Algorithmica
https://doi.org/10.1007/s00453-018-0409-6

The Sandwich Problem for Decompositions and Almost Monotone Properties

Maria Chudnovsky[1] · **Celina M. H. de Figueiredo**[2] · **Sophie Spirkl**[1]

Received: 21 March 2017 / Accepted: 16 January 2018

Abstract We consider the graph sandwich problem and introduce almost monotone properties, for which the sandwich problem can be reduced to the recognition problem. We show that the property of containing a graph in $\mathcal{C}$ as an induced subgraph is almost monotone if $\mathcal{C}$ is the set of thetas, the set of pyramids, or the set of prisms and thetas. We show that the property of containing a hole of length $\equiv j \mod n$ is almost monotone if and only if $j \equiv 2 \mod n$ or $n \leq 2$. Moreover, we show that the imperfect graph sandwich problem, also known as the Berge trigraph recognition problem, can be solved in polynomial time. We also study the complexity of several graph decompositions related to perfect graphs, namely clique cutset, (full) star cutset, homogeneous set, homogeneous pair, and 1-join, with respect to the partitioned and unpartitioned probe problems. We show that the clique cutset and full star cutset unpartitioned probe problems are *NP*-hard. We show that for these five decompositions, the partitioned probe problem is in P, and the homogeneous set, 1-join, 1-join in the complement, and full star cutset in the complement unpartitioned probe problems can be solved in polynomial time as well.

Maria Chudnovsky was supported by National Science Foundation Grant DMS-1550991 and US Army Research Office Grant W911NF-16-1-0404. Celina M. H. de Figueiredo was supported by Conselho Nacional de Desenvolvimento Científico e Tecnológico CNPq Grant 303622/2011-3.

✉ Sophie Spirkl
sspirkl@princeton.edu

Maria Chudnovsky
mchudnov@princeton.edu

Celina M. H. de Figueiredo
celina@cos.ufrj.br

1 Princeton University, Princeton, NJ 08544, USA

2 COPPE, Universidade Federal do Rio de Janeiro, Rio de Janeiro, Brazil

Published online: 19 January 2018

Keywords Graph theory · Graph algorithms · Sandwich problem · Probe problem · Trigraphs · Graph decompositions

Mathematics Subject Classification 05C85

1 Introduction

All graphs in this paper are finite and simple. Let G be a graph. G^c denotes the *complement* of G, obtained from G by replacing each edge with a non-edge and vice versa. For $X \subseteq V(G)$, $G|X$ denotes the induced subgraph of G with vertex set X. For $X, Y \subseteq V(G)$ with $X \cap Y = \emptyset$, we say that X is *complete* to Y if for all $x \in X$, $y \in Y$, $xy \in E(G)$; we say that X is *anticomplete* to Y if for all $x \in X$, $y \in Y$, $xy \notin E(G)$. For $v \in V(G)$, $X \subseteq V(G) \setminus \{v\}$, we say that v is *complete* (*anticomplete*) to X if $\{v\}$ is complete (anticomplete) to X.

Let $G_1 = (V_1, E_1)$, $G_2 = (V_2, E_2)$, then G_2 is a *supergraph* of G_1 if $V_1 = V_2$ and $E_1 \subseteq E_2$. A pair (G_1, G_2) of graphs such that G_2 is a supergraph of G_1 is called a *sandwich instance*. A graph G is called a *sandwich graph* for the sandwich instance (G_1, G_2) if G_2 is a supergraph of G and G is a supergraph of G_1. For a graph G and a set E' of edges with both endpoints in $V(G)$, $G \cup E'$ denotes the supergraph $G' = (V(G), E(G) \cup E')$ of G, and $G \setminus E'$ denotes the graph $G'' = (V(G), E(G) \setminus E')$, and G is a supergraph of G''.

Let $\mathcal{P}$ be a graph property. We define the complementary property $\mathcal{P}^c$ by saying that G satisfies $\mathcal{P}^c$ if and only if G^c satisfies $\mathcal{P}$.

The $\mathcal{P}$ RECOGNITION PROBLEM is the problem of deciding whether a given graph G satisfies $\mathcal{P}$. The $\mathcal{P}$ SANDWICH PROBLEM is the following: For a given sandwich instance (G_1, G_2), does there exist a sandwich graph G for (G_1, G_2) such that G satisfies $\mathcal{P}$? This generalization of the recognition problem was introduced by Golumbic and Shamir [23]. The sandwich problem becomes the recognition problem when $G_1 = G_2$, and thus, if the $\mathcal{P}$ recognition problem is *NP*-hard, so is the $\mathcal{P}$ sandwich problem.

Sandwich problems have attracted much attention lately, see [4,16,18,23,24,32,33]. Starting with [24], research has focused on the sandwich problem for subclasses of perfect graphs, and for decompositions related to perfect graphs. The complexity of the perfect graph sandwich problem remains one of the most prominent open questions in this area.

Let G, G' be a pair of graphs such that G' is a supergraph of G. Then G' is a (P, N)-*probe graph* for G if (P, N) is a partition of $V(G)$, N is a stable set in G, and every edge in $E(G') \setminus E(G)$ has both of its endpoints in N.

For a graph property $\mathcal{P}$, a graph $G = (V, E)$ is a $\mathcal{P}$ *probe graph with partition* (P, N) if there exists a (P, N)-probe graph G' for G such that G' satisfies $\mathcal{P}$. A graph G is a $\mathcal{P}$ *probe graph* if there exists a partition (P, N) of its vertex set such that G is a $\mathcal{P}$ probe graph with partition (P, N). The vertices in P are called *probes*, and the vertices in N are called *non-probes*.

For a graph property $\mathcal{P}$, the $\mathcal{P}$ PARTITIONED PROBE PROBLEM is the following: Given a graph $G = (V, E)$, and a stable set $N \subseteq V$, is G a $\mathcal{P}$ probe graph with partition $(V\backslash N, N)$? The partitioned probe problem was first introduced in [28,36] for interval graphs because of its applications to the physical mapping of DNA.

The $\mathcal{P}$ partitioned probe problem with input graph $G = (V, E)$ and stable set $N \subseteq V$ is a special case of the $\mathcal{P}$ sandwich problem in which $E(G_1) = E$ and the edges in $E(G_2)\backslash E(G_1)$ are precisely the edges between all pairs of distinct vertices in N.

The complexity of the $\mathcal{P}^c$ sandwich problem is the same as the complexity of the $\mathcal{P}$ sandwich problem, because an instance (G_1, G_2) is a YES instance for the former if and only if (G_2^c, G_1^c) is a YES instance for the latter. The same is true for the $\mathcal{P}$ partitioned probe problem: A graph G with partition (P, N) is a YES instance for the $\mathcal{P}$ partitioned probe problem if and only if the graph G' arising from G^c by removing all edges with both endpoints in N with the partition (P, N) is a YES instance for the $\mathcal{P}^c$ partitioned probe problem.

Let $\mathcal{P}$ be a graph property. The $\mathcal{P}$ UNPARTITIONED PROBE PROBLEM is the following: Given a graph G, is G a $\mathcal{P}$ probe graph? We also consider the $\mathcal{P}$ UNPARTITIONED PROBE PROBLEM IN THE COMPLEMENT: Given a graph G, is G^c a $\mathcal{P}^c$ probe graph? In other words, in the unpartitioned probe problem, the goal to decide whether there is a stable set N in G and a set of edges E' with both endpoints in N such that $G \cup E'$ satisfies $\mathcal{P}$, whereas in the unpartitioned probe problem in the complement, the goal to decide whether there is a clique N in G and a set of edges E' with both endpoints in N such that $G\backslash E'$ satisfies $\mathcal{P}$. Therefore, these problems are not equivalent in general, and indeed we will show an example (containing a full star cutset) for which the unpartitioned probe problem is *NP*-hard, but the unpartitioned probe problem in the complement is in P.

The partitioned and unpartitioned probe problems have been studied extensively, see for example [2,14,25,28,36]. Couto, Faria, Gravier and Klein [14] conjectured that the perfect partitioned and unpartitioned probe problems can be solved in polynomial time, and proved that if the perfect unpartitioned probe problem can be solved in polynomial time, this also follows for the partitioned case.

This paper is organized as follows: In Sect. 2, we show that the sandwich problem can be reduced to the recognition problem for almost monotone properties, and we prove that several properties related to containing an induced subgraph from a certain set of graphs are almost monotone. In particular, we give a polynomial-time algorithm for the recognition of Berge trigraphs. In Sect. 3, we consider several decompositions that are related to the study of perfect graphs, and we study the hardness of testing for these decompositions for the partitioned probe problem and the unpartitioned probe problem in the graph and in the complement. In Sect. 3.1, we present resulting polynomial-time algorithms, and in Sect. 3.2, we give *NP*-hardness results. In Sect. 4, we mention some open problems.

2 Almost Monotone Properties

A property $\mathcal{P}$ of graphs is *ancestral* if for all $G = (V, E)$ that satisfy $\mathcal{P}$ and $E' \supseteq E$, $G' = (V, E')$ also satisfies $\mathcal{P}$. It is *hereditary* if for all $G = (V, E)$ that satisfy $\mathcal{P}$ and $E' \subseteq E$, $G' = (V, E')$ also satisfies $\mathcal{P}$. If a property is either ancestral or hereditary, it is called *monotone*. If a property $\mathcal{P}$ is ancestral, then $\mathcal{P}^c$ is hereditary, and vice versa. For monotone properties, the sandwich problem reduces to the recognition problem for either G_1 or G_2. Since the partitioned probe problem is a special case of the sandwich problem, it follows that this holds for the partitioned probe problem as well. Moreover, the unpartitioned probe problem for a hereditary property $\mathcal{P}$ with input G is the same as the $\mathcal{P}$ recognition problem with input G, and the unpartitioned probe problem in the complement for an ancestral property $\mathcal{P}$ with input G is the same as the $\mathcal{P}^c$ recognition problem with input G^c.

In the following, we define a more general notion of monotonicity, which allows us to reduce solving the sandwich problem to solving a polynomial number of recognition problems in this case.

A property $\mathcal{P}$ of graphs is *k-edge monotone* if for all sandwich instances (G_1, G_2), if there exists a sandwich graph G that satisfies $\mathcal{P}$, then there exists a sandwich graph G' that satisfies $\mathcal{P}$ with the additional property that $|E(G')\backslash E(G_1)| \leq k$ or $|E(G_2)\backslash E(G')| \leq k$.

A property $\mathcal{P}$ of graphs is *k-vertex monotone* if for all sandwich instances (G_1, G_2), if there exists a sandwich graph G that satisfies $\mathcal{P}$, then there exists a sandwich graph G' that satisfies $\mathcal{P}$ and a set $S \subseteq V(G)$ satisfying $|S| \leq k$ and such that for $V_1 = \{v \in V(G) : N_{G'}(v)\backslash S = N_{G_1}(v)\backslash S\}$ and $V_2 = \{v \in V(G) : N_{G'}(v)\backslash S = N_{G_2}(v)\backslash S\}$ we have $V_1 \cup V_2 = V(G)$ and $V(G)\backslash V_1 \subseteq S$ or $V(G)\backslash V_2 \subseteq S$.

Clearly, any monotone property is 0-edge monotone and 0-vertex monotone. We also remark the following simple consequence of these definitions.

