A Faster Algorithm to Recognize Even-Hole-Free Graphs*

Hsien-Chih Chang† Hsueh-I Lu‡

August 9, 2013

Abstract

We study the problem of determining whether an n-node graph G has an *even hole*, i.e., an induced simple cycle consisting of an even number of nodes. Conforti, Cornuéjols, Kapoor, and Vušković gave the first polynomial-time algorithm for the problem, which runs in $O(n^{40})$ time. Later, Chudnovsky, Kawarabayashi, and Seymour reduced the running time to $O(n^{31})$. The best previously known algorithm for the problem, due to da Silva and Vušković, runs in $O(n^{19})$ time. In this paper, we solve the problem in $O(n^{11})$ time. Moreover, if G has even holes, our algorithm also outputs an even hole of G in $O(n^{11})$ time.

1 Introduction

A *hole* is an induced simple cycle consisting of at least four nodes. A hole is *even* (respectively, *odd*) if it consists of an even (respectively, odd) number of nodes. See Figure 1 for an illustration. Even-hole-free graphs have been extensively studied in the literature (see, e.g., [1, 12–14, 19, 20, 29, 37]). See Vušković [41] for a recent survey. This paper studies the problem of determining whether a graph has even holes. Let n (respectively, m) be the number of nodes (respectively, edges) of the input graph. Conforti, Cornuéjols, Kapoor, and Vušković [11, 15] gave the first polynomial-time algorithm for the problem, which runs in $O(n^{40})$ time [6]. Later, Chudnovsky, Kawarabayashi, and Seymour [6] reduced the running time to $O(n^{31})$. Chudnovsky et al. [6] also observed that the running time can be further reduced to $O(n^{15})$ as long as prisms can be detected efficiently, but Maffray and Trotignon [30] showed that detecting prisms is NP-hard. The best previously known algorithm for the problem, due to da Silva and Vušković [20], runs in $O(n^{19})$ time. We solve the problem in $O(n^{11})$ time, as stated in the following theorem.

*The current version slightly improves upon the preliminary version [3] appeared in SODA 2012: (a) The time complexity for recognizing even-hole-free n-node m-edge graphs G is reduced from $O(m^2n^7)$ to $O(m^3n^5)$, which is an improvement if $m = o(n^2)$; and (b) if G has even holes, the current version shows how to output an even hole of G also in $O(m^3n^5)$ time.

†Department of Computer Science, University of Illinois at Urbana-Champaign, USA. This research was performed while this author was affiliated with Department of Computer Science and Information Engineering, National Taiwan University. Email: hchang17@illinois.edu.

‡Corresponding author. Department of Computer Science and Information Engineering, National Taiwan University. This author also holds joint appointments in the Graduate Institute of Networking and Multimedia and the Graduate Institute of Biomedical Electronics and Bioinformatics, National Taiwan University. Address: 1 Roosevelt Road, Section 4, Taipei 106, Taiwan, ROC. Research of this author is supported in part by NSC grant 101–2221–E–002–062–MY3. Email: hil@csie.ntu.edu.tw. Web: www.csie.ntu.edu.tw/~hil.

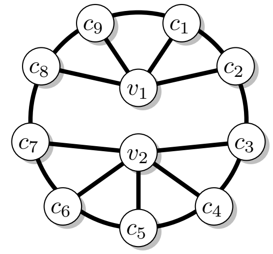


Figure 1: $C = v_1c_2c_3v_2c_7c_8v_1$ is a clean even hole of the 11-node graph G, since $M_G(C) = N_G^{2,2}(C) = \varnothing$. $C' = c_1c_2\cdots c_9c_1$ is an odd hole with $M_G(C') = \{v_2\}$.

Theorem 1.1. *It takes $O(m^3n^5)$ time to determine whether an n-node m-edge connected graph has even holes.*

Technical overview Throughout the paper, a *k-hole* (respectively, *k-cycle* and *k-path*) is a k-node hole (respectively, cycle and path). Our recognition algorithm for even-hole-free graphs consists of two phases. The first phase (see Lemma 2.3) either (1) ensures that the input graph G has even holes via the existence of a "beetle" (see §2 and Figure 2(a)) or a 4-hole in G or (2) produces a set $\mathbb{T}$ of "trackers" $(H, u_1u_2u_3)$, where H is a beetle-free and 4-hole-free induced subgraph of G that contains path $u_1u_2u_3$. $\mathbb{T}$ has the following even-hole-preserving property (see Property 1): If G has even holes, then $\mathbb{T}$ has a "lucky" tracker $(H, u_1u_2u_3)$ in that H has a shortest even hole C of G such that (a) C contains path $u_1u_2u_3$ and (b) the neighborhood of C in H is "super clean" (i.e., $M_H(C) = N_H^{2,2}(C) = N_H^{1,2}(C) = N_H^4(C) = \varnothing$ using notation to be defined in §2). The second phase applies an algorithm (see Lemma 2.4) on each tracker $(H, u_1u_2u_3) \in \mathbb{T}$ to either ensure that H has even holes or ensure that $(H, u_1u_2u_3)$ is not lucky. If all trackers in $\mathbb{T}$ are not lucky, the even-hole-preserving property of $\mathbb{T}$ implies that G is even-hole-free. Otherwise, G has an induced subgraph H containing an even hole, implying that G has even holes.

The recognition algorithm for beetle-free graphs (see the proof of Lemma 2.3) in the first phase is based on Chudnovsky and Seymour's three-in-a-tree algorithm [9] (see Lemma 3.1). If G has beetles or 4-holes, G has even holes. Otherwise, if G has even holes, the neighborhood of each shortest even hole C of G is "clean" (i.e., $N_G^{1,2}(C) = N_G^4(C) = \varnothing$, see the proof of Lemma 2.2). To further ensure that the neighborhood of C is super clean, we generate a set $\mathbb{S}$ of "super cleaners" $(S, u_1u_2u_3)$, where S is a node subset of G and $u_1u_2u_3$ is a path of G, such that at least one super cleaner $(S, u_1u_2u_3) \in \mathbb{S}$ satisfies $u_1u_2u_3 \subseteq C \subseteq H = G \setminus S$ and $M_H(C) = N_H^{2,2}(C) = \varnothing$ for some shortest even hole C of G (see the proof of Lemma 2.3). The set $\mathbb{T}$ consisting of the trackers $(G \setminus S, u_1u_2u_3)$ with $(S, u_1u_2u_2) \in \mathbb{S}$ has the required even-hole-preserving property.

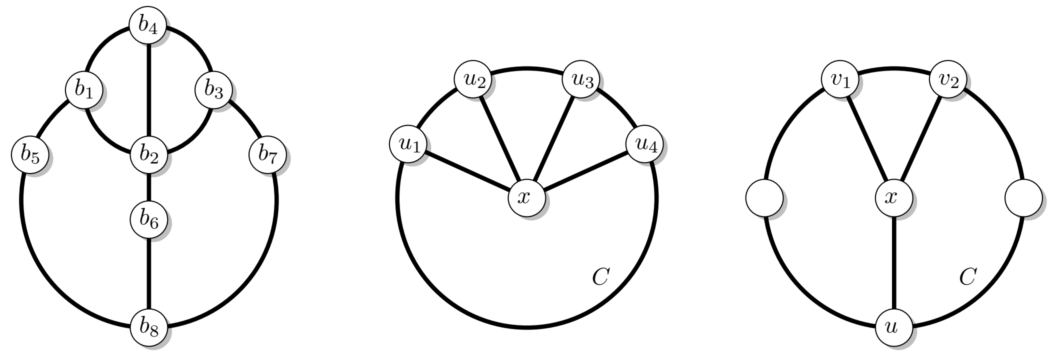
The algorithm applied on each tracker $T = (H, u_1u_2u_3) \in \mathbb{T}$ in the second phase is a decomposition algorithm (see, e.g., the categorization of Vušković [41, §4]) based upon an observation of da Silva and Vušković [20] (see Lemma 4.9) that if a connected graph H is even-hole-free, star-cutset-free, and non-path 2-join-free, then H is an extended clique tree. Since even holes can be efficiently detected in an extended clique tree (see Lemma 4.6, which is a slightly faster implementation of the algorithm of da Silva and Vušković [20]), our algorithm performs two stages of even-hole-preserving decompositions on H, first via star-cutsets and then via non-path 2-joins, until each of the resulting graph either is an extended clique tree or has $O(1)$ nodes. If all of the resulting graphs are even-hole-free, T is not lucky; otherwise, H has even holes. An immediate challenge for the first stage of decompositions is that there are no known polynomial-time detection algorithms for star-cutsets. Fortunately, as noted by Chvátal [10] (see Lemma 4.3), if H is dominated-node-free, a star-cutset of H has to be a full star-cutset of H, which can be efficiently detected. Thus, at the beginning of each decomposition in the first stage, we preprocess $(H, u_1u_2u_3)$ by deleting all dominated nodes of H and carefully updating nodes u_1, u_2, and u_3 such that the luckiness of $(H, u_1u_2u_3)$ is preserved (see Lemma 4.4). The correctness of this preprocessing step relies on the fact that H is beetle-free and the requirement for $(H, u_1u_2u_3)$ to be lucky that the neighborhood of some shortest even hole C in H with $u_1u_2u_3 \subseteq C$ is super clean. Path $u_1u_2u_3$ is crucial in the stage of decompositions via star-cutsets for the dominated-node-free graph H. Specifically, if S is a star-cutset of H, by merely examining the neighborhood of path $u_1u_2u_3$ in H, we can efficiently identify a connected component B of $H \setminus S$ such that $(H[S \cup B], u_1u_2u_3)$ preserves the luckiness of $(H, u_1u_2u_3)$ (see Step 3 in the proof of Lemma 4.1). We then let $H = H[C \cup B]$ and repeat the above procedure for $O(n)$ iterations until H is star-cutset-free. The second stage, i.e., decompositions via non-path 2-joins for star-cutset-free graphs, is based upon the detection algorithm for non-path 2-joins of Charbit et al. [4] (see Lemma 4.8). This stage decomposes an m-edge star-cutset-free graph into a set of $O(m)$ smaller graphs, each of which either consists of $O(1)$ nodes or is an extended clique tree (see the proof of Lemma 4.2).

