean in G', and so by Theorem 6.3, $P \cup L_2$ is a shortest long even hole. It cannot be lighter than C, and so P is not lighter than L_1. On the other hand L_1 is not lighter than P, since P is the lightest uv-path in G'. Hence $P = L_1$. This proves Theorem 6.5.

□

Now the main result of the section:

Theorem 6.6. *There is an algorithm with the following specifications:*

Input: *A candidate G, and a linear ordering of $E(G)$.*

Output: *Decides either that G has a long even hole or that there is no clean lightest long even hole in G.*

Running time: *$\mathcal{O}(|G|^4)$.*

Proof. For all distinct $u, v \in V(G)$, compute a lightest uv-path $Q(uv) = Q(vu)$, and compute the set $N(uv)$ of all vertices that belong to or have a neighbour in $Q(uv)^*$. Enumerate all triples (v_1, v_2, v_3) of distinct vertices in G, and check whether

$$Q(v_1v_2) \cup Q(v_2v_3) \cup Q(v_3v_1)$$

is a long even hole, and if so, report this and stop. If all triples are examined without success, report that G contains no clean lightest long even hole. That concludes the description of the algorithm.

Each triple can be handled in time $\mathcal{O}(|G|)$ (by using the sets $N(uv)$), and so the total running time is $\mathcal{O}(|G|^4)$.

To prove correctness, let C be a clean lightest long even hole in G; we must show that there is a triple (v_1, v_2, v_3) for which the algorithm will find a long even hole. Since C has length at least 12 and hence $|E(C)| \leq 3(|E(C)|/2 - 2)$, there exist $v_1, v_2, v_3 \in V(C)$ such that each pair of vertices in this triple is joined by a path of C of length at most $|E(C)|/2 - 2$ that does not contain the third vertex in the triple. By Theorem 6.5 $Q(v_1v_2), Q(v_2v_3)$ and $Q(v_3v_1)$ are all paths of C and they have union C. This proves Theorem 6.6.

□

7 Cleaning a shortest long even hole

Our method of cleaning is very much like that used for shortest long near-prisms, and the next result is an analogue of Lemma 4.5. Let C be a shortest long even hole in a candidate G. For a C-major vertex x, we call a path P of C of length at least two a *(C, x)-gap* if both ends of P are neighbours of x and no interior vertex of P is adjacent to x. Thus, adding x to P yields a hole.

We begin with:

Lemma 7.1. *Let C be a shortest long even hole in G, and let x, y be nonadjacent C-major vertices. Let P be a (C, x)-gap of length at least $\ell - 2$, with ends p_1, p_2. If y has a neighbour in $V(P)$, then either*

- *for some $i \in \{1, 2\}$, some neighbour v of y in $V(P)$ satisfies $d_P(p_i, v) \leq \ell - 5$; or*
- *for some $i \in \{1, 2\}$, y is adjacent to a neighbour of p_i in C; or*
- *y has exactly two neighbours in $V(P)$ and they are adjacent.*

Proof. Let Q be the p_1p_2-path of C different from P. Thus Q has length at least three. The hole x-p_1-P-p_2-x is long and shorter than C, and so odd, and hence P, Q are odd. Let R be the graph obtained from Q by deleting its first two and last two vertices. We may assume that the first two bullets of the theorem are false.

(1) *x and y each have a neighbour in $V(R)$.*

By Lemma 6.1, x has a neighbour in $V(R)$. Suppose that y does not. Since the first two bullets of the theorem are false, it follows that all neighbours of y in $V(C)$ belong to P^*. Let P' be the induced p_1p_2-path with interior in $V(P) \cup \{y\}$ that contains y. Hence P' is shorter than P because y is K-major. Since the first bullet of the theorem is false, it follows that P' has length at least

$2(\ell-4)+2 \geq \ell$, and so the hole x-p_1-P'-p_2-x is long and shorter than C, and so odd. Hence P' is odd; but Q is also odd, and $P' \cup Q$ is a long even hole shorter than C, a contradiction. This proves (1).