Lemma 1 *If a property $\mathcal{P}$ is k-edge monotone, it is $2k$-vertex monotone. The converse is not true in general.*

Lemma 2 *Let $\mathcal{P}$ be a k-edge monotone property, then the $\mathcal{P}$ sandwich problem for a sandwich instance (G_1, G_2) with $|V(G_1)| = n$ can be decided by solving the $\mathcal{P}$ recognition problem for $\mathcal{O}(kn^{2k})$ graphs.*

Proof If there exists a sandwich graph that satisfies $\mathcal{P}$, then there exists a sandwich graph G with $|E(G)\backslash E(G_1)| \leq k$ or $|E(G_2)\backslash E(G)| \leq k$. Thus, it suffices to check for all subsets $F \subseteq E(G_2)\backslash E(G_1)$ with $|F| \leq k$ if $(V(G_1), E(G_1) \cup F)$ or $(V(G_2), E(G_2)\backslash F)$ satisfies $\mathcal{P}$. Since there are $\mathcal{O}(n^2)$ edges, it follows that there are $\mathcal{O}(kn^{2k})$ sets F to consider. □

Lemma 3 *Let $\mathcal{P}$ be a k-vertex monotone property, then the $\mathcal{P}$ sandwich problem for a sandwich instance (G_1, G_2) with $|V(G_1)| = n$ can be decided by solving the $\mathcal{P}$ recognition problem for $\mathcal{O}(kn^k 2^{\binom{k+1}{2}})$ graphs.*

Proof It suffices to solve the recognition problem for all sandwich graphs G with a set $S \subseteq V(G)$ satisfying $|S| \leq k$ and such that for $V_1 = \{v \in V(G) : N_G(v)\backslash S = N_{G_1}(v)\backslash S\}$ and $V_2 = \{v \in V(G) : N_G(v)\backslash S = N_{G_2}(v)\backslash S\}$ we have $V_1 \cup V_2 = V(G)$ and $V(G)\backslash V_1 \subseteq S$ or $V(G)\backslash V_2 \subseteq S$. There are $\mathcal{O}(kn^k)$ sets $S \subseteq V$ of size at most k, and two choices such that either $V(G)\backslash V_1 \subseteq S$ or $V(G)\backslash V_2 \subseteq S$. This determines all edges in G with both endpoints not in S. For each vertex in S, we choose whether it is in V_1 or V_2. There are 2^k options for this, and they determine all edges in G with exactly one endpoint in S. Finally, we choose any subset of the edges in $E(G_2)\setminus E(G_1)$ with both endpoints in S to be in G; there are at most $2^{\binom{k}{2}}$ possibilities. Thus, the number of possible graphs G is $\mathcal{O}(kn^k 2^{\binom{k+1}{2}})$. □

Let $\mathcal{C}$ be a set of graphs. We say that G is $\mathcal{C}$*-free* if no induced subgraph of G is isomorphic to a graph in $\mathcal{C}$. We say that $\mathcal{C}$ is *almost edge monotone* (*almost vertex monotone*) if there exists a k such that the property of not being $\mathcal{C}$-free is k-edge monotone (k-vertex monotone). If $\mathcal{C}$ is almost edge monotone or almost vertex monotone, so is the set of graphs whose complement is in $\mathcal{C}$. Moreover, any finite set $\mathcal{C}$ of graphs is almost edge monotone.

The following lemma is a simple consequence of the definition of almost monotone properties.

Lemma 4 *Let $\mathcal{C}, \mathcal{C}'$ be almost edge (vertex) monotone sets of graphs. Then their union is almost edge (vertex) monotone.*

An induced cycle C_k with $k \geq 4$ vertices is called a *hole*; it is called an *odd hole* if k is odd, and an *even hole* if k is even. An *antihole* is the complement of a hole. It is an *odd antihole* if its complement is an odd hole, and an *even antihole* otherwise.

Lemma 5 *Let $\mathcal{C}$ be the set of odd holes. Then $\mathcal{C}$ is almost edge monotone; in particular, the property of containing an odd hole is* 5*-edge monotone. Consequently, the property of containing an odd antihole is also* 5*-edge monotone.*

Proof Let (G_1, G_2) be a sandwich instance such that there is a sandwich graph for (G_1, G_2) that contains an odd hole. Let G be the sandwich graph for (G_1, G_2) with $|E(G_2)\setminus E(G)|$ minimum subject to G containing an odd hole, and let C be an odd hole in G. There is no edge in $E(G_2)\setminus E(G)$ with at least one endpoint not in $V(C)$, since adding such an edge to G would preserve the odd hole C. Our goal is to prove that $|E(G_2)\setminus E(G)| \leq 5$.

Let $v_1, \ldots, v_k$ denote the vertices of C in order along C. All edges in $E(G_2)\setminus E(G)$ have both endpoints in C. For each edge $e \in E(G_2)\setminus E(G)$, adding e to G splits C into two smaller induced cycles whose number of edges sums to $k+2$. Therefore, one of these cycles is odd, but since it is not an odd hole, it follows that it is a triangle. Let $v(e)$ denote the vertex of this triangle that is not an endpoint of e. Clearly, $v(e) = v(e')$ implies that $e = e'$. Suppose first that there are two edges $e_1, e_2 \in E(G)$ such that $v(e_1)$ and $v(e_2)$ are non-adjacent, then $\{v_1, \ldots, v_k\} \setminus \{v(e_1), v(e_2)\}$ induces an odd cycle in $G \cup \{e_1, e_2\}$ which is not an odd hole, and therefore, this cycle is a triangle. This implies that $k = 5$, and thus there are at most five edges connecting two non-adjacent vertices in C, which implies the result that $|E(G_2)\setminus E(G)| \leq 5$. Thus, we may assume that for all distinct $e_1, e_2 \in E(G_2)\setminus E(G)$, $v(e_1)$ is adjacent to $v(e_2)$. This implies that $\{v(e) : e \in E(G_2)\setminus E(G)\}$ is a clique in C, and since C has clique number two, we conclude in this case that $|E(G_2)\setminus E(G)| \leq 2$. □

A graph is *Berge* if it contains no odd hole and no odd antihole as an induced subgraph. A graph G is *perfect* if for each induced subgraph H of G, the clique number of H equals the chromatic number of H. The strong perfect graph theorem [8], first conjectured in [1], states that a graph is perfect if and only if it is Berge. An important tool for the proof of this theorem are Berge trigraphs, which were introduced by the first author in [5,7]. A *trigraph* is defined as a sandwich pair (G_1, G_2). A trigraph (G_1, G_2) satisfies a property $\mathcal{P}$ if there is no sandwich graph G for (G_1, G_2) which does not satisfy $\mathcal{P}$. In this sense, trigraphs are complementary to sandwich graphs.

It is known that Berge graphs can be recognized in polynomial time [6], but the recognition of Berge trigraphs was previously open. Note that it is not known if the recognition of graphs containing an odd hole is in P.

Corollary 1 *Recognizing Berge trigraphs is in P; equivalently, the imperfect sandwich problem is in P.*

Proof Note that (G_1, G_2) is a Berge trigraph if and only if (G_1, G_2) is a No instance for the imperfect sandwich problem.

By Lemma 5, the property of containing an odd hole is 5-edge monotone, and the property of containing an odd antihole is 5-edge monotone as well. Let (G_1, G_2) be a trigraph. Suppose that (G_1, G_2) is not Berge. Then there is a sandwich graph for (G_1, G_2) which contains an odd hole or an odd antihole, and consequently there is a sandwich graph G which differs from either G_1 or G_2 by at most five edges, and which is not Berge. We can check whether or not every such sandwich graph is Berge by using the Berge graph recognition algorithm. If we find a sandwich graph that is not Berge, then (G_1, G_2) is not a Berge trigraph. If all of the graphs we checked are Berge, then no sandwich graph for (G_1, G_2) contains an odd hole or an odd antihole, and consequently, (G_1, G_2) is a Berge trigraph. □

A *pyramid* is a graph consisting of distinct vertices a, b_1, b_2, b_3 and three induced internally vertex-disjoint paths P_1, P_2, P_3, each consisting of at least one edge, such that

- for $i = 1, 2, 3$, P_i has endpoints a and b_i; and
- for distinct $i, j \in \{1, 2, 3\}$, $b_i b_j$ is an edge, and this is the only edge between $V(P_i) \setminus \{a\}$ and $V(P_j) \setminus \{a\}$; and
- a is adjacent to at most one of b_1, b_2, b_3.

The vertex a is called the *apex* of the pyramid, and $\{b_1, b_2, b_3\}$ is called the *base* of the pyramid. P_1, P_2, P_3 are called the *paths* of the pyramid. A graph *contains a pyramid* if it contains a pyramid as an induced subgraph.

The recognition algorithm for Berge graphs in [6] uses a recognition algorithm for pyramid-free graphs as a subroutine. In particular, the recognition of graphs containing a pyramid is in P [6]. Pyramids are studied in relation to perfect graphs, because if a graph contains a pyramid, it contains an odd hole.

Theorem 1 *Let $\mathcal{C}$ be the set of all pyramids. Then $\mathcal{C}$ is almost vertex monotone.*

Proof Let (G_1, G_2) be a sandwich instance which is a Yes instance for the property of containing a pyramid. Let G be a sandwich graph for (G_1, G_2) with $|E(G_2) \setminus E(G)|$ minimum subject to G containing a pyramid P. Let $\{b_1, b_2, b_3\}$ be the base of P, and let a be the apex of P; let P_1, P_2, P_3 be the paths of P. Let S' be the set of vertices of P adjacent to at least one of $\{b_1, b_2, b_3, a\}$, and let $S = S' \cup \{b_1, b_2, b_3, a\}$. Then $|S| \leq 10$. Let G' be the sandwich graph with vertex set $V(G_1)$ in which $N_{G'}(v) \setminus S = N_{G_2}(v) \setminus S$

for all $v \in V(G) \setminus \{b_1, b_2, b_3, a\}$, and $N_{G'}(v) \setminus S = N_{G_1}(v) \setminus S$ for $v \in \{b_1, b_2, b_3, a\}$, and for $x, y \in S$, $xy \in E(G')$ if and only if x and y are adjacent in P. We claim that P is a pyramid in G', which then implies the result of the lemma.

Suppose for a contradiction that P is not a pyramid in G'. If some edge e of P is not an edge of G', then $e \in E(G_2)$, and so by definition of G', e has exactly one endpoint in S; consequently e is incident with $\{b_1, b_2, b_3, a\}$. But then the other endpoint of e is in S as well, contradicting the definition of G'. Thus, there is an edge $e = xy$ of G' with $x, y \in V(P)$ that is not an edge of P, and therefore $e \notin E(G)$, and so $e \in E(G_2) \setminus E(G_1)$. By definition of G', $x, y \notin \{a, b_1, b_2, b_3\}$. Suppose that $x, y \in V(P_i)$ for some $i \in \{1, 2, 3\}$, and let C denote the unique cycle in $(G|P_i) \cup \{xy\}$. Then $P' = P \setminus (V(C) \setminus \{x, y\}) \cup \{xy\}$ is an induced pyramid in $G \cup \{xy\}$, which contradicts the definition of G.

By symmetry, we may assume that $x \in V(P_1)$ and $y \in V(P_2)$, and $y \notin S$. Now let P_3' denote the subpath of P_1 from x to a, let P_2' denote the subpath of P_2 from y to b_2, and let P_1' denote the subpath of P_1 from x to b_1. Let Q denote the subpath of P_2 from a to y, and note that since $y \notin S$, it follows that Q contains more than one edge. Then $P \setminus (V(Q) \setminus \{a, y\}) \cup \{xy\}$ induces a pyramid with paths P_1', xyP_2', $P_3'P_3$, apex x and base $\{b_1, b_2, b_3\}$ in $G \cup \{xy\}$. Note that x is non-adjacent to b_2, b_3, where $P_3'P_3$ denotes the concatenation of the paths P_3' and P_3. This is a contradiction to the definition of G. Thus, P is a pyramid in G', and the result follows. □

A *theta* is a graph consisting of two distinct non-adjacent vertices a, b and three induced internally vertex-disjoint paths P_1, P_2, P_3 with ends a and b such that for all distinct $i, j \in \{1, 2, 3\}$, $V(P_i) \setminus \{a, b\}$ is anticomplete to $V(P_j) \setminus \{a, b\}$; the vertices a, b are the *ends* of the theta, and P_1, P_2, P_3 are the *paths* of the theta. A *prism* is a graph consisting of distinct vertices $a_1, a_2, a_3, b_1, b_2, b_3$ and three induced vertex-disjoint paths P_1, P_2, P_3 such that

- for $i = 1, 2, 3$, P_i has endpoints a_i and b_i; and
- $\{a_1, a_2, a_3\}$ is a clique and $\{b_1, b_2, b_3\}$ is a clique; and
- for distinct $i, j \in \{1, 2, 3\}$, there are no edges between $V(P_i) \setminus \{a_i, b_i\}$ and $V(P_j) \setminus \{a_j, b_j\}$.

The sets $\{a_1, a_2, a_3\}$ and $\{b_1, b_2, b_3\}$ are called the *triangles* of the prism, and P_1, P_2, P_3 are called the *paths* of the prism.

Testing if a graph contains a theta as an induced subgraph is in P [10], and testing if a graph contains a theta or a prism as an induced subgraph is in P as well [9], but testing if a graph contains a prism is *NP*-hard [27]. The theta-free sandwich problem is *NP*-hard [16], and as a consequence of [27], the prism-free sandwich problem and the not prism-free sandwich problem are *NP*-hard as well.

Theorem 2 *Let $\mathcal{C}$ be the set of thetas, and let $\mathcal{C}'$ be the set of thetas and prisms. Both $\mathcal{C}$ and $\mathcal{C}'$ are almost vertex monotone.*

Proof Let (G_1, G_2) be a sandwich instance, and suppose that some sandwich graph contains a theta. Let G be a sandwich graph with $|E(G_2) \setminus E(G)|$ minimum subject to G containing a theta. Let P be a theta in G with ends a, b and paths P_1, P_2, P_3. Let S be the set of vertices of P at distance at most one from $\{a, b\}$ in P. Let S'

be the set of vertices of P at distance at most two from $\{a, b\}$ in P; it follows that $|S'| \leq 14$. We claim that $E(G_2)\backslash E(G)$ does not contain an edge with both endpoints in $V(P)\backslash \{S\}$. Suppose for a contradiction that it does contain such an edge, say $e = xy$. If both endpoints of e are contained in the same path P_i, then we can replace P_i by a shorter path containing e and still have a theta in $G \cup \{e\}$; this is a contradiction. Therefore, there exist distinct $i, j \in \{1, 2, 3\}$ such that $x \in V(P_i)$ and $y \in V(P_j)$. Let $\{k\} = \{1, 2, 3\} \setminus \{i, j\}$. Let Q_i be the subpath of P_i with endpoints x and b; let Q_j be the concatenation of xy and the subpath of P_j from y to b; let Q_k be the concatenation of the subpath of P_i from x to a and P_k. Then $G \cup \{e\}$ contains a theta with ends x and b and paths Q_1, Q_2, Q_3. This is a contradiction, and thus our claim is proved.