Related work Even-hole-free planar graphs [33] can be recognized in $O(n^3)$ time. It is NP-complete to determine whether a graph has an even (respectively, odd) hole that contains a given node [2]. The strong perfect graph theorem of Chudnovsky, Robertson, Seymour, and Thomas [7] states that a graph G is perfect if and only if both G and the complement of G are odd-hole-free. Although perfect graphs can be recognized in $O(n^9)$ time [5], the tractability of recognizing odd-hole-free graphs remains open (see, e.g., [25]). Polynomial-time algorithms for detecting odd holes are known for planar graphs [24], claw-free graphs [28, 36], and graphs with bounded clique numbers [16]. Graphs without holes (i.e., chordal graphs) can be recognized in $O(m+n)$ time [34, 35, 38]. Graphs without holes consisting of five or more nodes (i.e., weakly chordal graphs) can be recognized in $O(m^2+n)$ time [31, 32]. It takes $O(n^2)$ time to detect a hole that contains any $o((\log n/\log\log n)^{2/3})$ given nodes in an $O(1)$-genus graph [26, 27]. See [8, 17, 21, 42] for more results on odd-hole-free graphs.

Road map The rest of the paper is organized as follows. Section 2 gives the preliminaries and proves Theorem 1.1 by Lemmas 2.3 and 2.4. Section 3 proves Lemma 2.3. Section 4 proves Lemma 2.4. Section 5 concludes the paper by explaining how to augment Theorem 1.1 into an $O(m^3n^5)$-time algorithm that outputs an even hole of an n-node m-edge graph with even holes.

2 Preliminaries and the topmost structure of our proof

Unless clearly specified otherwise, all graphs throughout the paper are undirected and simple. Let $|S|$ be the cardinality of set S. Let G be a graph. Let $V(G)$ consist of the nodes in G. For any subgraph H of G, let $G[H]$ denote the subgraph of G induced by $V(H)$. Subgraphs H and H' of graph G are *adjacent* in G if some node of H and some node of H' are adjacent in G. For any subset U of $V(G)$, let $G \setminus U = G[V(G) \setminus U]$. For any subgraph H of G, let $N_G(H)$ consist of the nodes of $V(G) \setminus V(H)$ that are adjacent to H in G and let $N_G[H] = N_G(H) \cup V(H)$.



(a) (b) (c)

Figure 2: (a) A beetle B, where $B[\{b_1, b_2, b_3, b_4\}]$ is a diamond. (b) If $x \in N_G^4(C)$, then $G[C \cup \{x\}]$ is a beetle B, where $B[\{u_1, u_2, u_3, x\}]$ is a diamond. (c) A node $x \in N_G^{1,2}(C)$.

Let C be a hole of G. A node $x \in V(G) \setminus V(C)$ is a *major node* [6] of C in G if at least three distinct nodes of $N_C(x)$ are pairwise non-adjacent in G. Let $M_G(C)$ consist of the major nodes of C in G. For instance, in Figure 1, $M_G(C) = \varnothing$ and $M_G(C') = \{v_2\}$.

Lemma 2.1 (Chudnovsky et al. [6, Lemma 2.2]). *If C is a shortest even hole of graph G and $x \in M_G(C)$, $|N_C(x)|$ is even.*

If $x \in N_G(C) \setminus M_G(C)$, $1 \leq |N_C(x)| \leq 4$ and $C[N_C(x)]$ has at most two connected components. Moreover, if $C[N_C(x)]$ is not connected, each connected component of $C[N_C(x)]$ has at most two nodes. Let $N_G^i(C)$ with $1 \leq i \leq 4$ consist of the nodes $x \in N_G(C) \setminus M_G(C)$ such that $|N_C(x)| = i$ and $C[N_C(x)]$ is connected. Let $N_G^{i,j}(C)$ with $1 \leq i \leq j \leq 2$ consist of the nodes $x \in N_G(C) \setminus M_G(C)$ such that $C[N_C(x)]$ is not connected and the two connected components of $C[N_C(x)]$ has i and j nodes, respectively. We have

$$N_G(C) = N_G^1(C) \cup N_G^2(C) \cup N_G^3(C) \cup N_G^4(C) \cup N_G^{1,1}(C) \cup N_G^{1,2}(C) \cup N_G^{2,2}(C) \cup M_G(C). \quad (1)$$

We say that C is a *clean* hole of G if $M_G(C) = N_G^{2,2}(C) = \varnothing$. For any 3-path $u_1u_2u_3$ of G, a clean hole C of G that contains path $u_1u_2u_3$ is a $u_1u_2u_3$*-hole* of G if C is a shortest even hole of G. For instance, if G is as shown in Figure 1, $C = v_1c_2c_3v_2c_7c_8v_1$ is a $v_1c_2c_3$-hole of G. If H is an induced subgraph of G and $u_1u_2u_3$ is a 3-path of H, we call $(H, u_1u_2u_3)$ a *tracker* of G. A tracker $(H, u_1u_2u_3)$ of G is *lucky* if H contains a $u_1u_2u_3$-hole of G. If G has lucky trackers, G has even holes. The following even-hole-preserving property of $\mathbb{T}$ reduces the problem of determining whether G is even-hole-free to the problem of determining whether all trackers in $\mathbb{T}$ are not lucky.

Property 1. *If G has even holes, $\mathbb{T}$ contains a lucky tracker of G.*

An induced subgraph of G is a *beetle* of G if it consists of (1) a 4-cycle $b_1b_2b_3b_4b_1$ with exactly one chord b_2b_4 (i.e., a *diamond* [15, 29] of G) and (2) a tree I of $G \setminus \{b_4\}$ having exactly three leaves b_1, b_2, and b_3 with the property that $I \setminus \{b_1, b_2, b_3\}$ is an induced tree of G not adjacent to b_4. See Figure 2(a) for an illustration. At least one of the three cycles in $B \setminus \{b_2\}$, $B \setminus \{b_1, b_4\}$, and $B \setminus \{b_3, b_4\}$ is an even hole of G. Nodes b_5, b_6, b_7, and b_8 need not be distinct. For instance, as illustrated by Figure 2(b), if C is a hole of G and x is a node of $N_G^4(C)$, then $G[C \cup \{x\}]$ is a beetle of G.

Lemma 2.2. *If G is a beetle-free graph, $N_G(C) \subseteq N_G^{1,1}(C) \cup N_G^1(C) \cup N_G^2(C) \cup N_G^3(C)$ holds for any clean shortest even hole C of G.*

Proof. By $M_G(C) = N_G^{2,2}(C) = \varnothing$ and Equation (1), it suffices to show $N_G^{1,2}(C) = N_G^4(C) = \varnothing$. If $x \in N_G^4(C)$ as illustrated by Figure 2(b), then $G[C \cup \{x\}]$ is a beetle of G, a contradiction. If $x \in N_G^{1,2}(C)$, then let u, v_1, and v_2 be the nodes of $N_C(x)$ such that v_1 and v_2 are adjacent in C, as illustrated by Figure 2(c). Let P_1 be the path of $C \setminus \{v_2\}$ between u and v_1. Let P_2 be the path of $C \setminus \{v_1\}$ between u and v_2. Either $G[\{x\} \cup P_1]$ or $G[\{x\} \cup P_2]$ is an even hole of G shorter than C, a contradiction. The lemma is proved. □

2.1 Proving Theorem 1.1

Lemma 2.3. *It takes $O(m^3n^5)$ time to complete either one of the following tasks for any n-node m-edge graph G. Task 1: Ensuring that G has even holes. Task 2: (a) Ensuring that G has no beetles and (b) obtaining a set $\mathbb{T}$ of $O(m^2n)$ trackers of G that satisfies Property 1.*

Lemma 2.4. *Given a tracker $T = (H, u_1u_2u_3)$ of an n-node beetle-free graph G, it takes $O(mn^4)$ time to either ensure that H has even holes or ensure that T is not lucky.*

Proof of Theorem 1.1. We apply Lemma 2.3 on G in $O(m^3n^5)$ time. If Task 1 is completed, then the theorem is proved. If Task 2 is completed, then G has no beetles and we have a set $\mathbb{T}$ of $O(m^2n)$ trackers of G that satisfies Property 1. By Property 1 of $\mathbb{T}$ and Lemma 2.4, one can determine whether G has even holes in time $|\mathbb{T}| \cdot O(mn^4) = O(m^3n^5)$. The theorem is proved. □

3 Proving Lemma 2.3

A *clique* of G is a complete subgraph of G. A clique of G is *maximal* if it is not contained by other cliques of G. We need the following four lemmas to prove Lemma 2.3.