By (1), there is an induced xy-path with interior in $V(R)$, say M. By hypothesis, y has at least one neighbour in $V(P)$. If y has only one neighbour v in $V(P)$, then there is a long theta formed by the two xv-paths with interior in $V(P)$ and x-M-y-v, (because the first two paths both have length at least $\ell-3$, and the third has length at least three) a contradiction. If y has two nonadjacent neighbours in $V(P)$, there is a long theta formed by the two induced xy-paths with interior in $V(P)$ and M, again a contradiction. Hence y has exactly two neighbours in $V(P)$ and they are adjacent. This proves Lemma 7.1.

□

Let C be a shortest long even hole. A *C-contrivance* is a six-tuple $(x, y, p_1, p_2, m, \mathcal{Q})$, where

- x, y are C-major vertices (possibly $y = x$), and there is a (C, x)-gap P with ends p_1, p_2 and midpoint m such that every C-major vertex has a neighbour in $V(P)$;
- $\mathcal{Q}$ is a set of paths of C, pairwise anticomplete;
- every neighbour of x or y in $V(P)$ belongs to $\mathcal{Q}^*$; and
- x, y and all C-major vertices nonadjacent to both x, y have a neighbour in $\mathcal{Q}^*$.

Its *cost* is the number of vertices in $V(\mathcal{Q})$. (For clarity, we will restate our definition of midpoint. For a path P with ends a, b, we call $v \in V(P)$ a *midpoint* of P if $|d_P(v, a) - d_P(v, b)| \leq 1$.)

These objects will be the analogue of $(K, \mathcal{F})$-contrivances, and we will use them in the same way. The next result is an analogue of Lemma 5.1.

Lemma 7.2. *Let G be a candidate and let C be a shortest long even hole in G such that for some C-major vertex x, there is a (C, x)-gap. Then there is a C-contrivance with cost at most $4\ell - 4$.*

Proof. Choose a maximal path P of C such that there is a C-major vertex x for which P is a (C, x)-gap. Let P have ends p_1, p_2, and let m be a midpoint of P. It follows that every C-major vertex has a neighbour in $V(P)$. For $i \in \{1, 2\}$ let Q_i be the path of C whose vertex set consists of all vertices of $V(P)$ with P-distance at most $\ell - 4$ from p_i and the two vertices of $V(C) \setminus V(P)$ with C-distance at most two from p_i.

Let S be the set of all C-major vertices with no neighbour in $Q_1^* \cup Q_2^* \cup \{x\}$. We may assume that $S \neq \emptyset$, because otherwise $(x, x, p_1, p_2, m, \mathcal{Q})$ is a C-contrivance satisfying the theorem, where Q is the set of components of $G[V(Q_1 \cup Q_2)]$. Hence Q_1, Q_2 do not share a vertex in $V(P)$. Since x is C-major, it follows that Q_1 and Q_2 are vertex-disjoint. For each $y \in S$, we define P_y to be the (C, y)-gap with $p_1 \in P_y^*$. Choose $y \in S$ with $|E(P_y) \setminus E(P)|$ maximum, and let p_3, p_4 be the ends of P_y, where $p_3 \notin V(P)$. (By Lemma 7.1, one end of P_y is not in $V(P)$.) Since y has no neighbour in $Q_1^* \cup Q_2^* \cup \{x\}$ and y has at least one neighbour in $V(P)$, it follows from Lemma 7.1 that y has exactly two neighbours in $V(P)$ and they are adjacent. One of them is p_4; let the other be p_5.

Let R denote the path p_1-P_y-p_3. For $i = 3, 4$, let Q_i be the path of C whose vertex set consists of all vertices of $V(P_y)$ with P_y-distance at most $\ell - 4$ from p_i and the two vertices of $V(C) \setminus V(P_y)$ with C-distance at most two from p_i. Then $p_5 \in Q_4^*$.

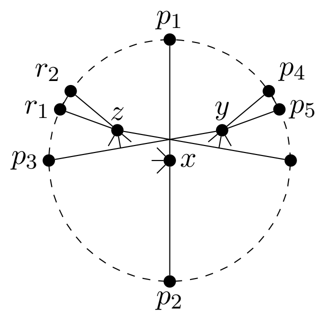

Figure 9: For Lemma 7.2.