Let G' be the graph with vertex set $V(G_1)$ and $N_{G'}(x)\backslash S' = N_{G_2}(x)\backslash S'$ for all $x \in V(G')\backslash S$, $N_{G'}(x)\backslash S' = N_{G_1}(x)\backslash S'$ for all $x \in S$, and for $x, y \in S'$, let xy be an edge if and only if xy is an edge in P. We claim that G' contains P as an induced subgraph. This follows because G' contains every edge of P, and if G' contains an edge e with endpoints in P which is not an edge in P, then e has an endpoint x in S by the claim proved above. The other endpoint, say y, of e is not in S', because by definition $G'|S' = P|S'$. But $N_{G'}(x)\backslash S' = N_{G_1}(x)\backslash S'$ for all $x \in S$, and so $y \in N_{G_1}(x)$, and thus $xy \in E(G)$ and $xy \in E(P)$, a contradiction. This proves that $\mathcal{C}$ is almost vertex monotone.

To prove that $\mathcal{C}'$ is almost vertex monotone, we may assume that no sandwich graph for (G_1, G_2) contains a theta. Suppose that some sandwich graph for (G_1, G_2) contains a prism, and let G be the sandwich graph with $|E(G_2)\backslash E(G)|$ minimum subject to G containing a prism; let P be a prism in G. Let $\{a_1, a_2, a_3\}$ and $\{b_1, b_2, b_3\}$ be the triangles of P, and let $T = \{a_1, a_2, a_3, b_1, b_2, b_3\}$. Let S be the set containing all vertices in T as well as their neighbors (with respect to G) in P.

By definition of G, every edge in $E(G_2)\backslash E(G)$ has both endpoints in P. Suppose that there exists $i \in \{1, 2, 3\}$ such that $E(G_2)\backslash E(G)$ contains an edge $e = xy$ with $\{x, y\} \subseteq V(P_i)\backslash \{S\}$. Then we can replace P_i by a shorter path using only e and edges of P_i, and obtain a prism in $G \cup \{e\}$. This contradicts the definition of G.

Next, we claim that for each pair P_i, P_j of paths of P, all edges in $E(G_2)\backslash E(G)$ with one endpoint in $V(P_i)\backslash \{S\}$ and one endpoint in $V(P_j)\backslash \{S\}$ share a common endpoint. Suppose not; then there exist edges xy and $x'y'$ in $E(G_2)\backslash E(G)$ with $x, x' \in V(P_i)\backslash \{S\}$ and $y, y' \in V(P_j)\backslash \{S\}$, and with $x \neq x'$ and $y \neq y'$. Without loss of generality, let a_i, x, x', b_i lie in this order on P_i. Let $k = \{1, 2, 3\} \setminus \{i, j\}$. We consider two cases. Suppose first that a_j, y, y', b_j lie in this order on P_j. Let Q_1 be the concatenation of the 1-edge path xy and the subpath of P_j with ends y and y'; let Q_2 be the concatenation of the subpath of P_i with ends x and x' and the 1-edge path $x'y'$; let Q_3 be the concatenation of the subpath of P_i with ends x and a_i, the 1-edge path $a_i a_k$, the path P_k, the 1-edge path $b_k b_j$, and the subpath of P_j with ends b_j and y'. Then $G \cup \left\{xy, x'y'\right\}$ contains a theta with ends x and y' and paths Q_1, Q_2, Q_3. This is a contradiction, because we assumed that no sandwich graph contains a theta. For the other case, suppose that a_j, y', y, b_j lie in this order along P_j. Let Q_1 be the concatenation of the 1-edge path xy and the subpath of P_j with endpoints y and y'; let Q_2 be the concatenation of the subpath of P_i with endpoints x and x' and the 1-edge path $x'y'$; let Q_3 be the concatenation of the subpath of P_i with endpoints x and a_i, and the 1-edge path $a_i a_j$, and the subpath of P_j with endpoints a_j and y'. Then

$G \cup \{xy, x'y'\}$ contains a theta with ends x and y' and paths Q_1, Q_2, Q_3. Again, this is a contradiction, and the claim follows.

Thus, there exists a set U of at most three vertices (one in each of P_1, P_2, P_3) such that each edge in $E(G_2)\backslash E(G)$ has an endpoint either in S or in U. Let S' be the set of all vertices in $S \cup U$ as well as their neighbors in P. Clearly, $|S'| \leq 27$. Let G' be the graph with vertex set $V(G_1)$ and $N_{G'}(x)\backslash S' = N_{G_2}(x)\backslash S'$ for all $x \in V(G')\backslash(S \cup U)$, $N_{G'}(x)\backslash S' = N_{G_1}(x)\backslash S'$ for all $x \in S \cup U$, and for $x, y \in S'$, let xy be an edge if and only if xy is an edge in P. As above, it follows that G' contains P as an induced subgraph. This proves that $\mathcal{C}'$ is almost vertex monotone. □

Theorem 3 *For $n, j \in \mathbb{N}$, the set of holes of length $j \mod n$ is almost edge monotone if and only if it is almost vertex monotone if and only if $j \equiv 2 \mod n$ or $n \leq 2$.*

Proof Let C be a cycle with vertex set $v_1, \ldots, v_l$ such that $v_i v_{i+1}$ is an edge for all i (where we use the convention $v_{l+1} = v_1$ from now on). Vertices v_i and v_{i+1} are called *consecutive*. An edge connecting two non-consecutive vertices is a *chord*. Two distinct chords $v_a v_b$, $v_c v_d$ of C are related in one of the following three ways: either they share an endpoint, or they are *parallel*, i. e. their endpoints are distinct and lie in the order $v_a v_b v_c v_d$ along C (up to cyclic permutation and switching the label of v_a with v_b as well as v_c with v_d), or they *cross*, i. e. their endpoints are distinct and lie in the order $v_a v_c v_b v_d$ along C (up to cyclic permutation and switching the label of v_a with v_b as well as v_c with v_d).

We first give constructions for $n \geq 3$ and $j \not\equiv 2 \mod n$ proving that the class of holes of length $j \mod n$ is not almost vertex monotone. Suppose for a contradiction that there exists a $k \in \mathbb{N}$ such that the property of containing such a hole is k-vertex monotone. Let $N = (2k+2)n + j$ and let G_2 be a graph with vertex set $\{v_1, \ldots, v_N\}$ and the following edges:

– For $i \in \{1, \ldots, N-1\}$, $v_i v_{i+1}$ is and edge, and $v_1 v_N$ is an edge; and
– for $i \in \{1, \ldots, k+1\}$, $v_{in-1} v_{N-in}$ is an edge, and these edges are called *special*.

In other words, G_2 is a long cycle C in which the special edges form parallel chords such that the number of edges of the hole C between the two consecutive endpoints of different special edges is n. This construction is shown in Fig. 1. By inspection, it follows that no hole in any sandwich graph contains three or more special edges; therefore, every hole in a sandwich graph contains at most two special edges. If it contains two special edges, its length is $2 \mod n$; if it contains one special edge, its length is either $j+1 \mod n$ or $1 \mod n$; if it contains no special edge, it is the hole C containing all vertices of G_2 in order, and this is the only hole of length $j \mod n$ in any sandwich graph for (G_1, G_2) unless $j \equiv 1 \mod n$.

Next, consider the following, slightly modified construction. Let $N = (2k+2)n + j$ and let G_2' be a graph with vertex set $\{v_1, \ldots, v_N\}$ and the following edges:

– For $i \in \{1, \ldots, N-1\}$, $v_i v_{i+1}$ is and edge, and $v_1 v_N$ is an edge; and
– for $i \in \{1, \ldots, k+1\}$, $v_{in-1} v_{N-in-2}$ is an edge, and these edges are called *special*.

Let G_1 be the graph with $V(G_1) = V(G_2')$ and $E(G_1) = \emptyset$. Then (G_1, G_2') is a sandwich instance and the sandwich graph obtained by removing all special edges from G_2' contains a hole of length $j \mod n$. This construction is shown in Fig. 2.

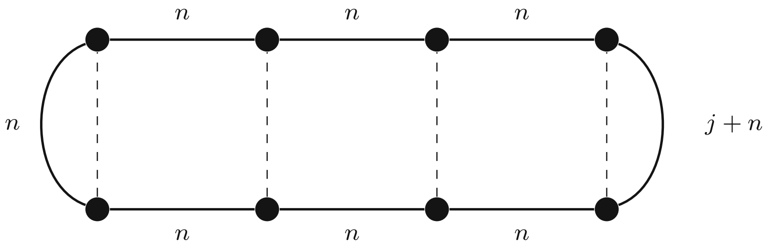


Fig. 1 Construction showing that $j \equiv 1 \mod n$

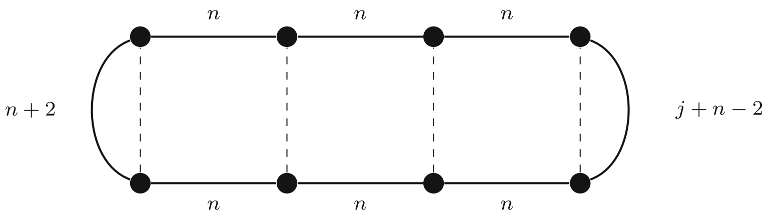


Fig. 2 Construction showing that $j \equiv 3 \mod n$

As before, every hole in a sandwich graph contains at most two special edges. If it contains two special edges, its length is $2 \mod n$; if it contains one special edge, its length is either $j-1 \mod n$ or $3 \mod n$; if it contains no special edge, it is the hole C containing all vertices of G_2 in order, and this is the only hole of length $j \mod n$ in any sandwich graph for (G_1, G_2') unless $j \equiv 3 \mod n$. If $3 \equiv 1 \mod n$, and then $n = 2$, but we assumed that $n \geq 3$.

Therefore, the hole C is the only hole of length $j \mod n$ in any sandwich graph for at least one of (G_1, G_2) and (G_1, G_2'). Since we assumed that the property of containing a hole of length $j \mod n$ was k-vertex monotone, it follows that there exists a set S of k vertices such that there is a sandwich graph G containing a hole of length $j \mod n$, and either all edges with no endpoint in S are as in G_1, or all edges with no endpoint in S are as in G_2. If edges outside S are as in G_1, then there are at most $3k$ edges in G, but C has $N \geq 2kn \geq 6k$ edges, so G does not contain the hole C. If edges outside S are as in G_2, then S does not include either endpoint for at least one of the special edges, and so C is not induced in G. In both cases, we reached a contradiction, and thus the property of containing a hole of length $j \mod n$ is not monotone if $n \geq 3$ and $j \not\equiv 2 \mod n$.

Let $n \leq 2$ and $j \not\equiv 2 \mod n$. Then we must have $j \equiv 1 \mod n$, and thus holes of length $\equiv j \mod n$ are precisely odd holes, for which we proved the result in Lemma 5.

Now, let $j = 2$ and $n \in \mathbb{N}$. Let (G_1, G_2) be a sandwich instance such that some sandwich graph contains a hole of length $2 \mod n$, and let G be the sandwich graph with $|E(G_2) \backslash E(G)|$ minimum subject to G containing a hole C of length $2 \mod n$. It follows that all edges in $E(G_2) \backslash E(G)$ have both endpoints in $V(C)$.