Lemma 3.1 (Chudnovsky and Seymour [9])**.** *It takes $O(n^4)$ time to determine whether an n-node graph has an induced tree that contains three given nodes.*

Lemma 3.2 (da Silva and Vušković [19, Section 2] and Farber [22, Proposition 2])**.** *If G is an n-node m-edge 4-hole-free graph, it takes $O(mn^2)$ time to either ensure that G has even holes or obtain all $O(n^2)$ maximal cliques of G.*

Lemma 3.3 (Chudnovsky, Kawarabayashi, and Seymour [6, Lemma 4.2])**.** *Any shortest even hole C of a 4-hole-free graph G contains an edge v_1v_2 with $N_G^{2,2}(C) \subseteq N_G(v_1) \cap N_G(v_2)$.*

Lemma 3.4. *For any shortest even hole C of a 4-hole-free graph G, if $G[M_G(C)]$ is not a clique of G, there is a node u of C with $M_G(C) \subseteq N_G(u)$.*

Before proving Lemma 3.4, we first prove Lemma 2.3 using Lemmas 3.1, 3.2, 3.3, and 3.4.

Proof of Lemma 2.3. We claim that G has beetles if and only if at least one of the $O(m^3n)$ choices of node b_4 and three distinct edges b_1b_5, b_2b_6, and b_3b_7 of G satisfies the following conditions.

- $G[\{b_1, b_2, b_3, b_4\}]$ is the 4-cycle $b_1b_2b_3b_4b_1$ with exactly one chord b_2b_4.
- The edges between $\{b_1, b_2, b_3\}$ and $\{b_5, b_6, b_7\}$ are exactly b_1b_5, b_2b_6, and b_3b_7.

- $\{b_5, b_6, b_7\} \cap \{b_1, b_2, b_3, b_4\} = \varnothing$, but nodes b_5, b_6, and b_7 need not be distinct.
- An induced tree I' of $G \setminus ((N_G[b_1] \cup \cdots \cup N_G[b_4]) \setminus \{b_5, b_6, b_7\})$ contains $\{b_5, b_6, b_7\}$.

The claim can be verified by seeing that if I'' is the minimal subtree of I' that contains $\{b_5, b_6, b_7\}$, then $I = I'' \cup \{b_1b_5, b_2b_6, b_3b_7\}$ is a tree of $G \setminus \{b_4\}$ with leaf set $\{b_1, b_2, b_3\}$ having the property that $I \setminus \{b_1, b_2, b_3\}$ is an induced tree of G not adjacent to b_4. By the claim and Lemma 3.1, it takes $O(m^3n^5)$ time to determine whether G has beetles. It takes $O(n^4)$ time to determine whether G has 4-holes. If G has 4-holes or beetles, G has even holes. The lemma is proved by completing Task 1 in $O(m^3n^5)$ time. The rest of the proof assumes that G is 4-hole-free and beetle-free.

By Lemma 3.2, it takes $O(mn^2)$ time to either ensure that G has even holes or obtain the $O(n^2)$ maximal cliques of G. If G has even holes, the lemma is proved by completing Task 1 in $O(mn^2)$ time. Otherwise, let $\mathbb{T}$ consist of the trackers of G that are in the form of $(G \setminus S_1, u_1u_2u_3)$ or $(G \setminus S_2, u_1u_2u_3)$ with

$$\begin{array}{lclcl} S_1 & = & S_1(u_1, u_2, u_3, v_1, v_2) & = & (N_G(v_1) \cap N_G(v_2)) \cup (N_G(u_2) \setminus \{u_1, u_3\}); \\ S_2 & = & S_2(u_1, u_2, K) & = & (N_G(u_1) \cap N_G(u_2)) \cup V(K), \end{array}$$

where u_1u_2 and v_1v_2 are edges of G and K is a maximal clique of G. We have $|\mathbb{T}| = O(m^2n) + O(mn^2) = O(m^2n)$. Since all $O(n^2)$ maximal cliques of G are available, $\mathbb{T}$ can be computed in time $O(m^2n) \cdot O(n+m) = O(m^3n)$ time. To ensure the completion of Task 2, it remains to show that $\mathbb{T}$ satisfies Property 1. Suppose that G has even holes. Let C be an arbitrary shortest even hole of G. The following case analysis shows that $\mathbb{T}$ contains lucky trackers of G.

Case 1: $M_G(C) \subseteq N_G(u_2)$ holds for a node u_2 of C. Let u_1 and u_3 be the neighbors of u_2 in C. By $M_G(C) \subseteq N_G(u_2) \setminus \{u_1, u_3\}$ and Lemma 3.3, there is an edge v_1v_2 of C with $M_G(C) \cup N_G^{2,2}(C) \subseteq S_1$. By the choices of u_1 and u_3, we have $(N_C(u_2) \setminus \{u_1, u_3\}) \cap C = \varnothing$. Since v_1v_2 is an edge of hole C, we have $N_G(v_1) \cap N_G(v_2) \cap C = \varnothing$. Thus, $S_1 \cap C = \varnothing$, implying that C is a clean hole of $G \setminus S_1$ containing path $u_1u_2u_3$. Since C is a shortest even hole of G, C is also a shortest even hole of $G \setminus S_1$. Therefore, C is a $u_1u_2u_3$-hole of $G \setminus S_1$.

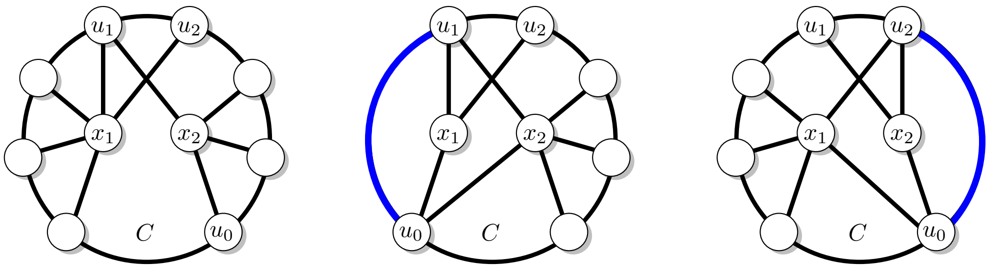
Case 2: $M_G(C) \not\subseteq N_G(u)$ holds for all nodes u of C. By Lemma 3.4, $G[M_G(C)]$ is a clique of G. Let K be a maximal clique of G with $M_G(C) \subseteq V(K)$. Combining with Lemma 3.3, there is an edge u_1u_2 of C with $M_G(C) \cup N_G^{2,2}(C) \subseteq S_2$. We have $V(K) \cap C = \varnothing$ or else $M_G(C) \cap C = \varnothing$ implies $M_G(C) \subseteq V(K) \setminus \{u\} \subseteq N_G(u)$ for any node $u \in V(K) \cap C$, a contradiction. Since u_1u_2 is an edge of C, we have $N_G(u_1) \cap N_G(u_2) \cap C = \varnothing$. Thus, $S_2 \cap C = \varnothing$, implying that C is a clean hole of $G \setminus S_2$ containing path $u_1u_2u_3$, where u_3 is the neighbor of u_2 in C other than u_1. Since C is a shortest even hole of G, C is also a shortest even hole of $G \setminus S_2$. Therefore, C is a $u_1u_2u_3$-hole of $G \setminus S_2$. □

The rest of the section proves Lemma 3.4. An edge u_1u_2 of hole C is a *gate* [6] of C with respect to major nodes x_1 and x_2 of C if the following conditions hold:

Condition G1: There are two edges u_1x_2 and u_2x_1 and at least one of edges u_1x_1 and u_2x_2.

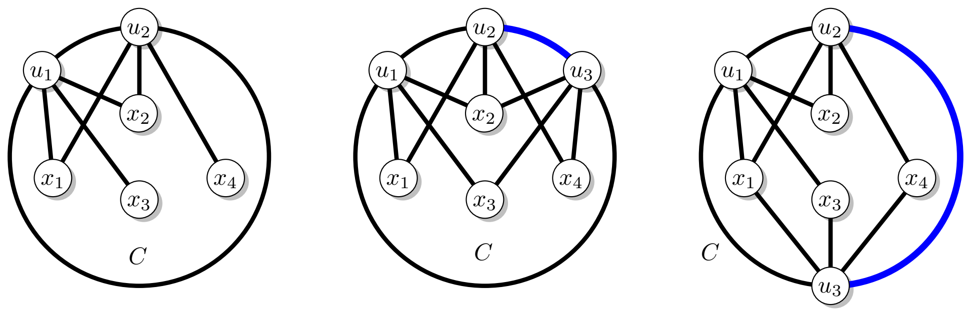
Condition G2: There is a node u_0 of $C \setminus \{u_1, u_2\}$ such that x_1 (respectively, x_2) is not adjacent to $C \setminus V(P_1)$ (respectively, $C \setminus V(P_2)$), where P_1 (respectively, P_2) is the path of C between u_2 (respectively, u_1) and u_0 that contains u_1 (respectively, u_2).