(1) *Every C-major vertex has a neighbour in $Q_1^* \cup Q_2^* \cup Q_3^* \cup Q_4^* \cup \{x, y\}$.*

Suppose that z is C-major and has no neighbour in this set. Thus $z \neq x, y$. Since z has a neighbour in $V(R)$ from the choice of y, it follows from Lemma 7.1 applied to z and P_y that z has exactly two neighbours in $V(P_y)$ and they are adjacent, say r_1, r_2. Since z has a neighbour in $V(R)$ and z is not adjacent to p_1, it follows that $r_1, r_2 \in V(R)$. Number them so that p_3, r_1, r_2, p_1 are in order in P_y. Since z has a neighbour in $V(P)$, and no neighbour in p_1-P-p_4, there is a zp_5-path M with interior in the vertex set of p_5-P-p_2. But then there is a long prism with bases $\{y, p_4, p_5\}$, $\{z, r_1, r_2\}$, and constituent paths M, y-p_3-P_y-r_1 and r_2-P_y-p_4, a contradiction. This proves (1).

Let $\mathcal{Q}$ be the set of components of the subgraph induced on $V(Q_1) \cup \cdots \cup V(Q_4)$. From (1), it follows that $(x, y, p_1, p_2, m, \mathcal{Q})$ satisfies the theorem. This proves Lemma 7.2. $\square$

If we know a C-contrivance $(x, y, p_1, p_2, m, \mathcal{Q})$ for some lightest long hole C (but we do not know C), it is possible to construct a set X of vertices that contains all C-major vertices and does not intersect C. To do so, we first need to reconstruct the path P (in the notation above). If we could do that, then since every C-major vertex has a neighbour in one of P^*, $\mathcal{Q}^*$, and no vertex in $V(C) \setminus (V(P) \cup V(\mathcal{Q}))$ has such a neighbour, we would have the desired set X. So, how to reconstruct P? As for long near-prisms, it is easier if C is the lightest long even hole, rather than just the shortest, and then we would like to use Theorem 6.5 as the analogue of Lemma 3.3. There is a slight problem that did not arise for long near-prisms: the path P we are trying to reconstruct might have length more than $|E(C)|/2$ or close to that, and then we cannot use Theorem 6.5 directly. But if we know a midpoint m of P, then m divides P into two subpaths that are short enough to be reconstructed via Theorem 6.5. For that reason we put the extra vertex m in the definition of a C-contrivance. We can now prove the main result of this section, an analogue of Lemmas 5.2, 5.3 and 5.4.

Theorem 7.3. *There is an algorithm with the following specifications:*

Input: *A candidate G, and a linear ordering of $E(G)$.*

Output: *A list of $\mathcal{O}(|G|^{4\ell-1})$ sets with the following property: for every lightest long even hole C there is some X in the list such that X contains all C-major vertices and $X \cap V(C) = \emptyset$.*

Running time: $\mathcal{O}(|G|^{4\ell+2})$

Proof. First we output the set of all neighbours of y different from x, z, for every induced path x-y-z in G.

Now guess three vertices x, y, m of G and a set $\mathcal{Q}$ of induced paths of G, pairwise anticomplete, with cost at most $4\ell - 4$; and guess $p_1, p_2 \in \mathcal{Q}^*$. If one of x, y belongs to $V(\mathcal{Q})$ or has no neighbour in $\mathcal{Q}^*$, go on to the next guess.

Let Z_1 be the set of vertices in $V(G) \setminus V(\mathcal{Q})$ with a neighbour in $\mathcal{Q}^*$. Let Z_2 be the set of all vertices in $V(G) \setminus (V(\mathcal{Q}) \cup \{m\})$ with a neighbour in $\{x, y\}$. Let $G' = G \setminus (Z_1 \cup Z_2)$, and let R, S be the lightest p_1m-path and p_2m-path in G' respectively. (If these do not exist, or if $R \cup S$ is not an induced path, go on to the next guess.) Let Z_3 be the set of vertices in $V(G) \setminus (V(\mathcal{Q} \cup R \cup S))$ with neighbours in $V(R \cup S)$. Output Z_3, and go on to the next guess. That completes the description of the algorithm.