Let $v \in V(C)$. The number of edges in $E(G_2) \backslash E(G)$ incident with v is at most n. (1)

Suppose for a contradiction that $v \in V(C)$ is the endpoint of $n+1$ distinct chords. Let $w_1, \ldots, w_{n+1}$ be the endpoints in $V(C) \setminus \{v\}$ of these chords, and without loss of generality, let $v, w_1, \ldots, w_{n+1}$ lie in this order along C. Let P_i denote the w_1-w_i path in $C \setminus \{v\}$. If there is an $i > 1$ such that the number of edges of P_i is $0 \mod n$, then $v \cup V(P_i)$ induces a hole of length $2 \mod n$ in $G \cup \{vw_1, vw_i\}$. This is a contradiction. Therefore, $(|E(P_i)| \mod n) \in \{1, \ldots, n-1\}$ for all $i > 1$, and by the pigeonhole principle, there exist $1 < i < j$ such that $|E(P_i)| \equiv |E(P_j)| \mod n$. But then $(V(P_j) \setminus V(P_i)) \cup \{w_i, v\}$ induces a hole of length $2 \mod n$ in $G \cup \left\{vw_i, vw_j\right\}$. This is a contradiction, and (1) is proved.

$$\text{Let } E' \subseteq E(G_2) \setminus E(G) \text{ such that either for all distinct } e, e' \in E', e \text{ and } e' \text{ cross, or for all distinct } e, e' \in E', e \text{ and } e' \text{ are parallel. Then } |E'| \leq n. \tag{2}$$

Suppose for a contradiction that there exist distinct vertices $v_1, \ldots, v_{n+1}$ and $w_1, \ldots, w_{n+1}$ such that either $v_1, \ldots, v_{n+1}, w_1, \ldots, w_{n+1}$ lie in this order along C, or $v_1, \ldots, v_{n+1}, w_{n+1}, \ldots, w_1$ lie in this order along C, and $v_i w_i \in E(G_2) \setminus E(G)$ for all $i \in \{1, \ldots, n+1\}$. Let P_i denote the v_1-v_i path in $C \setminus \{w_1, w_i\}$, and let P_i' denote the w_1-w_i path in $C \setminus \{v_1, v_i\}$. If there is an $i > 1$ such that $|E(P_i) + E(P_i')| \equiv 0 \mod n$, then $V(P_i) \cup V(P_i')$ induces a hole of length $2 \mod n$ in $G \cup \{v_1 w_1, v_i w_i\}$, a contradiction. Thus, $(|E(P_i) + E(P_i')| \mod n) \in \{1, \ldots, n-1\}$ for all $i > 1$. By the pigeonhole principle, there exist $1 < i < j$ such that $|E(P_i) + E(P_i')| \equiv |E(P_j) + E(P_j')| \mod n$. But then $(V(P_j) \setminus V(P_i)) \cup (V(P_j') \setminus V(P_i')) \cup \{w_i, v_i\}$ induces a hole of length $2 \mod n$ in $G \cup \left\{v_i w_i, v_j w_j\right\}$. This is a contradiction, and (2) is proved.

By Ramsey's theorem [30], there exists a number $R(n)$ such that if C has at least $R(n)$ chords, then C has at least n chords that either all have a common endpoint, or all pairs of them cross, or all pairs of them are parallel. Thus, $|E(G_2) \setminus E(G)| \leq R(n)$, which proves that the set of holes of length $2 \mod n$ is $R(n)$-edge monotone. □

In particular, the set of even holes is almost vertex monotone. Since even-hole-free graphs can be recognized in polynomial time [13], we obtain the following.

Corollary 2 *The sandwich problems for the following properties can be solved in polynomial time:*

- *containing a pyramid as an induced subgraph;*
- *containing a theta as an induced subgraph;*
- *containing a theta or a prism as an induced subgraph;*
- *containing an even hole.*

In particular, we proved that the property of containing a pyramid is 10-vertex monotone, containing a theta is 14-vertex monotone, and containing a theta or a prism is 27-vertex monotone. These constants are not best possible, and it is not hard to see that these properties are almost edge monotone as well. We leave the proof to the reader.

We presented a number of results that imply polynomial-time algorithms for the not $\mathcal{C}$-free sandwich problem, and thus also for the corresponding partitioned probe problem. The following lemma shows that both the unpartitioned and the partitioned probe problem can be reduced to the recognition problem in this context, even if $\mathcal{C}$ is not almost vertex monotone.

Lemma 6 *The unpartitioned probe and partitioned probe problem are in P for all not $\mathcal{C}$-free problems such that recognition of $\mathcal{C}$-free graphs is in P.*

Proof We may assume that $\mathcal{C} \neq \emptyset$. Let k be the minimum number of vertices of a graph in $\mathcal{C}$. Let G be a graph, possibly with a given partition into probe and non-probe vertices P and N. If $|N| \geq k$ in the partitioned probe problem, or if G contains a stable set of size at least k in the unpartitioned probe problem, then there is a choice of optional edges such that a subset of N induces a graph in $\mathcal{C}$. Otherwise, in the partitioned probe problem, $|N|$ is constant and thus the number of optional edges is constant, so we may check $\mathcal{C}$-freeness for each choice of optional edges. In the unpartitioned probe problem, there are at most $|V(G)|^k$ possible choices for N, and for each of them, we check in polynomial time whether the resulting partitioned probe graph is a not $\mathcal{C}$-free probe graph. □

3 Decompositions

In this section, we will focus on the partitioned and unpartitioned probe problems, and consider the property of having a certain decomposition.

Let G be a graph. A *cutset* in G is a set $X \subseteq V(G)$ such that $G \backslash X$ is not connected. A *cut vertex* is a vertex x such that $\{x\}$ is a cutset. A *clique cutset* in G is a cutset X such that X is a clique in G. A *star cutset* in G is a cutset X with a special vertex v such that v is complete to $X \backslash v$; here, v is called a *center* of the star cutset. A star cutset is *full* if its center has no neighbors outside the cutset. A *homogeneous set* in G is a set $X \subseteq V(G)$ with $|X| \geq 2$ and $|V(G) \backslash X| \geq 1$ such that for all $v \in V(G) \backslash X$, either v is complete to X or v is anticomplete to X. A *homogeneous pair* in G is a partition $(Q_1, Q_2, A, B, S_1, S_2)$ of $V(G)$ such that

- $|Q_1| \geq 2$ or $|Q_2| \geq 2$ and $|V(G) \backslash (Q_1 \cup Q_2)| \geq 2$; and
- A is complete to Q_1 and Q_2; and
- B is anticomplete to Q_1 and Q_2; and
- S_1 is complete to Q_1 and anticomplete to Q_2; and
- S_2 is complete to Q_2 and anticomplete to Q_1.

A *1-join* in G is a partition (A_1, B_1, A_2, B_2) of $V(G)$ such that A_1 is complete to A_2, B_1 is anticomplete to $A_2 \cup B_2$ and B_2 is anticomplete to $A_1 \cup B_1$, and $|A_1 \cup B_1| \geq 2$, $|A_2 \cup B_2| \geq 2$.

Table 1 gives an overview of the hardness of the decomposition problems we will consider. New results are in bold; known results are shown for clique cutset due to [32,35], star cutset due to [12,32] (for completeness, we give an algorithm for the full star cutset sandwich problem in Lemma 9) homogeneous set due to [4], homogeneous pair due to [21], and 1-join due to [15,18].

3.1 Algorithms

We first consider the clique cutset partitioned probe problem. The clique cutset sandwich problem is known to be *NP*-complete [32]. Whitesides [35] gave a

Table 1 Hardness of decomposition problems for recognition, sandwich problem, partitioned probe problem, unpartitioned probe problem, and unpartitioned probe problem in the complement

	Recogn.	Sandwich	Part.	Unpart.	Unp. in G^c
Clique cutset	*P*	*NPc*	***P***	***NPc***	?
Full star cutset	*P*	*P*	*P*	***NPc***	***P***
Homogeneous set	*P*	*P*	*P*	***P***	***P***
Homogeneous pair	*P*	?	***P***	?	?
1-Join	*P*	*NPc*	***P***	***P***	***P***

New results are shown in bold

polynomial-time algorithm for the problem of finding a clique cutset in a graph, which we adapt here.

A graph is *chordal* if it does not contain a hole as an induced subgraph. Every chordal graph either is a complete graph or has a clique cutset [19].

Theorem 4 (Berry et al. [2]) *A graph G is a chordal probe graph with partition (P, N) if and only if N is stable and for every hole C of G, $G|(V(C) \cap P)$ is stable. The chordal partitioned probe problem can be solved in polynomial time.*

Theorem 5 *The clique cutset partitioned probe problem can be solved in polynomial time.*

Proof Let G be a graph and $N \subseteq V(G)$ be a stable set; let $P = V(G)\backslash N$. Suppose G is chordal probe with partition (P, N) and G' is a supergraph of G which is chordal and such that every edge in $E(G')\backslash E(G)$ has both of its endpoints in N. If G' has a clique cutset, then G is a YES instance for the clique cutset partitioned probe problem. If G' is a complete graph, then either G is a complete graph, and thus there is no clique cutset in any probe graph and G is a NO instance, or there exist $x, y \in N$. Let G'' arise from G' by removing the edge xy. Then G'' has the clique cutset $V(G)\backslash \{x, y\}$, and hence G is a YES instance.

Now we may assume that G is not a chordal probe graph with partition (P, N), and thus there exists an induced subgraph of G which is a hole containing two consecutive vertices $x, y \in P$. We find such a hole as follows: for each edge xy with $x, y \in P$, let X be the set of vertices adjacent to both x and y. Then there is a hole in G using xy if and only if there is a path from x to y in $G'' = (G\backslash \{xy\})\backslash X$, which can be checked in polynomial time, and by choosing an induced x-y-path in G'', we can find such a hole C. Let z be the neighbor $\neq x$ of y in C.

We say that $S \subseteq V(G)$ is *inseparable* if for every (P, N)-probe graph H for G and every clique cutset K of H, $S\backslash K$ is included in a connected component of $H\backslash K$. If S is a clique, then S is inseparable. We claim that $S_0 = \{x, y, z\}$ is inseparable. Let H be a (P, N) probe graph for G. Then $H|V(C)$ contains x, y, z, and an induced path Q from x to z not using any neighbors of y (because $y \in P$, $N_H(y) = N_G(y)$). Since $x \in P$, xz is not an edge, it follows that Q has at least two edges, and hence $H|(V(Q) \cup \{y\})$ is a hole C' containing x, y, z. But a hole has no clique cutset, and thus, for every clique cutset K of H, $C'\backslash K$ is connected. This proves our claim.

Now let S_i be an inseparable set which is not a clique in any (P, N)-probe graph H for G. We claim that either $S_i = V(G)$, or there exists an inseparable set S_{i+1} which is a proper superset of S_i and can be found in polynomial time, or some (P, N)-probe graph for G has a clique cutset. This claim implies that starting with S_0, which is not a clique in any (P, N) probe graph since $x \in P$ is non-adjacent to z, it follows that we can grow a maximal sequence $S_0, S_1, \ldots, S_k$ with $k \leq |V(G)|$ and S_i a proper subset of S_{i+1} and S_i inseparable for all i in polynomial time, and if $S_k = V(G)$, then $V(G)$ is inseparable, and so G is a NO instance; if $S_k \neq V(G)$, then G is a YES instance. Thus, our result follows from the claim.

To prove the claim, let S_i be an inseparable set which is not a clique in any (P, N)-probe graph, and let $S_i \neq V(G)$. Let Z be a connected component of $G \backslash S_i$, and let Y be the set of neighbor of Z in S_i. If $Y \cap P$ is a clique complete to $Y \cap N$, then the (P, N)-probe graph H for G in which we add an edge ab if and only if $a, b \in Y \cap N$ has the clique cutset Y separating Z from $H \backslash (Z \cup Y)$, and since S_i is not a clique in H, but Y is, it follows that $V(H) \backslash (Z \cup Y) \supseteq S_i \backslash Y \neq \emptyset$. Thus, we may assume that there exists $a \in Y \cap P$, $b \in Y$ with a non-adjacent to b. Let Q be an induced a-b path in $G|(Z \cup \{a, b\})$, and let c be the neighbor of a in Q. Suppose that $S_i \cup \{c\}$ is not inseparable. Then there exists a (P, N)-probe graph H for G and a clique cutset K in H such that in $H \backslash K$, $S_i \cup \{c\}$ contains vertices from at least two connected components. Since S_i is inseparable, it follows that there exists a connected component T of $H \backslash K$ containing $S_i \backslash K$, and $S_i \cap V(T) \neq \emptyset$ since S_i is not a clique in H. Thus, there exists a second connected component T' of $H \backslash K$ containing c. Since a is adjacent to c, it follows that $a \in K$. Since $a \in P$, it follows that $V(Q) \cap K = \{a\}$, because a has exactly one neighbor in $V(Q) \backslash \{a\}$, and this neighbor is $c \in T'$. Since $G|(V(Q) \backslash \{a\})$ is connected, so is $H|(V(Q) \backslash \{a\})$, and therefore, $V(Q) \backslash \{a\} \subseteq T'$. But now $b \in S_i \subseteq T \cup K$, and also $b \in V(Q) \backslash \{a\} \subseteq T'$. This is a contradiction as $(T \cup K) \cap T' = \emptyset$. Thus, $S_i \cup \{c\}$ is inseparable, and we may choose $S_{i+1} = S_i \cup \{c\}$. This concludes the proof. □

The disconnected sandwich problem can be solved in polynomial time, because it is hereditary; thus the partitioned probe problem and the unpartitioned probe problem can be solved in polynomial time as well.