See Figure 3 for an illustration.



(a) (b) (c)

Figure 3: (a) Edge u_1u_2 is a gate of the 8-hole C induced by nodes other than x_1 and x_2, which are the major nodes of C. (b) and (c) Illustrations for the proof of Lemma 3.4.



(a) (b) (c)

Figure 4: Illustrations for the proof of Lemma 3.4.

Lemma 3.5 (Chudnovsky et al. [6, Lemmas 2.3 and 2.4])**.** *The following statements hold for any shortest even hole C of a 4-hole-free graph G.*

1. *If x_1 and x_2 are non-adjacent nodes of $M_G(C)$, there is a gate of C with respect to x_1 and x_2 in G.*
2. *If X is a subset of $M_G(C)$ with $|X| = 3$ such that $G[X]$ has at most one edge, $X \subseteq N_G(u)$ holds for some node u of C.*

Proof of Lemma 3.4. Let x_1 and x_2 be two non-adjacent nodes of $M_G(C)$. Let U consist of the nodes u of C that are adjacent to both of x_1 and x_2. By Lemma 3.5(1), there is a gate u_1u_2 of C with respect to x_1 and x_2. We have $\varnothing \neq U \subseteq \{u_1, u_2, u_0\}$, where u_0 is a node of C ensured by Condition G2. Assume $u_0 \in U$ for contradiction. By Condition G1, u_0 is adjacent u_1 or u_2 in G or else one of $u_1x_1u_0x_2u_1$ and $u_2x_1u_0x_2u_2$ would be a 4-hole of G. If u_0 is adjacent to u_1 as illustrated by Figure 3(b), then Condition G2 implies $N_C(x_1) = \{u_0, u_1, u_2\}$, which contradicts with $x_1 \in M_G(C)$. If u_0 is adjacent to u_2 as illustrated by Figure 3(c), then Condition G2 implies $N_G(x_2) = \{u_0, u_1, u_2\}$, which contradicts with $x_2 \in M_G(C)$. We have $u_0 \notin U$, and thus $U \subseteq \{u_1, u_2\}$. The lemma holds trivially if $|M_G(C)| = 2$. To prove the lemma for $|M_G(C)| \geq 3$, we first show the claim: "*Each node $x \in M_G(C) \setminus \{x_1, x_2\}$ is adjacent to U.*" If one of x_1 and x_2 is not adjacent to x, the claim follows from Lemma 3.5(2). If both of x_1 and x_2 are adjacent to x, each node $u \in U$ is adjacent to x in G or else ux_1xx_2u is a 4-hole, a contradiction. The claim is proved.

By the above claim, the lemma holds if $|M_G(C)| = 3$ or $|U| = 1$. It remains to consider the cases with $|M_G(C)| \geq 4$ and $U = \{u_1, u_2\}$ (thus, there are edges u_1x_1 and u_2x_2) by showing that either u_1

or u_2 is adjacent to each node $x \in M_G(C)$. Assume $x_3 \in M_G(C) \setminus N_G(u_2)$ and $x_4 \in M_G(C) \setminus N_G(u_1)$ for contradiction. By the above claim, G has edges u_1x_3 and u_2x_4. We know $x_3 \notin N_G(x_4)$ or else $u_1u_2x_4x_3u_1$ is a 4-hole. See Figure 4(a). Observe that x_4 cannot be adjacent to both of x_1 and x_2 or else $u_1x_1x_4x_2u_1$ is a 4-hole. *Case 1: x_4 is not adjacent to x_2.* By Lemma 3.5(2), a node u_3 of C is adjacent to all of x_2, x_3, and x_4. Since u_3 is adjacent to both of x_3 and x_4, we have $u_3 \notin \{u_1, u_2\}$. See Figure 4(b). If C has edge u_2u_3, $u_1x_3u_3u_2u_1$ is a 4-hole; otherwise, $u_2x_2u_3x_4u_2$ is a 4-hole, a contradiction. *Case 2: x_4 is not adjacent to x_1.* By Lemma 3.5(2), a node u_3 of C is adjacent to all of x_1, x_3, and x_4. Since u_3 is adjacent to both of x_3 and x_4, we have $u_3 \notin \{u_1, u_2\}$. See Figure 4(c). If C has edge u_2u_3, $u_1x_3u_3u_2u_1$ is a 4-hole; otherwise, $u_2x_1u_3x_4u_2$ is a 4-hole, a contradiction. The lemma is proved. □

4 Proving Lemma 2.4

Subset S of $V(H)$ is a *star-cutset* [10] of graph H if $S \subseteq N_H[s]$ holds for some node s of S and the number of connected components of $H \setminus S$ is more than that of H.

Lemma 4.1. *For any tracker $T = (H, u_1u_2u_3)$ of an n-node m-edge beetle-free connected graph G, it takes $O(mn^3)$ time to complete one of the following three tasks. Task 1: Ensuring that H has even holes. Task 2: Ensuring that T is not lucky. Task 3: Obtaining a star-cutset-free induced subgraph H' of H such that if T is lucky, H' has even holes.*

Lemma 4.2. *It takes $O(mn^4)$ time to determine if an n-node m-edge star-cutset-free graph has even holes.*

Proof of Lemma 2.4. We apply Lemma 4.1 on the input tracker $T = (H, u_1u_2u_3)$ of G in $O(mn^3)$ time. If Task 1 or 2 is completed, the lemma is proved. If Task 3 is completed, since H' is star-cutset-free, Lemma 4.2 implies that it takes $O(mn^4)$ time to determine whether H' has even holes. Since H' is an induced subgraph of H, if H' has even holes, so does H; otherwise, T is not lucky. The lemma is proved. □

Subsection 4.1 proves Lemma 4.1. Subsection 4.2 proves Lemma 4.2.

4.1 Proving Lemma 4.1

A star-cutset S of graph H is *full* if $S = N_H[s]$ holds for some node s of S. No polynomial-time algorithms are known for detecting star-cutsets (see, e.g., [15]), but full star-cutsets in an n-node m-edge graph can be detected in $O(mn)$ time. Node x *dominates* node y in graph H if $x \neq y$ and $N_H[y] \subseteq N_H[x]$. Node y is *dominated* in H if some node of H dominates y in H. We need the following three lemmas to prove Lemma 4.1.

Lemma 4.3 (Chvátal [10, Theorem 1])**.** *A graph without dominated nodes and full star-cutsets has no star-cutsets.*

Lemma 4.4. *If $T = (H, u_1u_2u_3)$ is a tracker of an n-node m-edge beetle-free connected graph G, it takes $O(mn^2)$ time to obtain a tracker $T' = (H', u'_1u'_2u'_3)$ of G, where H' is a dominated-node-free induced subgraph of H, such that if T is lucky, so is T'.*

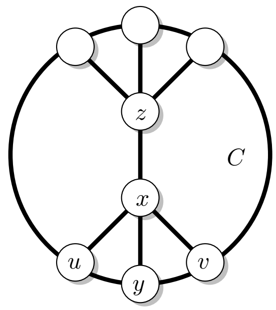


Figure 5: An illustration for the proof of Lemma 4.4.

Proof. We first prove the following claim for any beetle-free graph H: *"If a node x of H dominates a node y of a clean shortest even hole C of H, then $C' = H[C \cup \{x\} \setminus \{y\}]$ is a clean shortest even hole of H."* Let u and v be the neighbors of y on C. Since C is a hole and $y \in C$, we know $x \notin C$, implying $x \in N_H(C)$. Since x dominates y and $|N_C[y]| = 3$, there is a connected component of $C[N_C(x)]$ with at least 3 nodes. By Lemma 2.2, we have $x \in N_H^3(C)$, implying $N_C(x) = \{u, y, v\}$. Thus, C' is a shortest even hole of H. Assume $z \in M_H(C') \cup N_H^{2,2}(C')$ for contradiction. By $y \in N_H^3(C')$, $z \neq y$. By $C \setminus \{y\} = C' \setminus \{x\}$, exactly one of x and y is adjacent to z in H or else $z \in M_H(C) \cup N_H^{2,2}(C)$, contradicting to the fact that C is clean. *Case 1:* $z \in N_H^{2,2}(C')$. If $z \in N_H(y) \setminus N_H(x)$, we have $z \in M_H(C)$, contradicting to the assumption that C is a clean hole of H. If $z \in N_H(x) \setminus N_H(y)$, we have $z \in N_H^{1,2}(C)$, contradicting to Lemma 2.2. *Case 2:* $z \in M_H(C')$. By $|N_{C'}(z)| \geq 3$ and Lemma 2.1, $|N_{C'}(z)| \geq 4$. By $M_H(C) = N_H^{2,2}(C) = \varnothing$ and Lemma 2.2, $|N_C(z)| \leq 3$. By $C \setminus \{x\} = C \setminus \{y\}$, we have $z \in N_H(x) \setminus N_H(y)$, $|N_C(z)| = 3$, and $|N_{C'}(z)| = 4$. By Lemma 2.2, $z \in N_H^3(C)$. See Figure 5 for an illustration. Thus, $C[N_C(z)]$ is a 3-path, implying that $H[C' \cup \{z\}]$ is a beetle B of H in which $B[N_B[z]]$ is a diamond, a contradiction. The claim is proved.