There are $\mathcal{O}(|G|^{4\ell-1})$ guesses of $(x, y, p_1, p_2, m, \mathcal{Q})$ to check (because $p_1, p_2 \in V(\mathcal{Q})$), and so the output list has size $\mathcal{O}(|G|^{4\ell-1})$. For each guess, we compute Z_1, Z_2, Z_3 in time $\mathcal{O}(|G|^3)$. Hence the total running time is $\mathcal{O}(|G|^{4\ell+2})$.

Now we prove the output is correct. Suppose that C is a lightest long even hole in G. If every C-major vertex is complete to $V(C)$, then the set X satisfies our requirement, where X is the set of all neighbours of y different from x, z, for some three-vertex path x-y-z of C. So we may assume that some C-major vertex is not complete to $V(C)$.

By Lemma 7.2, G contains a C-contrivance $(x, y, p_1, p_2, m, \mathcal{Q})$ with cost at most $4\ell - 4$. We will show that when we guess this C-contrivance, we output the set X that we need. Let P denote the (C, x)-gap with ends p_1, p_2 and with midpoint m. It remains to show that $Z_3 \cap V(C) = \emptyset$, and every C-major vertex belongs to Z_3.

The path $C \setminus P^*$ contains all neighbours of x in $V(C)$, and so by Lemma 6.1, $C \setminus P^*$ has length at least four. Hence $|E(P)| \leq |E(C)| - 4$, and so the paths p_1-P-m and m-P-p_2 both have length at most $\lceil |E(P)|/2 \rceil \leq |E(C)|/2 - 2$. Moreover, p_1-P-m is a path of G', and so the algorithm will compute the lightest p_1m-path R in G', since such a path exists. So p_1-P-m is not lighter than R. But R contains no C-major vertices of G, and by Theorem 6.5, the path p_1-P-m is the lightest p_1m-path of G that contains no C-major vertices, so R is not lighter than p_1-P-m. Consequently R equals the path p_1-P-m. Similarly S is the path m-P-p_2, and so $R \cup S = P$. By definition, the ends of P are in $\mathcal{Q}^*$. Consequently $Z_3 \cap V(C) = \emptyset$. Since every C-major vertex has a neighbour in $V(P)$ it follows that Z_3 contains every C-major vertex. This proves correctness, and so proves Theorem 7.3.

□

8 The main algorithm

Now we prove our main result Theorem 1.1, which we restate:

Theorem 8.1. *For each even integer $\ell \geq 4$ there is an algorithm with the following specifications:*

Input: *A graph G.*

Output: *Decides whether G has an even hole of length at least ℓ.*

Running time: $\mathcal{O}(|G|^{9\ell+1})$.

Proof. The algorithm is as follows. At each step, if we find that G contains a long even hole, we output that fact and stop, so in steps 1,2,3,4,5,7 we can assume the algorithm called at that step outputs the negative answer. Fix a linear ordering of $E(G)$.

Step 1: Apply the algorithm of Theorem 2.1 to test whether G contains a long even hole of length at most 2ℓ in time $\mathcal{O}(|G|^{2\ell})$.

Step 2: Apply the algorithm of Theorem 2.2 to test whether G contains a long jewel of order at most $\ell+1$ in time $\mathcal{O}(|G|^{2\ell+1})$.

Step 3: Apply the algorithm of Theorem 2.4 to test whether G contains a long theta in time $\mathcal{O}(|G|^{2\ell-1})$.

Step 4: Apply the algorithm of Theorem 2.5 to test whether G contains a long ban-the-bomb, in time $\mathcal{O}(|G|^{2\ell+1})$. (If we have not yet found a long even hole, then G is a prospect.)

Step 5: Apply the algorithm of Theorem 3.1 to test whether G contains a long near-prism, in time $\mathcal{O}(|G|^{9\ell+1})$. (If we have still not found a long even hole, then G is a candidate.)

Step 6: Apply the algorithm of Theorem 7.3 to obtain a list $\mathcal{L}$ of subsets of $V(G)$ of length $\mathcal{O}(|G|^{4\ell-1})$ in time $\mathcal{O}(|G|^{4\ell+2})$, with the property that for every lightest long even hole C of G there exists $X \in \mathcal{L}$ with $X \cap V(C) = \emptyset$ that contains all C-major vertices.