Lemma 7 *The disconnected unpartitioned probe problem in the complement can be solved in polynomial time.*

Proof A graph G is a YES instance for the disconnected unpartitioned probe problem in the complement if there exists $N \subseteq V(G)$ such that N is a clique in G and a partition (A, B) of $V(G)$ such that the edges with one endpoint in each of A and B have both endpoints in N. In other words, G has a *biclique cutset*, which is defined as a partition (A_1, B_1, A_2, B_2) of $V(G)$ with $A_1 \cup B_1 \neq \emptyset$, $A_2 \cup B_2 \neq \emptyset$ such that B_1 is anticomplete to $A_2 \cup B_2$, B_2 is anticomplete to $A_1 \cup B_1$, and $A_1 \cup A_2$ is a clique.

If G is disconnected, then G has a biclique cutset with $A_1 = A_2 = \emptyset$. Otherwise, every biclique cutset satisfies that $A_1, A_2 \neq \emptyset$.

Let $v, w \in V(G)$. Suppose that there is a biclique cutset $(A_1^*, B_1^*, A_2^*, B_2^*)$ with $v \in A_1^*$ and $w \in A_2^*$. We find this biclique cutset as follows. First, if v is non-adjacent to w, then no such biclique cutset exists. Let A be the set containing v, w and all common neighbors of v and w. From the definition of a biclique cutset it follows that $A = A_1^* \cup A_2^*$. If A is not a clique, then no such biclique cutset exists. Let $C_1, \ldots, C_k$ be the connected components of $G \backslash A$. For $i = 1, \ldots, k$, let D_i be the set of neighbors of C_i in A. If there is a vertex u in $A \backslash (D_1 \cup \cdots \cup D_k)$, then $(\{u\}, \emptyset, A \backslash \{u\}, V(G) \backslash A)$ is a biclique cutset. Therefore, we may assume that $D_1 \cup \cdots \cup D_k = A$. Let H be the hypergraph with vertex set A and edges $D_1, \ldots, D_k$. If H is not connected, then there exists a partition (A_1, A_2) of A with $A_1, A_2 \neq \emptyset$ such that for $i \in \{1, \ldots k\}$, either $D_i \subseteq A_1$ or $D_i \subseteq A_2$. Let B_1 be the union of $V(C_i)$ for i with $D_i \subseteq A_1$ and B_2 the union of $V(C_i)$ for i with $D_i \subseteq A_2$. Then (A_1, B_1, A_2, B_2) is a biclique cutset. If H is connected, then there exists an $i \in \{1, \ldots, k\}$ such that $D_i \cap A_1^*, D_i \cap A_2^* \neq \emptyset$. But then $C_i \subseteq B_1$ and also $C_i \subseteq B_2$, because C_i is connected and has neighbors in A_1^* and A_2^*. This is a contradiction, which proves that if H is connected, then no biclique cutset containing $v \in A_1^*$ and $u \in A_2^*$ exists.

Every step of the procedure described above can be done in polynomial time, and by applying it to all pairs of vertices, we find a biclique cutset if there is one. Therefore, this solves the disconnected unpartitioned probe problem in the complement. □

Lemma 8 (Chvátal [12]) *In a graph G, v is the center of a star cutset if and only if either*

- $G \backslash (\{v\} \cup N(v))$ *is disconnected; or*
- $N(v) = V(G) \backslash \{v\}$ *and* $N(v)$ *contains two non-adjacent vertices; or*
- $N(v)$ *contains a vertex anticomplete to* $V(G) \backslash (\{v\} \cup N(v))$.

Lemma 9 *The full star cutset sandwich problem can be solved in polynomial time.*

Proof Let (G_1, G_2) be a sandwich instance, and suppose that v is the center of a full star cutset in some sandwich graph G for (G_1, G_2); let X be the cutset and let (A, B) be a partition of $G \backslash X$ such that $A, B \neq \emptyset$ and A is anticomplete to B. If $G_1 \backslash (\{v\} \cup N_{G_2}(v))$ is disconnected, then v is the center of a full star cutset in the sandwich graph arising from G_1 by adding all edges incident with v in G_2. If v is complete to $V(G_1) \backslash \{v\}$ in G_2, then v has at least two non-adjacent non-neighbors x and y in G_1 (one in A, one in B). Therefore, v is the center of a full star cutset in $G_2 \backslash \{xv, yv, xy\}$.

Finally, we consider the case that $G_1 \backslash (\{v\} \cup N_{G_2}(v))$ is non-empty and connected, and without loss of generality, $V(G_1) \backslash (\{v\} \cup N_{G_2}(v)) \subseteq A$. Let $x \in B$, then x is anticomplete to $A \cup \{v\}$ in G_1. Thus, v is the center of a full star cutset in the sandwich graph arising from G_2 by removing all edges incident with x in $E(G_2) \backslash E(G_1)$.

Applying this to every vertex $v \in V(G_1)$ yields a polynomial-time algorithm for checking if some sandwich graph has a full star cutset. □

This implies that the full star cutset partitioned probe problem can be solved in polynomial time as well.

Lemma 10 *The star cutset unpartitioned probe problem in the complement can be solved in polynomial time. The same is true for the full star cutset unpartitioned probe problem in the complement.*

Proof Let G be a graph. For each vertex $v \in V(G)$, we check if there is a probe graph in the complement G' for G in which v is the center of a star cutset, i. e. if there is a clique N in G so that G' arises from G by removing a set of edges with both endpoints in N. Let X be the cutset and let (A, B) be a partition of $G' \backslash X$ such that $A, B \neq \emptyset$ and A is anticomplete to B.

Suppose first that $N_G(v) \cup \{v\} = V(G)$. If v has two adjacent neighbors x, y, then $G \backslash \{xy, xv, yv\}$ has a full star cutset with center v. Thus, we may assume that $V(G) \backslash \{v\}$ is a stable set. If $|V(G)| \leq 2$, then no unpartitioned probe graph in the complement for G has a star cutset. If $|V(G)| \geq 3$, let w be a neighbor of v, then $\{v, w\}$ is a full star cutset with center w. This can be done in polynomial time.

The next case we consider is when $G' \backslash (N_G(v) \cup \{v\})$ is connected and non-empty. Without loss of generality, let $V(G) \backslash (\{v\} \cup N_G(v)) \subseteq A$. Then B contains a vertex $x \in N_G(v)$ anticomplete to $V(G) \backslash (\{v\} \cup N_G(v))$ in G' and adjacent to v, i. e. $N_G(x) \backslash (\{v\} \cup N_G(v))$ is a clique. If $N_G(v)$ contains such a vertex x, then let $N = N_G(x) \backslash (\{v\} \cup N_G(v))$ and let G' be the probe graph in the complement for G in which all edges with both endpoints in N are removed. Then G' contains a star cutset with center v in which x is one of the connected components of $G' \backslash (\{v\} \cup N_G(v))$. Now, suppose that X is a full star cutset in G'. Since $B \subseteq N_G(v)$, this implies that $v \in N$, and thus every vertex in B is non-adjacent to every vertex in $V(G) \backslash (\{v\} \cup N_G(v))$, because no such vertex is in a clique also containing v. Let $x \in B$, and let $N = \{x, v\}$. Let $G'' = G \backslash \{xv\}$. Then G'' is a probe graph in the complement for G, and G'' has a full star cutset with center v, because x is an isolated vertex of $G'' \backslash (\{v\} \cup N_{G''}(v))$. This shows that in this case, we can test all combinations of v and x and find a (full) star cutset in polynomial time.

Therefore, we may assume that $G' \backslash (N_G(v) \cup \{v\})$ is disconnected. This implies that $G \backslash (N_G(v) \cup \{v\})$ is a YES instance for the disconnected unpartitioned probe problem in the complement. By Lemma 7, we find a clique N in polynomial time such that if G'' is the graph arising from G after removing edges with both endpoints in N, $G'' \backslash (N_G(v) \cup \{v\})$ is disconnected, and so $N_G(v) \cup \{v\}$ is a full star cutset in G''. This concludes the proof. $\square$

Lemma 10 is of particular interest because we will prove in Theorem 13 that the full star cutset unpartitioned probe problem is *NP*-hard, thus giving an example of a problem for which the unpartitioned probe problem has a different complexity in the graph and in its complement assuming that $P \neq NP$.

In the following, we will use a tool from [22]. Let $k \in \mathbb{N}$, and let M be a symmetric $(k \times k)$-matrix with entries in $\{0, 1, *\}$. Let G be a graph, and let $L : V(G) \to 2^{\{1,\dots,k\}}$ be a function assigning to each vertex a subset of $\{1, \dots, k\}$. An *M-list partition* of G with respect to L is a partition of $V(G)$ into sets $(A_1, \dots, A_k)$ such that

- if $v \in A_i$, then $i \in L(v)$; and
- for all $i \in \{1, \dots, k\}$, if $M_{ii} = 0$, then A_i is a stable set in G, and if $M_{ii} = 1$, then A_i is a clique in G; and
- for all distinct $i, j \in \{1, \dots, k\}$, if $M_{ij} = 0$, then A_i is anticomplete to A_j, and if $M_{ij} = 1$, then A_i is complete to A_j.

This problem is quite general, but here we will only use Lemma 11:

Lemma 11 (Feder et al. [22]) *The list partition problem with lists of size at most two can be solved in polynomial time.*

By slightly adapting the proof of Lemma 11, we can extend its result to the sandwich problem.

Corollary 3 *The M-list partition sandwich problem with respect to L with lists of size at most two can be solved in polynomial time.*

Proof Let (G_1, G_2) be a sandwich instance with $V(G_1) = V(G_1) = V$ and $E(G_1) \subseteq E(G_2)$. The reduction uses a variable v_i for each $v \in V, i \in L(v)$ which is true if $v \in A_i$. If $L(v) = \{i, j\}$, we add the clause $(v_i \vee v_j)$, and if $L(v) = \{i\}$, we add the clause (v_i). For each pair v_i, w_j with $v \neq w$, if $M_{ij} = 0$, and $vw \in E(G_1)$, we add a clause $(\overline{v_i} \vee \overline{w_j})$; if $M_{ij} = 1$, and $vw \notin E(G_2)$, we add a clause $(\overline{v_i} \vee \overline{w_j})$ as well. If there is a valid list partition $(A_1, \ldots, A_k)$, then the assignment in which v_i is true if and only if $v \in A_i$ satisfies all clauses. For the other direction, if we have a satisfying assignment, then for each variable, v_i is true for some $i \in L(v)$; put v in A_i. Suppose that this is not a valid list partition, then there exists $i, j \in \{1, \ldots, k\}$ and $v \in A_i, w \in A_j, v \neq w$, such that either $M_{ij} = 1$ and $vw \notin E(G_2)$, or $M_{ij} = 0$ and $vw \in E(G_1)$. Therefore, the instance has a clause $(\overline{v_i} \vee \overline{w_j})$, but by definition, v_i and w_j are true in our assignment, and hence it is not a satisfying assignment. This reduction uses at most $2|V(G)|$ variables and $|V(G)|^2$ clauses, hence the fact that 2-SATISFIABILITY can be solved in polynomial time [20] implies the result. □

Theorem 6 *The unpartitioned probe homogeneous set problem and the same problem in the complement can be solved in polynomial time.*

Proof First, note that if H is a homogeneous set in G, then H is a homogeneous set in G^c. Therefore, the property $\mathcal{P}$ of having a homogeneous set satisfies $\mathcal{P} = \mathcal{P}^c$, and hence the complexity of the unpartitioned probe problem is the same in the graph and in the complement.

To solve the unpartitioned probe homogeneous set problem, we note that a homogeneous set in G is a partition of $V(G)$ into H, A and B with $|H| \geq 2$ and $|V(G) \backslash H| \geq 1$, A complete to H and B anticomplete to H. Suppose that there exists a partition (P, N) of $V(G)$ and a (P, N)-probe graph G' for G such that G' has a homogeneous set H with A complete to H and B anticomplete to H in G'. We may assume $N \subseteq H \cup A$, because if $E(G') \backslash E(G)$ contains any edge with an endpoint in B, removing it from $E(G')$ preserves that H is a homogeneous set complete to A and anticomplete to B.

If $N \cap A = \emptyset$ or $N \cap H = \emptyset$, then H is complete to A in G, and therefore G has a homogeneous set. A homogeneous set in G can be found in polynomial time [31]. Therefore, we may assume that $N \cap A \neq \emptyset$ and $N \cap H \neq \emptyset$. If $H \subseteq N$, then H is complete to $A \backslash N$ and anticomplete to $B \cup (A \cap N)$ in G, and thus H is a homogeneous set in G, and again, H can be found in polynomial time. Thus, we may assume that $H \backslash N \neq \emptyset$.