The algorithm first iteratively updates $(H, u_1u_2u_3)$ by the following steps until H has no dominated nodes, and then outputs the resulting $(H, u_1u_2u_3)$ as $(H', u'_1u'_2u'_3)$.

Step 1: Let x and y be two nodes of H such that x dominates y.
Step 2: If there is an $i \in \{1, 2, 3\}$ with $y = u_i$, then let $u_i = x$.
Step 3: Let $H = H \setminus \{y\}$.

It takes $O(mn)$ time to detect nodes x and y such that x dominates y. Each iteration of the loop decreases $|V(H)|$ by one via Step 3. Therefore, the overall running time is $O(mn^2)$. Graph H' is a dominated-node-free induced subgraph of the initial H. It suffices to ensure that if the tracker $T = (H, u_1u_2u_3)$ of G at the beginning of an iteration is lucky, the tracker at the end of the iteration, denoted $T' = (H', u'_1, u'_2, u'_3)$, remains lucky. Let C be a $u_1u_2u_3$-hole of H. If $y \notin C$, C is a $u'_1u'_2u'_3$-hole of $H' = H \setminus \{y\}$. If $y \in C$, the above claim ensures that $C' = H[C \cup \{x\} \setminus \{y\}]$ is a clean shortest even hole of H. Since x dominates y, hole C' contains path $u'_1u'_2u'_3$. Thus, C' is a $u'_1u'_2u'_3$-hole of H'. Either way, $(H', u'_1u'_2u'_3)$ is lucky. The lemma is proved. □

Lemma 4.5. *If $(H, u_1u_2u_3)$ is a lucky tracker of graph G and S is a full star-cutset of H, one of the following two conditions holds.*

Condition B1: For each $u_1u_2u_3$-hole C of H, there exists a connected component B of $H \setminus S$ satisfying $C \subseteq H[B \cup S]$.

Condition B2: There are two non-adjacent nodes s_1 and s_2 of S and two connected components B_1 and B_2 of $H \setminus S$ with $\{s_1, s_2\} \subseteq N_H(B_1)$ and $\{s_1, s_2\} \subseteq N_H(B_2)$.

Proof. Let s be a node of S with $N_H[s] = S$. Let C be a $u_1u_2u_3$-hole of H. Assume that Condition B1 does not hold. There exist two distinct connected components B_1 and B_2 of $H \setminus S$ such that $V(C) \cap V(B_1) \neq \varnothing$ and $V(C) \cap V(B_2) \neq \varnothing$. Thus, $C[S]$ has at least two connected components. Let s_1 and s_2 be two nodes in distinct connected components of $C[S]$. By $\{s_1, s_2\} \subseteq N_H[s]$, we have $s \notin C$ or else s, s_1, and s_2 are in the same connected component of $C[S]$. By Lemma 2.2, we have $s \in N_H^{1,1}(C)$, implying $\{s_1, s_2\} = V(C) \cap S$. It follows that both s_1 and s_2 are adjacent to both B_1 and B_2. Let paths P_1 and P_2 be the two connected components of $C \setminus \{s_1, s_2\}$. B_1 has to contain one of P_1 and P_2 and B_2 has to contain other one of P_1 and P_2. Therefore, Condition B2 holds. The lemma is proved. □

Proof of Lemma 4.1. Let T_0 be the initial given tracker $(H, u_1u_2u_3)$ of G. The algorithm iteratively updates $(H, u_1u_2u_3)$ by the following three steps until Task 1, 2, or 3 is completed.

Step 1: Apply Lemma 4.4 in $O(mn^2)$ time on tracker $T = (H, u_1u_2u_3)$ to obtain a tracker $T' = (H', u'_1u'_2u'_3)$ of G, where H' is a dominated-node-free induced subgraph of H, such that if T is lucky, so is T'. Determine in $O(mn)$ time whether H' has full star-cutsets. If H' has full star-cutsets, then let $(H, u_1u_2u_3) = (H', u'_1u'_2u'_3)$ and proceed to Step 2; Otherwise, complete Task 3 by outputting H'.

Step 2: Let S be a full star-cutset of H. If Condition B2 of Lemma 4.5 holds, then complete Task 1 by outputting that G has even holes. Otherwise, proceed to Step 3.

Step 3: If either one of the following statements hold for $U = \{u_1, u_2, u_3\}$:

- $U \subseteq S$ and a connected component B of $H \setminus S$ is adjacent to both u_1 and u_3;
- $U \nsubseteq S$ and $U \subseteq B \cup S$ holds for a connected component B of $H \setminus S$,

then let $H = H[B \cup S]$ and proceed to the next iteration of the loop. Otherwise, complete Task 2 by outputting that T_0 is not lucky.

Step 1 does not increase $|V(H)|$. If Step 3 updates H, then $|V(H)|$ is decreased by at least one, since $H \setminus S$ has more than one connected component. The algorithm halts in $O(n)$ iterations. Step 1 takes $O(mn^2)$ time. Step 2 takes $O(mn^2)$ time: For any two non-adjacent nodes s_1 and s_2 in S, it takes $O(m)$ time to determine whether s_1 and s_2 have two or more common neighboring connected components of $H \setminus S$. Step 3 takes $O(m)$ time. The overall running time is $O(mn^3)$.

We first show the following claim for each iteration of the algorithm: *"If the $(H, u_1u_2u_3)$ at the beginning of an iteration is a lucky tracker of G, then (1) the intermediate $(H, u_1u_2u_3)$ throughout the iteration remains a lucky tracker of G, and (2) Step 3, if reached, proceeds to the next iteration."* It suffices to consider the situation that Step 3 is reached and focus on the update operation that replaces H with $H[B \cup S]$ via Step 3. By definition of Step 2, Condition B2 does not hold. By Lemma 4.5, Condition B1 holds. That is, there is a connected component B^* of $H \setminus S$ such that $H[B^* \cup S]$ contains some $u_1u_2u_3$-hole C of H. We prove the claim by showing that B^* has to be the connected component B of $H \setminus S$ in Step 3. Since $B^* = B$ holds trivially for the case $\{u_1, u_2, u_3\} \nsubseteq S$, we assume $\{u_1, u_2, u_3\} \subseteq S$. If $s \in C$, there are exactly two nodes of C that are adjacent to s in H;

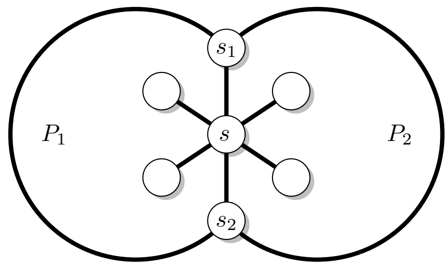


Figure 6: An illustration for the proof of Lemma 4.1.

otherwise, Lemma 2.2 implies that s has at most three neighbors of H in C. Either way, we have $|V(C) \cap S| \leq 3$. Since $u_1u_2u_3$ is a path of even hole C, nodes u_1 and u_3 are not adjacent in H. Since Condition B2 does not hold, at most one connected component of $H \setminus S$ can be adjacent to both u_1 and u_3 in H. By $V(C) \subseteq B^* \cup S$ and $|V(C) \cap S| \leq 3$, we have $(N_C(u_1) \cup N_C(u_3)) \setminus \{u_2\} \subseteq B^*$, implying $B^* = B$. The claim is proved.

For the correctness of the algorithm, we consider the three possible steps via which the algorithm halts. Step 1: Since H' is dominated-node-free and full-star-cutset-free, Lemma 4.3 implies that H' has no star-cutsets. By the above claim, Task 3 is completed. Step 2: Condition B2 holds. Let P_1 be a shortest path between s_1 and s_2 in $H[B_1 \cup \{s_1, s_2\}]$. Let P_2 be a shortest path between s_1 and s_2 in $H[B_2 \cup \{s_1, s_2\}]$. Since s_1 and s_2 are not adjacent, at least one of the three cycles of graph $P_1 \cup P_2 \cup \{ss_1, ss_2\}$ is an even hole of H. Since H is an induced subgraph of G, G has even holes. See Figure 6 for an illustration. Task 1 is completed. Step 3: By the above claim, if T_0 is lucky, Step 3 always proceeds to the next iteration of the loop. Thus, Task 2 is completed. The lemma is proved. □

4.2 Proving Lemma 4.2

4.2.1 Extended clique trees

Graph H is an *extended clique tree* [20] if there is a set S of two or less nodes of H such that each biconnected component of $H \setminus S$ is a clique. da Silva and Vušković [20, §2.3] described an $O(n^5)$-time algorithm to determine whether an n-node extended clique tree contains even holes, which can actually be implemented to run in $O(n^4)$ time.