Step 7: For every $X \in \mathcal{L}$, apply the algorithm of Theorem 6.6 to $G \setminus X$, to decide that either $G \setminus X$ has a long even hole, or $G \setminus X$ has no clean lightest long even hole, in time $\mathcal{O}(|G|^4)$ for each X, and so in time $\mathcal{O}(|G|^{4\ell+3})$ altogether.

Step 8: Output that G has no long even hole.

For correctness, certainly if the algorithm returns that G has a long even hole then that is true. For the converse, suppose that G has a long even hole, and hence a lightest long even hole C say. Steps 1-5 will either output that there is a long even hole or decide that G is a candidate, and we may assume the latter. Hence, with $\mathcal{L}$ is computed in step 6, there exists $X \in \mathcal{L}$ disjoint from $V(C)$ and containing all C-major vertices. Then in step 7, since C is a clean lightest long even hole of $G \setminus X$, the algorithm of Theorem 6.6 cannot report that $G \setminus X$ has no clean lightest long even hole, and so it will report that $G \setminus X$ has a long even hole, and we return this fact correctly.

For the running time, testing whether G is a candidate (steps 1-5) takes time $\mathcal{O}(|G|^{9\ell+1})$, and determining whether the candidate G contains a long even hole (steps 6-8) takes time $\mathcal{O}(|G|^{4\ell+3})$. Hence, the total running time is $\mathcal{O}(|G|^{9\ell+1})$. This proves Theorem 8.1. $\square$

References

[1] Hsien-Chih Chang and Hsueh-I Lu, "A faster algorithm to recognize even-hole-free graphs", *J. Combinatorial Theory, Ser. B* **113** (2015), 141–161.

[2] Hou-Teng Cheong and Hsueh-I Lu, "Finding a shortest even hole in polynomial time", `arXiv:2008.06740`.

[3] Maria Chudnovsky, Ken-ichi Kawarabayashi and Paul Seymour, “Detecting even holes”, *J. Graph Theory* **48** (2005), 85–111.

[4] Maria Chudnovsky, Alex Scott and Paul Seymour, “Detecting a long odd hole”, *Combinatorica* **41** (2021), 1–30.

[5] Maria Chudnovsky and Paul Seymour, “The three-in-a-tree problem”, *Combinatorica* **30** (2010), 387–417.

[6] Maria Chudnovsky and Rohan Kapadia, “Detecting a theta or a prism”, *SIAM J. Discrete Mathematics* **22** (2008), 1164–1186.

[7] Maria Chudnovsky, Gérard Cornuéjols, Xinming Liu, Paul Seymour and Kristina Vušković, “Recognizing Berge graphs”, *Combinatorica*, **25** (2005), 143-186.

[8] Maria Chudnovsky, Alex Scott, Paul Seymour and Sophie Spirkl, “Detecting an odd hole”, *J. Assoc. Comput. Mach.*, **67** (2020), 1–12, `arXiv:1903.00208`.

[9] Maria Chudnovsky, Alex Scott and Paul Seymour, “Finding a shortest odd hole”, submitted for publication, `arXiv:2004.11874`.

[10] Maria Chudnovsky, Alex Scott, Paul Seymour and Sophie Spirkl, “Proof of the Kalai-Meshulam conjecture”, *Israel J. Math.*, to appear, `arXiv:1810.00065`.

[11] Michele Conforti, Gérard Cornuéjols, Ajai Kapoor, and Kristina Vušković, “Even-hole-free graphs part I: Decomposition theorem”, *J. Graph Theory* **39** (2002), 6–49.

[12] Michele Conforti, Gérard Cornuéjols, Ajai. Kapoor, and Kristina Vušković, “Even-hole-free graphs part II: Recognition algorithm”, *J. Graph Theory* **40** (2002), 238–266.

[13] Linda Cook, *On recognition algorithms and structure of graphs with restricted induced cycles*, PhD thesis, Princeton University, May 2021.

[14] Frédéric Maffray and Nicolas Trotignon, “Algorithms for perfectly contractile graphs”, *SIAM J. Discrete Mathematics* **19** (2005), 553–574.

[15] Kai-Yuan Lai, Hsueh-I Lu and Mikkel Thorup, “Three-in-a-tree in near linear time”, *Proc. 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020*, 1279–1292.

[16] Sepehr Hajebi, private communication, 2019.