To prove the result, we need to show for $v, w, u \in V(G)$ how to find a homogeneous set H complete to A and anticomplete to B in a (P, N)-probe graph for G with $v \in H \backslash N$, $w \in H \cap N$, and $u \in A \cap N$. Let X be the set containing u as well as all vertices of G that are non-adjacent to w, non-adjacent to u, and adjacent to v. It follows that $(N \cap A) \subseteq X \subseteq N$, and hence $G \backslash X$ has a homogeneous set containing v and w (because $H \backslash X$ is complete to $A \backslash X$ and anticomplete to B).

Let $H' \subseteq H$. If there is a vertex $x \in V(G) \backslash (H' \cup X)$ such that x has a neighbor and a non-neighbor in H', we call x a *mixed vertex* for H'; then $x \notin A \backslash X$, $x \notin B$, $x \notin X$, and so $x \in H$, which implies that $\{x\} \cup H' \subseteq H$. If there is a vertex $x \in X \backslash H'$ such that $H' \backslash N(x)$ is not a stable set, we call x a *non-stable vertex* for H'; then $x \notin N \cap A$, and so $x \in N \cap H$, and thus $\{x\} \cup H' \subseteq H$. If there is a vertex $x \in X \backslash H'$ such that x has a neighbor $y \in H'$ with y non-adjacent to u, we call x a *conflict vertex* for H'; since all non-neighbors of u in $H' \subseteq H$ are in N, it follows that $y \in N$, but since $X \subseteq N$, $x \in N$. But then N is not stable, which is a contradiction, and so $H' \nsubseteq H$. If there is a vertex $x \in X \backslash H'$ such that x has a non-neighbor $y \in H'$ with y adjacent to u, we call x a *small vertex* for H'; then $x \notin B$, but since u is adjacent to y, it follows that $y \in P \cap H$, and so $x \notin A$, and thus $x \in H$; therefore, $\{x\} \cup H' \subseteq H$.

This gives rise to the following algorithm. For all $v, w, u \in V(G)$, let $H' = \{v, w\}$. Compute X as above. While there exists a mixed vertex, a non-stable vertex, or a small vertex for H', we add it to H'. If X is not stable, or u was added to H', or there is a conflict vertex, the algorithm terminates with a No, because there is no homogeneous set H in a (P, N)-probe graph for G with $v \in H \backslash N$, $w \in H \cap N$, and $u \in A \cap N$. Clearly, this algorithm takes polynomial time, since it runs for at most $|V(G)|$ steps, each of which takes time polynomial in $|V(G)|$.

Let H'' be the set we obtain if the algorithm does not terminate with a No. Let $N'' = (H'' \backslash N_G(u)) \cup (X \backslash H'')$, and let G'' be the graph arising from G by adding edges between every pair of vertices in N''. Since $u \notin H''$ and u is not a non-stable vertex, it follows that $(H'' \backslash N_G(u))$ is stable, and X is stable. If there is a vertex x in $X \backslash H''$ with a neighbor in $H'' \setminus N_G(u)$, then x is a conflict vertex. Since the algorithm did not terminate with a No, it follows that N'' is a stable set, and so G'' is a probe graph for G. Let $x \in V(G'') \backslash H''$, and suppose that x has a neighbor in H'' and a non-neighbor in H'' with respect to G''. Then x is not a mixed vertex for H'' in G, and so $x \in X \backslash H''$. Let y be a non-neighbor of x in H'' with respect to G'', then $y \in N_G(u)$, but then x is a small vertex for H'', a contradiction. Thus, no such vertex x exists. Since $v, w \in H''$ and $u \notin H''$, it follows that H'' is a homogeneous set in the $(V(G) \backslash N'', N'')$-probe graph G'' for G. We found H'' in polynomial time, which proves the result. □

Theorem 7 *The partitioned probe homogeneous pair problem can be solved in polynomial time.*

Proof Let G be a graph and N a stable set in G; let $P = V(G) \backslash N$. Suppose that there is a partition $(Q_1, Q_2, A, B, S_1, S_2)$ of $V(G)$ which is a homogeneous pair in a (P, N)-probe graph G' for G, and that $S_1, S_2 \subseteq N$ and $N \cap Q_1, N \cap Q_2 \neq \emptyset$. Let $Q = (Q_1 \cup Q_2) \cap N$. We claim that $(Q_1 \backslash Q, Q_2 \cup Q, A, B, S_1, S_2)$ is a homogeneous pair in some (P, N)-probe graph G'' for G. Let G'' arise from G' by removing all edges from S_1 to Q and adding all edges from S_2 to Q. Then S_1 is complete to $Q_1 \backslash Q$ and anticomplete to $Q_2 \cup Q$, S_2 is complete to $Q_2 \cup Q$, and A is complete to $Q_1 \cup Q_2$, B is anticomplete to $Q_1 \cup Q_2$. Since $N \cap Q_1$, $N \cap Q_2 \neq \emptyset$, it follows that $|Q_2 \cup Q| \geq 2$. Moreover, $|A \cup B \cup S_1 \cup S_2| \geq 2$, because these sets remain unchanged. By symmetry, this proves that if G with partition (P, N) is a Yes instance for the partitioned probe homogeneous pair problem, then there exists a partition $(Q_1, Q_2, A, B, S_1, S_2)$ of $V(G)$ which is a homogeneous pair in a (P, N)-probe graph G', and $S_1 \cap P \neq \emptyset$ or $Q_1 \subseteq P$.

We consider two steps. First, if $Q_2 = \emptyset$, or if $S_1 = S_2 = \emptyset$, then we are looking for a homogeneous set Q_1 in some (P, N)-probe graph for G with the additional requirement that $|V(G)\backslash Q_1| \geq 2$. This can be found as follows. For all pairs of vertices $p, q \in V(G)$, we test if there is such a homogeneous set containing p and q. Let $H = \{p, q\}$. While there is a vertex x with a neighbor y and a non-neighbor z in H such that $\{x, z\} \cap P \neq \emptyset$, add x to H. Let H' be the set after this terminates. In the beginning, $H \subseteq Q_1 \cup Q_2$. At every step, we add a vertex x to H if there exist y and z in H such that xy is an edge and xz is a non-edge in every (P, N)-probe graph for G. Since $Q_1 \cup Q_2$ is a homogeneous set containing H, it follows that x is in $Q_1 \cup Q_2$, and thus, $H' \subseteq Q_1 \cup Q_2$. Let G' be the (P, N)-probe graph for G in which we add an edge from $x \in N \cap (V(G)\backslash(Q_1 \cup Q_2))$ to $y \in N \cap (Q_1 \cup Q_2)$ if and only if x has a neighbor (in G) in $(Q_1 \cup Q_2)\backslash N$. Suppose that there is a vertex $x \in V(G')\backslash H'$ such that x has a neighbor in H' and a non-neighbor in H' with respect to G'. Then $x \notin P$, because we would have added x to H'. So $x \in N$, and since x has a neighbor in H', x has a neighbor in $H'\backslash N$. If x had a non-neighbor in $H'\backslash N$, we would have added x to H'. But then, by definition of G', x is complete to $H'\backslash N$ and to $H' \cap N$. Thus, every vertex in $V(G')\backslash H'$ is either complete to anticomplete to H'. Moreover, $|H'| \geq 2$ as H' includes p, q, and $|V(G)\backslash H'| \geq |V(G)\backslash(Q_1 \cup Q_2)| \geq 2$. Therefore, we have found H', a homogeneous pair with $Q_2 = \emptyset$, and with $S_1 = S_2 = \emptyset$, in G', a (P, N)-probe graph for G, in polynomial time.

For the second step, suppose that there is a partition $(Q_1, Q_2, A, B, S_1, S_2)$ of $V(G)$ which is a homogeneous pair in a (P, N)-probe graph for G, and $Q_1, Q_2 \neq \emptyset$ and hence $|Q_1 \cup Q_2| \geq 3$. Suppose further that $S_1 \cup S_2 \neq \emptyset$, and $S_1 \cap P \neq \emptyset$ or $Q_1 \subseteq P$. For $p, q, r, x \in V(G)$, we will show how to test if there is such a partition with $\{p, q, r\} \subseteq Q_1 \cup Q_2$, and one of the following holds:

(a) $x \in S_1 \cap P$; or
(b) $x \in S_1 \cap N$, $Q_1 \subseteq P$; or
(c) $x \in S_2 \cap N$, $Q_1 \subseteq P$.

Every homogeneous pair that was not found in the first step satisfies one of these assumptions (up to symmetry) for some choice of $\{p, q, r, x\}$. Now suppose such a partition $(Q_1, Q_2, A, B, S_1, S_2)$ of $V(G)$ which is a homogeneous pair in a (P, N)-probe graph for G exists with $\{p, q, r, x\}$ as above. Let $Q_1', Q_2' = \emptyset$. For each vertex v in $\{p, q, r\}$, if we are in case (a) or (b), add v to Q_1' is v is adjacent to x, and add v to Q_2' otherwise. If we are in case (c), add v to Q_1' if v is non-adjacent to x and $v \in P$, and add v to Q_2' otherwise. It follows that $Q_1' \subseteq Q_1$ and $Q_2' \subseteq Q_2$.

While there is a vertex $v \in V(G)\backslash(Q_1 \cup Q_2)$ and there exist a, b with $\{a, b\} \subseteq Q_1$ or $\{a, b\} \subseteq Q_2$ such that $\{v, a\} \cap P \neq \emptyset$ and $\{v, b\} \cap P \neq \emptyset$, and $va \in E(G)$, $vb \notin E(G)$, we add v to $Q_1' \cup Q_2'$. In case (a), we add v to Q_1' if $vx \in E(G)$, and we add v to Q_2' otherwise. In case (b), we add v to Q_2' if $v \in N$, and otherwise proceed as for (a). In case (c), we add v to Q_2' if $v \in N$ or $vx \in E(G)$, and we add v to Q_1' otherwise. In each case, it follows that this algorithm preserves the property that $Q_1' \subseteq Q_1$, $Q_2' \subseteq Q_2$. After at most $|V(G)|$ iterations, this algorithm terminates with $Q'_1 \subseteq Q_1, Q'_2 \subseteq Q_2$ in polynomial time. Let G' be the (P, N) probe graph for G arising from G by adding all edges from $z \in V(G)\backslash(Q'_1 \cup Q'_2)$ to $N \cap Q'_1$ if z has a neighbor in $Q'_1\backslash N$, and adding all edges from $z \in V(G)\backslash(Q'_1 \cup Q'_2)$ to $N \cap Q'_2$ if z has a neighbor in $Q'_2\backslash N$. Suppose for a contradiction that there is a vertex $z \in V(G')\backslash Q'_1 \cup Q'_2$ such that either z is neither complete nor anticomplete to Q'_1 in G', or z is neither complete nor anticomplete to Q'_2 in G'; without loss of generality, let this be the case for Q'_1. Then $z \notin P$, for otherwise the algorithm would have added z to Q'_1 or Q'_2. It follows that $z \in N$, and since z has a neighbor in Q'_1 with respect to G', by definition, z has a neighbor a in $Q'_1 \cap P$ with respect to G. But then there exists $b \in Q'_1\backslash N_{G'}(z) \subseteq P$, and a and b would have caused the algorithm to add z to Q'_1 or Q'_2. This implies that $V(G')$ can be partitioned into (S'_1, S'_2, A', B') such that S'_1 is complete to Q'_1 and anticomplete to Q'_2, S'_2 is complete to Q'_2 and anticomplete to Q'_1, A' is complete to $Q'_1 \cup Q'_2$ and B' is anticomplete to $Q'_1 \cup Q'_2$. If $|S'_1 \cup S'_2 \cup A' \cup B'| \geq 2$, then this is a homogeneous pair in a (P, N)-probe graph for G. If not, then $Q'_1 \cup Q'_2 \subseteq Q_1 \cup Q_2$ implies that $(Q_1, Q_2, S_1, S_2, A, B)$ is not a homogeneous pair either, a contradiction showing that no homogeneous pair with x, p, q, r as chosen exists. By checking all combinations of x, p, q, r, and each of cases (a), (b) and (c), we find a homogeneous pair with the specified properties in a (P, N)-probe graph in polynomial time, if there is one. This concludes the proof. □

Theorem 8 *The 1-join partitioned probe problem and the 1-join unpartitioned probe problem can be solved in polynomial time.*

Proof For the partitioned probe problem, we claim that for G and a partition (P, N), if there is a (P, N)-probe graph for G that has a 1-join, then there is a (P, N)-probe graph with a 1-join (A_1, B_1, A_2, B_2) such that either $A_1 \subseteq N$ and $A_2 \subseteq P$, or $A_1 \cap P \neq \emptyset$ and $A_2 \cap P \neq \emptyset$. Suppose not, then there is a (P, N)-probe graph G' for G with a 1-join (A_1, B_1, A_2, B_2), and without loss of generality $A_1 \subseteq N$, $A_2 \cap N \neq \emptyset$. Let G'' be the graph obtained from G' by removing all edges with one endpoint in A_1 and one endpoint in $A_2 \cap N$. This is a (P, N)-probe graph for G, because we have only modified edges with both endpoints in N. But now $(A_1, B_1, A_2 \cap P, (A_2 \cap N) \cup B_2)$ is a 1-join in G'', and it satisfies the first condition. This proves the claim.