Lemma 4.6. *It takes $O(n^4)$ time to determine whether an n-node extended clique tree has even holes.*

Proof. Let H_0 be the n-node extended clique tree. Let x and y be two nodes of H_0 such that each biconnected component of $H = H_0 \setminus \{x, y\}$ is a clique. For nodes u and v of H, let $P(u, v)$ be the shortest path of H between u and v and let $p(u, v)$ be the number of edges in $P(u, v)$. We spend $O(n^4)$ time to store the following information in a table M_1 for every two nodes u and v of H: (i) $p(u, v)$ and (ii) whether or not $P(u, v) \setminus \{u, v\}$ is adjacent to x (respectively, y). With M_1, it takes $O(n^2)$ time to determine whether H_0 has an even hole that contains y but not x: $H_0 \setminus \{x\}$ has an even hole if and only if there are two non-adjacent neighbors u and v of y in H such that $p(u, v)$ is even and $P(u, v) \setminus \{u, v\}$ is not adjacent to y. Similarly, with M_1, it takes $O(n^2)$ time to determine whether H_0 has an even hole that contains x but not y.

To determine whether H_0 has an even hole containing both x and y, we store in a table M_2 for every four nodes u_1, v_1, u_2, v_2 whether or not $P(u_1, v_1)$ and $P(u_2, v_2)$ are both disjoint and

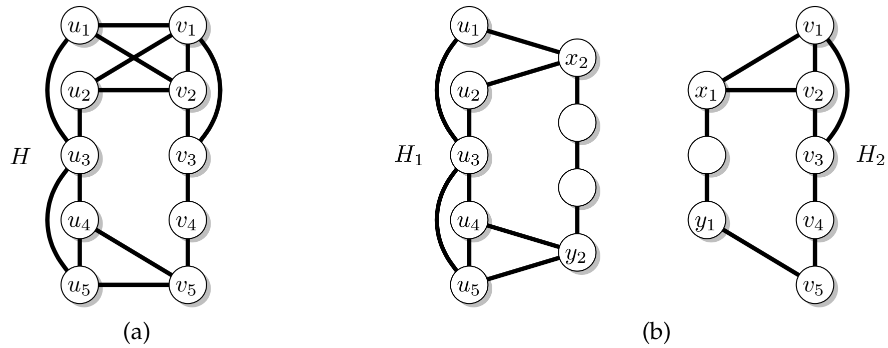


Figure 7: (a) A connected non-path 2-join $V_1|V_2$ of graph H with split (X_1, Y_1, X_2, Y_2), where $X_1 = \{u_1, u_2\}$, $X_2 = \{v_1, v_2\}$, $Y_1 = \{u_4, u_5\}$, $Y_2 = \{v_5\}$, $V_1 = X_1 \cup Y_1 \cup \{u_3\}$, and $V_2 = X_2 \cup Y_2 \cup \{v_3, v_4\}$. (b) The parity-preserving blocks of decomposition H_1 and H_2 of H for the connected 2-join $V_1|V_2$ with respect to the split (X_1, Y_1, X_2, Y_2).

non-adjacent. It takes $O(n^2)$ time to compute the connected components of $H \setminus N_H[P(u_1, v_1)]$. Paths $P(u_1, v_1)$ and $P(u_2, v_2)$ are both disjoint and non-adjacent if and only if u_2 and v_2 are in the same connected component of $H \setminus N_H[P(u_1, v_1)]$. Therefore, M_2 can also be computed in $O(n^4)$ time. With tables M_1 and M_2, it takes $O(n^4)$ time to determine whether H_0 has an even hole containing both x and y: *Case 1: x and y are adjacent in H_0.* H_0 has an even hole containing both x and y if and only if there are nodes u and v such that (1) $H_0[\{u, x, y, v\}]$ is path $uxyv$, (2) $p(u, v)$ is odd, and (3) $P(u, v) \setminus \{u, v\}$ is not adjacent to $\{x, y\}$. *Case 2: x and y are not adjacent in H_0.* H_0 has an even hole containing both x and y if and only if there are nodes u_x, v_x, u_y, v_y of H such that (1) $H_0[\{u_x, x, v_x\}]$ is path $u_x x v_x$ and $H_0[\{u_y, y, v_y\}]$ is path $u_y y v_y$, (2) $p(u_x, u_y) + p(v_x, v_y)$ is even, and (3) $P(u_x, u_y)$ and $P(v_x, v_y)$ are both disjoint and non-adjacent. The lemma is proved. □

4.2.2 2-joins and non-path 2-joins

We say that $V_1|V_2$ is a *2-join* [18, 40] of a graph H with *split* (X_1, Y_1, X_2, Y_2) if (1) V_1 and V_2 form a disjoint partition of $V(H)$ with $|V_1| \geq 3$ and $|V_2| \geq 3$, (2) X_1 and Y_1 (respectively, X_2 and Y_2) are disjoint non-empty subsets of V_1 (respectively, V_2), and (3) each node of X_1 is adjacent to each node of X_2, each node of Y_1 is adjacent to each node of Y_2, and there are no other edges between V_1 and V_2. See Figure 7(a) for an example.

Lemma 4.7 (Trotignon et al. [40, Lemma 3.2])**.** *If $V_1|V_2$ is a 2-join of a star-cutset-free graph H with split (X_1, Y_1, X_2, Y_2), then the following statements hold for each $i \in \{1, 2\}$.*

1. *Each connected component of $H[V_i]$ contains at least one node in X_i and at least one node in Y_i.*
2. *Each node of V_i has a neighbor in V_i.*
3. *Each node of X_i has a non-neighbor in Y_i. Each node of Y_i has a non-neighbor in X_i.*
4. *$|V_i| \geq 4$.*

A 2-join $V_1|V_2$ of H with split (X_1, Y_1, X_2, Y_2) is a *non-path 2-join* [39] of H if $H[V_1]$ is not a path between a node of X_1 and a node of Y_1 and $H[V_2]$ is not a path between a node of X_2 and a node of Y_2. For instance, the 2-join in Figure 7(a) is a non-path 2-join. (Non-path 2-joins are called 2-joins by da Silva and Vušković [20, §1.3].)

Lemma 4.8 (Charbit et al. [4, Theorem 4.1]). *Given an n-node connected graph H, it takes $O(n^4)$ time to either output a non-path 2-join of H together with a split or ensure that H has no non-path 2-joins.*

Lemma 4.9 (da Silva and Vušković [20, Corollary 1.3]). *A connected graph that is even-hole-free, star-cutset-free, and non-path-2-join-free is an extended clique tree.*

Combining Lemmas 4.6, 4.8, and 4.9, we have the following lemma.

Lemma 4.10. *Given an n-node star-cutset-free graph H, it takes $O(n^4)$ time to either (a) determine whether H has even holes or (b) obtain a non-path 2-join of H with a split.*

Proof. It takes $O(n^4)$ time to determine whether the graph H is an extended clique tree: For any set S of two or less nodes of H, it takes $O(n^2)$ time to obtain the biconnected components of subgraph $H \setminus S$ [23] and determine whether all of them are cliques. If H is an extended clique tree, Lemma 4.6 implies that it takes $O(n^4)$ time to determine whether H has even holes. If H is not an extended clique tree, Lemma 4.8 implies that it takes $O(n^4)$ time to either obtain a non-path 2-join of H with a split or ensure that H has no non-path 2-joins. If H has no non-path 2-joins, Lemma 4.9 implies that H has even holes. □

4.2.3 Parity-preserving blocks of decomposition for connected 2-joins

A 2-join $V_1|V_2$ with split (X_1, Y_1, X_2, Y_2) is *connected* [40] if, for each $i \in \{1, 2\}$, there is an induced path P_i of $H[V_i]$ between a node x_i of X_i and a node y_i of Y_i such that $V(P_i) \setminus \{x_i, y_i\} \subseteq V_i \setminus (X_i \cup Y_i)$. For instance, the 2-join $V_1|V_2$ in Figure 7(a) is connected. By Lemma 4.7(1), any 2-join of a star-cutset-free graph is connected with respect to any split.

Let $V_1|V_2$ be a connected 2-join of graph H with split (X_1, Y_1, X_2, Y_2). For each $i \in \{1, 2\}$, let P_i be a shortest induced path P_i of $H[V_i]$ between a node x_i of X_i and a node y_i of Y_i with $V(P_i) \setminus \{x_i, y_i\} \subseteq V_i \setminus (X_i \cup Y_i)$. If $|V(P_i)|$ is even (respectively, odd), then let $p_i = 4$ (respectively, $p_i = 5$). The *parity-preserving blocks of decomposition* [40] of H for 2-join $V_1|V_2$ with respect to split (X_1, Y_1, X_2, Y_2) are the following graphs H_1 and H_2.

- H_1 consists of (a) $H[V_1]$, (b) a p_2-path between nodes x_2 and y_2, (c) edges x_2x for all nodes x of X_1, and (d) edges y_2y for all nodes y of Y_1.
- H_2 consists of (a) $H[V_2]$, (b) a p_1-path between nodes x_1 and y_1, (c) edges x_1x for all nodes x of X_2, and (d) edges y_1y for all nodes y of Y_2.

See Figure 7(b) for an example of H_1 and H_2.

Lemma 4.11 (Trotignon and Vušković [40, Lemma 3.8]). *If $V_1|V_2$ is a connected 2-join of a star-cutset-free graph H with split (X_1, Y_1, X_2, Y_2), the parity-preserving blocks of decomposition H_1 and H_2 of H for $V_1|V_2$ with respect to (X_1, Y_1, X_2, Y_2) are star-cutset-free graphs such that H is even-hole-free if and only if both H_1 and H_2 are even-hole-free.*

Lemma 4.12. *Let H be an n-node m-edge star-cutset-free graph. Either one of the parity-preserving blocks H_1 and H_2 of decomposition for an arbitrary non-path 2-join of H with respect to an arbitrary split has at most n nodes and $m - 1$ edges.*

Proof. We prove the lemma for H_1. The proof for H_2 is similar. Let $V_1|V_2$ be the non-path 2-join. Let (X_1, Y_1, X_2, Y_2) be the split. Let P_2 be a shortest path of $H[V_2]$ between a node of X_2 and a node of Y_2. For the case that $|V(P_2)|$ is even, we have $p_2 = 4$. By Lemma 4.7(4), $|V_2| \geq 4$, implying $|V(H_1)| = n - |V_2| + p_2 \leq n$. By the following case analysis, H_1 has at most $m - 1$ edges.

- $|V(P_2)| \geq 6$: By $P_2 \subseteq H[V_2]$, $H[V_2]$ has at least five edges. Thus, H_1 has at most $m-2$ edges.
- $|V(P_2)| = 4$: Since $V_1|V_2$ is a non-path 2-join of H, $P_2 \subsetneq H[V_2]$. If $V(P_2) = V_2$, $H[V_2]$ has at least four edges. If $V(P_2) \subsetneq V_2$, Lemma 4.7(2) implies that $H[V_2]$ has at least four edges. Either way, H_1 has at most $m-1$ edges.
- $|V(P_2)| = 2$: Lemma 4.7(3) ensures $|X_2| \geq 2$ and $|Y_2| \geq 2$. Lemma 4.7(1) implies that $H[V_2]$ has at least two edges. By $|X_2| \geq 2$ and $|Y_2| \geq 2$, the number of edges between V_1 and V_2 in H is at least two more than the number of edges between V_1 and $V(H_1)\setminus V_1$ in H_1. Therefore, H_1 has at most $m-1$ edges.

As for the case that $|V(P_2)|$ is odd, we have $p_2 = 5$. The following case analysis shows that H_1 has at most n nodes and at most $m-1$ edges.

- $|V(P_2)| \geq 5$: By $|V_2| \geq 5$, we have $|V(H_1)| \leq n$. P_2 has at least four edges. Since $V_1|V_2$ is a non-path 2-join of H, $P_2 \subsetneq H[V_2]$. If $V(P_2) = V_2$, then $H[V_2]$ has at least five edges. If $V(P_2) \subsetneq V_2$, then Lemma 4.7(2) implies that $H[V_2]$ has at least five edges. Either way, H_1 has at most $m-1$ edges.
- $|V(P_2)| = 3$: By Lemma 4.7(4), the proper subset $Z = V_2\setminus V(P_2)$ of V_2 is non-empty. We know $Z \cap (X_2 \cup Y_2) \neq \varnothing$ or else $V(P_2)$ would be a star-cutset of H. Assume $Z \cap X_2 \neq \varnothing$ without loss of generality. Let B be an arbitrary connected component of $H[Z]$ with $B \cap X_2 \neq \varnothing$. We know that B is adjacent to Y_2 in H or else $N_H[x] \setminus Z$ would be a star-cutset of H, where x is the endpoint of P_2 in X_2. Since P_2 is a shortest path between a node of X_2 and a node of Y_2, at least one node of B is not in $X_2 \cup Y_2$. Therefore, $|V_2| \geq 5$, implying $|V(H_1)| \leq n$. Moreover, $H[V_2]$ has at least four edges. By $|X_2| \geq 2$, the number of edges between V_1 and V_2 in H is at least one more than the number of edges between V_1 and $V(H_1) \setminus V_1$ in H_1. Thus, H_1 has at most $m-1$ edges.

The lemma is proved. □

4.2.4 Proving Lemma 4.2

We now prove Lemma 4.2 by Lemmas 4.10, 4.11 and 4.12.

Proof of Lemma 4.2. Assume without loss of generality that the given n-node m-edge star-cutset-free graph H_0 is connected. Let set $\mathbb{H}$ initially consist of a single graph H_0. We then repeat the following loop until $\mathbb{H} = \varnothing$ or we output that H_0 has even holes. Let H be a graph in $\mathbb{H}$. *Case 1: H has at most 11 edges.* It takes $O(1)$ time to determine whether H has even holes. If H has even holes, we output that H_0 has even holes. Otherwise, we delete H from $\mathbb{H}$. *Case 2: H has at least 12 edges.* We first delete H from $\mathbb{H}$ and then apply Lemma 4.10 on H. If H has even holes, we output that H_0 has even holes. If we obtain a non-path 2-join $V_1|V_2$ of H with split (X_1, Y_1, X_2, Y_2), we add to $\mathbb{H}$ the parity-preserving blocks H_1 and H_2 of decomposition for $V_1|V_2$ with respect to (X_1, Y_1, X_2, Y_2). If the above loop stops with $\mathbb{H} = \varnothing$, we output that H_0 has no even holes.

The correctness of our algorithm follows immediately from Lemma 4.11. By Lemma 4.12, each graph ever in $\mathbb{H}$ throughout our algorithm has at most n nodes. By Lemma 4.10, each iteration of the loop takes $O(n^4)$ time. It remains to show that the loop halts in $O(m)$ iterations. Observe that each iteration increases the overall number of edges of the graphs in $\mathbb{H}$ by no more than 10.

Let $f(m)$ be the maximum number of iterations of the above loop in which Lemma 4.10 is applied. Lemma 4.12 implies

$$f(m) \leq \begin{cases} 0 & \text{if } m \leq 11 \\ \max\{1 + f(m_1) + f(m_2) \mid m_1, m_2 \leq m - 1, m_1 + m_2 \leq m + 10\} & \text{if } m \geq 12. \end{cases}$$

By induction on m, we show $f(m) \leq \max(m - 11, 0)$, which clearly holds for $m = 1, 2, \ldots, 11$. If $m \geq 12$, the inductive hypothesis implies

$$\begin{aligned} f(m) &\leq \max\{1 + \max(m_1 - 11, 0) + \max(m_2 - 11, 0) \mid m_1, m_2 \leq m - 1, m_1 + m_2 \leq m + 10\} \\ &\leq \max\{\max(m_1 + m_2 - 21, m_1 - 10, m_2 - 10, 1) \mid m_1, m_2 \leq m - 1, m_1 + m_2 \leq m + 10\} \\ &\leq \max(m - 11, m - 11, m - 11, 1) \\ &= \max(m - 11, 0). \end{aligned}$$

By $f(m) = O(m)$, the number of iterations of the above loop is $O(m)$. The lemma is proved. $\square$

5 Concluding remarks

For any class $\mathbb{G}$ of induced subgraphs, one can augment a recognition algorithm for $\mathbb{G}$-free graphs into a $\mathbb{G}$-detection algorithm for an n-node graph G with a factor-$O(n)$ increase in the time complexity by a node-deletion method: (1) Let $H = G$. (2) For each node v of G, if $H \setminus \{v\}$ is not $\mathbb{G}$-free, then let $H = H \setminus \{v\}$. (3) Output the resulting graph H. See, e.g., [41, §4] for the case that $\mathbb{G}$ consists of even holes. Thus, Theorem 1.1 immediately yields a detection algorithm that runs in time $O(m^3n^6) = O(n^{12})$. However, our $O(m^3n^5)$-time recognition algorithm can be augmented into an even-hole-detection algorithm without increasing the time complexity.

The combination of the proofs of Theorem 1.1 and Lemma 2.3 actually gives two algorithms. The first algorithm determines if G is both beetle-free and 4-hole-free. The second algorithm determines if a beetle-free and 4-hole-free graph G is also even-hole-free. We first describe how to augment the first algorithm into an $O(m^3n^5)$-time detection algorithm. Since it takes $O(n^4)$ time to detect a 4-hole in G, it suffices to show how to detect an even hole in a graph G with beetles in $O(m^3n^5)$ time. As stated in the proof of Lemma 2.3, for each of the $O(m^3n)$ choices of node b_4 and edges b_1b_5, b_2b_6, and b_3b_7, it takes $O(n^4)$ time via Lemma 3.1 to determine if b_4, b_1b_5, b_2b_6, and b_3b_7 are contained by a beetle B in which $\{b_1, b_2, b_3, b_4\}$ induces a diamond. Once we know that there exists a beetle B containing a particular choice of b_4, b_1b_5, b_2b_6, and b_3b_7, by an augmented version of Lemma 3.1 via the above node-deletion method, it takes $O(n^5)$ time to actually detect such a beetle B. Therefore, if G has beetles, it takes $O(m^3n) \cdot O(n^4) + O(n^5) = O(m^3n^5)$ time to find a beetle of G, in which an even hole of G can be detected in $O(n)$ time.