Next, note that if G contains a 1-join, we can find it in polynomial time [15]. Thus, we may assume that G does not contain a 1-join, and hence if there is a 1-join (A_1, B_1, A_2, B_2) in a probe graph for G, then $N \cap A_1, N \cap A_2 \neq \emptyset$. In particular, if there is a 1-join in a probe graph with $A_1 \subseteq N$, $A_2 \subseteq P$, then G has a 1-join. This implies that we only need to show how to find a 1-join (A_1, B_1, A_2, B_2) in a probe graph with $A_1 \cap N, A_1 \cap P, A_2 \cap N, A_2 \cap P \neq \emptyset$.

For distinct $u, v \in P$, we show how to find a (P, N)-probe graph with a 1-join (A_1, B_1, A_2, B_2) such that $u \in A_1 \cap P, v \in A_2 \cap P$, if it exists. We consider the four sets A, B, S_1, S_2 where A is the set of common neighbors of u and v in G, B is the set of vertices of G non-adjacent to both u and v, S_1 is the set of vertices of G adjacent to u and non-adjacent to v, and S_2 is the set of vertices of G adjacent to v and non-adjacent to u. Clearly, (A, B, S_1, S_2) is a partition of $V(G)\backslash \{u, v\}$. Moreover, by definition of a 1-join, and since we cannot modify edges adjacent with either u or v in a (P, N)-probe graph, it follows that $A \subseteq A_1 \cup A_2$, $B \subseteq B_1 \cup B_2$, $S_1 \subseteq B_1 \cup A_2$, and $S_2 \subseteq B_2 \cup A_1$.

We can now formulate the 1-join partitioned probe problem as a list partition sandwich problem with $G_1 = G$ and G_2 the graph arising from G by adding edges between every pair of vertices in N. For each $w \in V(G)\setminus\{u, v\}$, if $w \in A$, we set $L(w) = \{A_1, A_2\}$; if $w \in B$, we set $L(w) = \{B_1, B_2\}$; if $w \in S_1$, we set $L(w) = \{B_1, A_2\}$; and if $w \in S_2$, we set $L(w) = \{B_2, A_1\}$. We set $L(u) = \{A_1\}$, $L(v) = \{A_2\}$. Moreover, we require that A_1 is complete to A_2, B_1 is anticomplete to A_2 and B_2, and B_2 is anticomplete to A_1. To satisfy the cardinality constraint, we check for all pairs $x, y \in V(G)\setminus\{u, v\}$ if the list partition sandwich instance has a solution when $L(x)$ is replaced by $L(x) \cap \{A_1, B_1\}$ and $L(y)$ is replaced by $L(y) \cap \{A_2, B_2\}$. If there is a solution for any pair x, y, then the corresponding partition is a 1-join in the (P, N)-probe graph arising from G by adding all edges between $A_1 \cap N$ and $A_2 \cap N$. On the other hand, if there is a 1-join in a (P, N)-probe graph, then there exists $x \in (A_1 \cup B_1)\setminus\{u\}$ and $y \in (A_2 \cup B_2)\setminus\{v\}$, and for this choice of x, y, there is a valid solution of the list partition sandwich instance. By Corollary 3, the list partition sandwich problem with lists of size at most two can be solved in polynomial time. Therefore, we can find a 1-join (A_1, B_1, A_2, B_2) in a (P, N)-probe graph such that $u \in A_1$, $v \in A_2$ in polynomial time, if it exists.

For the unpartitioned probe problem, we consider the same cases, and show how to find a partition (P, N) and a 1-join (A_1, B_1, A_2, B_2) in a probe graph for G. As before, we may assume that G does not contain a 1-join, and we only need to show how to find a 1-join (A_1, B_1, A_2, B_2) in a probe graph with $A_1 \cap N$, $A_1 \cap P$, $A_2 \cap N$, $A_2 \cap P \neq \emptyset$. For $p, q, r, s \in V(G)$, we will give an algorithm for finding a 1-join (A_1, B_1, A_2, B_2) in a probe graph with some partition (P, N) and with $p \in A_1 \cap N$, $q \in A_1 \cap P$, $r \in A_2 \cap N$, $s \in A_2 \cap P$. We may assume that $B_1 \cap N = B_2 \cap N = \emptyset$.

We now show that this can be written as a list partition problem with six parts $A_1 \cap P$, $A_1 \cap N$, $A_2 \cap P$, $A_2 \cap N$, B_1, B_2 such that B_1 is anticomplete to $B_2 \cup A_2$, B_2 is anticomplete to $B_1 \cup A_1$, A_1 is complete to $A_2 \cap P$, A_2 is complete to $A_1 \cap P$, $A_1 \cap N$ is anticomplete to $A_2 \cap N$, and $A_1 \cap N$, $A_2 \cap N$ are stable sets. This is a list partition problem, and if it has a solution with $p \in A_1 \cap N$, $q \in A_1 \cap P$, $r \in A_2 \cap N$, $s \in A_2 \cap P$, then the graph G' arising from G by adding all edges with both endpoints in $N \cap (A_1 \cup A_2)$ is a $(N, V(G)\setminus N)$-probe graph for G in which (A_1, B_1, A_2, B_2) is a 1-join.

By Lemma 11, it suffices to show that in this list partition problem, all lists have size at most two. Then the problem can be solved in polynomial time, and by solving it for every choice of $\{p, q, r, s\}$, we solve the 1-join unpartitioned probe problem in polynomial time. Let $w \in V(G)\setminus\{p, q, r, s\}$. If w is non-adjacent to q and s, then $w \in B_1 \cup B_2$. If w is non-adjacent to q, adjacent to s, and non-adjacent to r, then $w \in B_2 \cup (A_1 \cap N)$. If w is non-adjacent to q, adjacent to s, and adjacent to r, then $w \in B_2 \cup (A_1 \cap P)$. If w is adjacent to q, non-adjacent to s, and non-adjacent to p, then $w \in B_1 \cup (A_2 \cap N)$. If w is adjacent to q, non-adjacent to s, and adjacent to p, then $w \in B_1 \cup (A_2 \cap P)$. Now we may assume that w is adjacent to q and s. If w is non-adjacent to p and r, then $w \in (A_1 \cap N) \cup (A_2 \cap N)$. If w is adjacent to at least one of p and r, then $w \notin N$, and so $w \in (A_1 \cap P) \cup (A_2 \cap P)$. Every vertex in $\{p, q, r, s\}$ has a list of size one, and as we have shown, every other vertex has a list of size two. This shows that the list partition problem can be solved in polynomial time, which implies the result. □

Theorem 9 *The 1-join unpartitioned probe problem in the complement can be solved in polynomial time.*

Proof Let G be a graph, and suppose that there exists a partition (P, N) of $V(G)$ and a (P, N)-probe graph in the complement G' for G such that G' has a 1-join (A_1, B_1, A_2, B_2). As in Theorem 8, note that if G contains a 1-join, we can find it in polynomial time [15]. Thus, we may assume that G does not contain a 1-join, and hence if there is a 1-join (A_1, B_1, A_2, B_2) in a probe graph in the complement for G with partition (P, N), then $(B_1 \cup B_2) \cap N \neq \emptyset$.

Suppose first that there is a 1-join (A_1, B_1, A_2, B_2) in a probe graph in the complement for G with partition (P, N) and with $B_1 \cap N, B_2 \cap N \neq \emptyset$. For $u, v \in V(G)$, we will show how to find such a 1-join with $u \in B_1 \cap N$, $v \in B_2 \cap N$, if it exists. Let N' be the set containing u and v as well as all of the common neighbors of u and v in G. Then $N' \subseteq N$, because vertices in $(B_1 \cup A_1)\setminus N$ are non-adjacent to v, and vertices in $(B_2 \cup A_2)\setminus N$ are non-adjacent to u, but N is a clique, so $N \subseteq N'$. Thus, we have reduced this to the partitioned probe problem, which can be solved in polynomial time by Theorem 8. By repeating this for all $u, v \in V(G)$, we find a 1-join of this kind in polynomial time, if it exists.

Now suppose that $B_1 \cap N = \emptyset$. Then we may assume that $A_2 \cap N = \emptyset$, because B_1 is already anticomplete to A_2 in G, and thus $N \subseteq A_1 \cup B_2$. Since G does not have a 1-join, it follows that $N \not\subseteq A_1$ and $N \not\subseteq B_2$, and consequently, $B_2 \cap N, A_1 \cap N \neq \emptyset$. Moreover, $(A_1, B_1, (B_2 \cap N) \cup A_2, B_2 \cap P)$ is not a 1-join in G, and so $A_1 \cap P \neq \emptyset$. Furthermore, $(A_1 \cap N, B_1 \cup (A_1 \cap P), A_2 \cup (B_2 \cap N), B_2 \cap P)$ is not a 1-join in G, and so $A_2 \cap P \neq \emptyset$. For u, v, x, y we show how to find such a 1-join with $u \in A_1 \cap N$, $v \in B_2 \cap N$, $x \in A_1\setminus N$, $y \in A_2 \subseteq P$. Let N' be the set containing v and all vertices adjacent to u and v, and non-adjacent to x. It follows that $B_2 \cap N \subseteq N' \subseteq N = (A_1 \cap N) \cup (B_2 \cap N)$. We can now reduce the 1-join problem to a list partition problem with lists of size at most two. We partition into the six sets $A_1 \cap N$, $A_1 \cap P$, $B_1 \subseteq P$, $A_2 \subseteq P$, $B_2 \cap N$, $B_2 \cap P$. For each of u, v, x, y, we have a list of size one. For $n \in N'$, we let $L(n) = \{A_1 \cap N, B_2 \cap N\}$. For $n \notin N'$, it follows that $n \notin B_2 \cap N$. If n is non-adjacent to x and y, then $L(n) = \{B_1, B_2 \cap P\}$. If n is adjacent to x and non-adjacent to y, then $L(n) = \{B_1, A_2\}$. If n is non-adjacent to x and n is adjacent to v, then $L(n) = \{A_1 \cap N, B_2 \cap P\}$. If n is adjacent to y and non-adjacent to x, and n is non-adjacent to v, then $L(n) = \{A_1 \cap P, B_2 \cap P\}$. If n is adjacent to x and adjacent to y and adjacent to v, then $L(n) = \{A_1 \cap N, A_2\}$. If n is adjacent to x and adjacent to y and non-adjacent to v, then $L(n) = \{A_1 \cap P, A_2\}$. We require that $A_1 \cap P$ is complete to A_2 and anticomplete to B_2, $A_1 \cap N$ is a clique and complete to A_2 and $B_2 \cap N$, and anticomplete to $B_2 \cap P$, B_1 is anticomplete to A_2 and B_2, and $B_2 \cap N$ is a clique. If there is a list partition with these lists and these properties, then N is a clique, and by removing all edges with both endpoints in N, we obtain a 1-join (A_1, B_1, A_2, B_2). By solving this list partition problem with lists of size two in polynomial time by Lemma 11, and by checking all choices of u, v, x, y, we find a 1-join in a probe graph in the complement in polynomial time, if there is one. This concludes the proof. □

3.2 Hardness Results

Let G be a graph. A set $M \subseteq E(G)$ is a *matching* if no two edges in M share an endpoint. G is *decomposable* if there exists a partition (V_1, V_2) of $V(G)$ with $V_1, V_2 \neq \emptyset$ such that the set of edges of G with one endpoint in V_1 and one endpoint in V_2 is a matching; (V_1, V_2) is called a *decomposition* of G if this holds. The *line graph* $L(G)$ is the graph with vertex set $E(G)$, and in which distinct $e, f \in E(G)$ are connected by an edge in $E(L(G))$ if and only if e and f share an endpoint.

Theorem 10 (Chvátal [11]) *Recognizing decomposable graphs is NP-hard, even when the maximum degree of the input graph is bounded by four.*

Lemma 12 *Let G be a graph. If G is not connected, then G is decomposable. If G has a cut vertex v separating $G \setminus \{v\}$ into A and B with A anticomplete to B, then G is decomposable if and only if at least one of $G|(A \cup \{v\})$, $G|(B \cup \{v\})$ is.*

Proof Let G be a graph that has a cut vertex v separating $G \setminus \{v\}$ into A and B with A anticomplete to B.

Suppose that G is decomposable with decomposition (V_1, V_2) such that $V_1, V_2 \neq \emptyset$, and $v \in V_1$. Since $V_2 \neq \emptyset$, without loss of generality, let $V_2 \cap A \neq \emptyset$. Then $((A \cup \{v\}) \cap V_1, (A \cup \{v\}) \cap V_2)$ is a decomposition of $G|(A \cup \{v\})$.