The second algorithm can also be augmented into an $O(m^3n^5)$-time detection algorithm for a beetle-free graph G that has even holes. By Lemma 2.3, we obtain in $O(m^3n^5)$ time a set $\mathbb{T}$ of $O(m^2n)$ trackers that satisfies Property 1. Since G has even holes, there must be a tracker $(H, u_1u_2u_2)$ of $\mathbb{T}$ such that H contains an even hole of G, which according to Lemma 2.4 can be found in time $O(m^2n) \cdot O(mn^4) = O(m^3n^5)$. By the proof of Lemma 2.4, H is ensured to have even holes in two ways. (1) If it is ensured through completing Task 1 of Lemma 4.1, the proof of Lemma 4.1 actually gives a constructive proof for the existence of an even hole of H, which is also an even hole of G. (2) If it is ensured through completing Task 3 of Lemma 4.1 and then by

Lemma 4.2, we have a star-cutset-free induced subgraph H' of H that has even holes. We then apply the above node-deletion method on H' using Lemma 4.2 to detect in $O(mn^5)$ time an even hole of H', which is also an even hole of H and G. Therefore, if G is a 4-hole-free and beetle-free graph that has even holes, it takes time $O(m^3n^5)+O(mn^5) = O(m^3n^5)$ to output an even hole of G. Combining the above two detection algorithms, we have an $O(m^3n^5)$-time algorithm to output an even hole in an n-node m-edge graph with even holes.

Acknowledgement

We thank Gerard J. Chang for discussion.

References

[1] L. Addario-Berry, M. Chudnovsky, F. Havet, B. Reed, and P. Seymour. Bisimplicial vertices in even-hole-free graphs. *Journal of Combinatorial Theory, Series B*, 98(6):1119–1164, 2008.

[2] D. Bienstock. On the complexity of testing for odd holes and induced odd paths. *Discrete Mathematics*, 90(1):85–92, 1991.

[3] H.-C. Chang and H.-I. Lu. A faster algorithm to recognize even-hole-free graphs. In *Proceedings of the 23rd Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1286–1297, 2012.

[4] P. Charbit, M. Habib, N. Trotignon, and K. Vušković. Detecting 2-joins faster. *Journal of Discrete Algorithms*, 17:60–66, 2012.

[5] M. Chudnovsky, G. Cornuéjols, X. Liu, P. Seymour, and K. Vušković. Recognizing Berge graphs. *Combinatorica*, 25(2):143–186, 2005.

[6] M. Chudnovsky, K.-i. Kawarabayashi, and P. Seymour. Detecting even holes. *Journal of Graph Theory*, 48(2):85–111, 2005.

[7] M. Chudnovsky, N. Robertson, P. Seymour, and R. Thomas. The strong perfect graph theorem. *Annals of Mathematics*, 164(1):51–229, 2006.

[8] M. Chudnovsky, N. Robertson, P. D. Seymour, and R. Thomas. K_4-free graphs with no odd holes. *Journal of Combinatorial Theory, Series B*, 100(3):313–331, 2010.

[9] M. Chudnovsky and P. Seymour. The three-in-a-tree problem. *Combinatorica*, 30(4):387–417, 2010.

[10] V. Chvátal. Star-cutsets and perfect graphs. *Journal of Combinatorial Theory, Series B*, 39:189–199, 1985.

[11] M. Conforti, G. Cornuéjols, A. Kapoor, and K. Vušković. Finding an even hole in a graph. In *Proceedings of the 38th Annual Symposium on Foundations of Computer Science*, pages 480–485, 1997.

[12] M. Conforti, G. Cornuéjols, A. Kapoor, and K. Vušković. Even and odd holes in cap-free graphs. *Journal of Graph Theory*, 30(4):289–308, 1999.

[13] M. Conforti, G. Cornuéjols, A. Kapoor, and K. Vušković. Triangle-free graphs that are signable without even holes. *Journal of Graph Theory*, 34(3):204–220, 2000.

[14] M. Conforti, G. Cornuéjols, A. Kapoor, and K. Vušković. Even-hole-free graphs Part I: Decomposition theorem. *Journal of Graph Theory*, 39(1):6–49, 2002.

[15] M. Conforti, G. Cornuéjols, A. Kapoor, and K. Vušković. Even-hole-free graphs Part II: Recognition algorithm. *Journal of Graph Theory*, 40(4):238–266, 2002.

[16] M. Conforti, G. Cornuéjols, X. Liu, K. Vušković, and G. Zambelli. Odd hole recognition in graphs of bounded clique size. *SIAM Journal on Discrete Mathematics*, 20:42–48, 2006.

[17] M. Conforti, G. Cornuéjols, and K. Vušković. Decomposition of odd-hole-free graphs by double star cutsets and 2-joins. *Discrete Applied Mathematics*, 141(1-3):41–91, 2004.

[18] G. Cornuéjols and W. Cunningham. Compositions for perfect graphs. *Discrete Mathematics*, 55(3):245–254, 1985.

[19] M. V. G. da Silva and K. Vušković. Triangulated neighborhoods in even-hole-free graphs. *Discrete Mathematics*, 307(9-10):1065–1073, 2007.

[20] M. V. G. da Silva and K. Vušković. Decomposition of even-hole-free graphs with star cutsets and 2-joins. *Journal of Combinatorial Theory, Series B*, 103(1):144–183, 2013.

[21] D. Défossez. Complexity of clique-coloring odd-hole-free graphs. *Journal of Graph Theory*, 62(2):139–156, 2009.

[22] M. Farber. On diameters and radii of bridged graphs. *Discrete Mathematics*, 73(3):249–260, 1989.

[23] J. E. Hopcroft and R. E. Tarjan. Efficient algorithms for graph manipulation. *Communications of the ACM*, 16(6):372–378, 1973.

[24] W.-L. Hsu. Recognizing planar perfect graphs. *Journal of the ACM*, 34(2):255–288, 1987.

[25] D. S. Johnson. The NP-completeness column. *ACM Transactions on Algorithms*, 1(1):160–176, 2005.

[26] K.-i. Kawarabayashi and Y. Kobayashi. Algorithms for finding an induced cycle in planar graphs and bounded genus graphs. In *Proceedings of the 20th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1146–1155, 2009.

[27] K.-i. Kawarabayashi and Y. Kobayashi. Algorithms for finding an induced cycle in planar graphs. *Combinatorica*, 30(6):715–734, 2010.

[28] W. S. Kennedy and A. D. King. Finding a smallest odd hole in a claw-free graph using global structure. *Discrete Applied Mathematics*, to appear, 2013.

[29] T. Kloks, H. Müller, and K. Vušković. Even-hole-free graphs that do not contain diamonds: A structure theorem and its consequences. *Journal of Combinatorial Theory, Series B*, 99(5):733–800, 2009.

[30] F. Maffray and N. Trotignon. Algorithms for perfectly contractile graphs. *SIAM Journal on Discrete Mathematics*, 19(3):553–574, 2005.

[31] S. D. Nikolopoulos and L. Palios. Hole and antihole detection in graphs. In *Proceedings of the 15th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 850–859, 2004.

[32] S. D. Nikolopoulos and L. Palios. Detecting holes and antiholes in graphs. *Algorithmica*, 47(2):119–138, 2007.

[33] O. Porto. Even induced cycles in planar graphs. In *Proceedings of the 1st Latin American Symposium on Theoretical Informatics*, Lecture Notes in Computer Science 583, pages 417–429. Springer, 1992.

[34] D. J. Rose and R. E. Tarjan. Algorithmic aspects of vertex elimination. In *Proceedings of the 7th Annual ACM Symposium on Theory of Computing*, pages 245–254, 1975.

[35] D. J. Rose, R. E. Tarjan, and G. S. Lueker. Algorithmic aspects of vertex elimination on graphs. *SIAM Journal on Computing*, 5(2):266–283, 1976.

[36] S. Shrem, M. Stern, and M. C. Golumbic. Smallest odd holes in claw-free graphs. In C. Paul and M. Habib, editors, *Proceedings of the 35th International Workshop on Graph-Theoretic Concepts in Computer Science*, Lecture Notes in Computer Science 6410, pages 329–340. Springer, 2009.

[37] A. Silva, A. A. da Silva, and C. L. Sales. A bound on the treewidth of planar even-hole-free graphs. *Discrete Applied Mathematics*, 158(12):1229–1239, 2010.

[38] R. E. Tarjan and M. Yannakakis. Simple linear-time algorithms to test chordality of graphs, test acyclicity of hypergraphs, and selectively reduce acyclic hypergraphs. *SIAM Journal on Computing*, 13:566–579, 1984.

[39] N. Trotignon. Decomposing berge graphs and detecting balanced skew partitions. *Journal of Combinatorial Theory, Series B*, 98(1):173–225, 2008.

[40] N. Trotignon and K. Vušković. Combinatorial optimization with 2-joins. *Journal of Combinatorial Theory, Series B*, 102(1):153–185, 2012.

[41] K. Vušković. Even-hole-free graphs: A survey. *Applicable Analysis and Discrete Mathematics*, 4(2):219–240, 2010.

[42] Y. Zwols. K_4-free graphs with no odd hole: Even pairs and the circular chromatic number. *Journal of Graph Theory*, 65(4):303–322, 2010.