For the other direction, suppose that $G|(A \cup \{v\})$ has a decomposition (V_1, V_2), and without loss of generality, $v \in V_1$. Then $(V_1 \cup B, V_2)$ is a decomposition of G. □

By Lemma 12, it follows that the decomposable problem is still *NP*-hard in 2-connected graphs. Theorem 10 was used in [3] to prove, by going to the line graph and using Lemma 13, that the problem of finding a stable cutset in a graph is *NP*-hard.

Lemma 13 (Brandstädt et al. [3]) *If $L(G)$ has a stable cutset, then G is decomposable. If G is decomposable and has minimum degree at least two, then $L(G)$ has a stable cutset.*

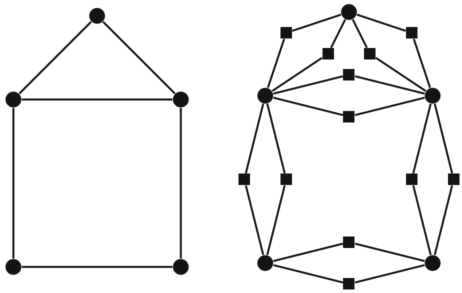
Theorem 11 (Moshi [29]) *The problem of recognizing decomposable graphs is NP-hard, even when the input graph is required to be bipartite.*

Theorem 11 uses the following construction: Let G be a graph. Then $\Diamond(G)$ is defined as the graph containing a vertex for each vertex in G, as well as two vertices e_1 and e_2 for each $e \in E(G)$. For each $v \in V(G)$ and each edge $e \in E(G)$ incident with v, we add two edges ve_1 and ve_2 to $\Diamond(G)$, and no other edges. Clearly, $\Diamond(G)$ is bipartite (the two parts correspond to vertices of G and edges of G, respectively), and Moshi [29] showed that $\Diamond(G)$ is decomposable if and only if G is. An example is shown in Fig. 3.

In a graph G, a *vertex star* at $v \in V(G)$ is a set of edges of G that are all incident with v.

Theorem 12 *The clique cutset unpartitioned probe problem is NP-hard, even when the input is restrict to line graphs of bipartite graphs with clique number at most eight.*

Fig. 3 An example with $G = P_5^c$ (left) and $\Diamond(G)$ (right)



Proof We give a reduction from the problem of recognizing 2-connected decomposable graphs with maximum degree four.

Let G be a 2-connected graph with maximum degree four. Consider the graph $H = L(\Diamond(G))$. We claim that H is a clique cutset probe graph if and only if G is decomposable. Note that since G has maximum degree four, $\Diamond(G)$ has maximum degree eight. Since $\Diamond(G)$ is bipartite, it follows that $L(\Diamond(G))$ has clique number at most eight.

By Lemma 13 and Theorem 11, it suffices to show that H is a clique cutset probe graph if and only if H has a stable cutset. If H has a stable cutset N, then H is a clique cutset probe graph with partition $(V(H)\backslash N, N)$, because the graph H' obtained from H by adding all edges with both endpoints in N has the clique cutset N.

For the converse direction, let H' be a clique cutset probe graph for H with partition (P, N), and let S be a clique cutset in H'. We may assume that $N \subseteq S$, because removing all edges in $E(H')\backslash E(H)$ that do not have both endpoints in S preserves that S is a clique in H' and $H'\backslash S$ is disconnected. If $N = S$, then S is a stable cutset in H, which is what we wanted to show. Therefore, we may assume that $|P \cap S| \geq 1$.

If $|N| \leq 1$, then $H' = H$, and H contains a clique cutset. A clique in the line graph of a bipartite graph corresponds to a vertex star in the bipartite graph, and a clique cutset in the line graph of a bipartite graph corresponds to 1-vertex cutset in the bipartite graph. Since G and $\Diamond(G)$ are 2-connected, it follows that $\Diamond(G)$ has no 1-vertex cutset, and therefore, $|N| \geq 2$.

It is well-known that line graphs of bipartite graphs contain neither a *claw* $(K_{1,3})$ nor a *diamond* $(K_4\backslash e)$ as an induced subgraph. If $|N| \geq 3$, then H contains a claw, and if $|P \cap S| \geq 2$, then H contains a diamond. Therefore, $|N| = 2$, $|P \cap S| = 1$. Let $\{n_1, n_2\} = N$, $\{p\} = P \cap S$. Then, there exists an edge $e = vw \in E(G)$ such that p corresponds to the edge ve_1 or ve_2 in $\Diamond(G)$; by symmetry, we may assume that the former holds. Since n_1 and n_2 are non-adjacent and the edges of $\Diamond(G)$ incident with v form a clique in $L(\Diamond(G))$, it follows that one of n_1, n_2 corresponds to the edge $e_1 w$ in $\Diamond(G)$; by symmetry, we may assume that $n_1 = e_1 w$. It follows that n_2 corresponds either to ve_2 or to ve_1' or ve_2' for some $e' \neq e$.

Since S is a cutset in H, it follows that $\Diamond(G)\backslash\{n_1, n_2, p\}$ has more than one connected component that is not just a single vertex. If $n_2 = ve_1'$ or $n_2 = ve_2'$ for some edge $e' \neq e$, then every vertex of $V(G)$ can be reached from every other vertex of $V(G)$ in $\Diamond(G)\backslash\{n_1, n_2, p\}$, because every edge $e = xy$ in a path in G can be replaced with xe_2 and $e_2 y$ in $\Diamond(G)$. Thus, every vertex of $V(G)$ is in the same connected component in $\diamondsuit(G)\setminus\{n_1, n_2, p\}$, and every other component is therefore a single vertex (because $\diamondsuit(G)\setminus V(G)$ is a stable set). This is a contradiction, since S was a cutset in H.

It follows that $n_2 = ve_2$. The vertex v can be reached from w in $\diamondsuit(G)\setminus\{n_1, n_2, p\}$, because G is 2-connected, and hence there exists a path in G from v to w not using the edge $e = vw$. Every edge of $\diamondsuit(G)$ not incident with e_1 or e_2 is not in the cutset, and since $G\setminus e$ is connected, every vertex of $V(G)$ can be reached from every other vertex of $V(G)$ in $\diamondsuit(G)\setminus\{n_1, n_2, p\}$. As before, this yields a contradiction.

This proves that if H is a clique cutset probe graph, then H has a stable cutset. $\square$

In a graph G, two vertices $x, y \in V(G)$ are *clones* if $N_G(x) = N_G(y)$. A graph G' arises from G by *cloning* $x \in V(G)$ if $V(G') = V(G) \cup \{x'\}$, $G'|V(G) = G$, and $N_{G'}(x') = N_G(x)$.

Theorem 13 *The full star cutset unpartitioned probe problem is NP-hard.*

Proof To prove this, we modify the previous construction as follows. Let G be a 2-connected graph, and let G' arise from G by adding a vertex v complete to $V(G)$. Let $\diamondsuit_v(G')$ arise from $\diamondsuit(G')$ by cloning twice each vertex e_1 for $e \in E(G')$ with e incident to v to obtain two new vertices e_3, e_4 with the same set of neighbors as e_1 (and e_2), and $\{e_1, e_2, e_3, e_4\}$ a stable set. We claim that $H' = L(\diamondsuit_v(G'))$ is a full star cutset probe graph if and only if G is decomposable. $\diamondsuit_v(G')$ consists of $\diamondsuit(G)$, v, and for each vertex w of G, four vertices e_1, e_2, e_3, e_4, each adjacent to precisely v and w. In the line graph, we_1, we_2, we_3, we_4 correspond to a K_4 we will call $t(w)$, and ve_1, ve_2, ve_3, ve_4 correspond to a K_4 we will call $k(w)$. The edges between $t(w)$ and $k(w)$ are precisely edges from ve_i to we_i for $i = 1, 2, 3, 4$. Moreover, for $w, u \in V(G)$, $t(w)$ is anticomplete to $t(u) \cup k(u)$ and $k(w)$ is complete to $k(u)$. For $w \in V(G)$, we denote by $s(w)$ the clique in H' corresponding to edges incident with w in $\diamondsuit(G)$; $s(w) \cup t(w)$ is a clique. Let V^* denote the union of the cliques $s(w)$, i. e. denote the vertices in H' corresponding to edges of $\diamondsuit(G)$. Then $H'|V^* = L(\diamondsuit(G))$. Let K denote the union of the cliques $k(w)$, i. e. the clique in H' corresponding to the vertex star at v in $\diamondsuit_v(G')$. Let T denote the union of the cliques $t(w)$ for $w \in V(G)$. Then $V(H') = V^* \cup K \cup T$, where K is anticomplete to V^*, and $k(w)$ is anticomplete to $V^* \cup (T\setminus s(w))$.

From the proof of Theorem 12, we know that G is decomposable if and only if $H = L(\diamondsuit(G))$ is a clique cutset probe graph, if and only if H has a stable cutset.

If G is decomposable, then $H = L(\diamondsuit(G))$ has a stable cutset S and a partition (A, B) of $V(H)\setminus S$ such that A is anticomplete to B. Then K is anticomplete to S in H', and so we may choose $k \in K$ and $N = \{k\} \cup S$, make k complete to S, and obtain a probe graph H'' for H' which has a star cutset $N(k) \cup S \supset S \cup K$ with center k. This is a cutset, because S is a cutset of H, and for $w \in V(G)$, $N(x) \subseteq s(w) \cup t(w) \cup k(w)$ for $x \in t(w)$, and since $s(w)$ is a clique, $s(w) \cap A = \emptyset$ or $s(w) \cap B = \emptyset$. If $s(w) \cap A = \emptyset$, add $t(w)$ to B, otherwise, to A. By the properties of H' it follows that the resulting sets are still anticomplete to each other.

To prove the other direction, let X be a full star cutset in a probe graph H'' with partition (P, N) for H', and let b be the center of X, and $A = X\setminus\{b\} = N_{H''}(b)$. Let (C, D) be a partition of $H''\setminus X$ such that C is anticomplete to D.

Suppose first that $b \in V^*$, and so $b \in s(w)$ for some $w \in V(G)$. Then b corresponds to an edge $e = ww'$, say $b = we_1$. Then $N_{H'}(b) = (s(w)\setminus\{b\}) \cup t(w) \cup \left\{e_1w'\right\}$. In particular, we may assume that $|K \cap X| \leq 1$ and $K\setminus X \subseteq C$. For $w' \in V(G)\setminus\{w\}$, $|t(w') \cap X| \leq 1$, and so $t(w') \cap C \neq \emptyset$. Consequently, $s(w') \cup t(w') \subseteq X \cup C$ for all $w' \neq w$. But then $D = \emptyset$, a contradiction.

Now suppose that $b \in t(w)$ for some $w \in V(G)$. Then b has exactly one neighbor $k \in K$, and $N_{H'}(b) = (t(w)\setminus\{b\}) \cup s(w) \cup \{k\}$. Therefore, X contains at most two vertices of K, and hence we may assume that $|K\setminus C| \leq 2$. Since X contains at most one vertex from $t(w')$ for all $w' \in V(G)\setminus\{w\}$, it follows that each such $t(w')$ intersects C, and thus $t(w') \cup s(w') \subseteq C \cup X$. But then $D = \emptyset$, a contradiction.

This implies that $b \in K$, and let $w \in V(G)$ such that $b \in k(w)$; let b' be the unique neighbor of b in $t(w)$. Then $N_{H'}(b) = K \cup \left\{b'\right\}$, and thus $X \cap V^*$ is a stable set. We may assume that $X \cap V^*$ is not a cutset of $H'|V^*$, and thus $V^*\setminus\{X\} \subseteq C$. For all $w' \in V(G)$, X contains at most one vertex in $s(w')$, and thus, $s(w') \cup t(w') \subseteq C \cup X$. But then $D = \emptyset$, a contradiction. This concludes the proof. $\square$

Note that this proof does not imply that the star cutset unpartitioned probe problem is *NP*-hard: In the bipartite graph $\Diamond_v(G')$, the maximum degree on one side of the bipartition is two. This implies that in the line graph, every vertex w has a neighborhood consisting of a single vertex x anticomplete to a clique C. By picking $y \in C$, setting $N = \{x, y\}$, and adding the edge xy, we have produced the star cutset $\{x\} \cup C$ with center y separating w from the rest of the graph.

4 Conclusion and Open Questions

We introduced almost monotone properties, and showed that the sandwich problem can be reduced to the recognition problem for almost monotone properties. We proved that the imperfect sandwich problem can be solved in polynomial time.

In the not $\mathcal{C}$-free sandwich problem, we are asking if there exists a sandwich graph in which there exists an induced subgraph isomorphic to a graph in $\mathcal{C}$, whereas in the $\mathcal{C}$-free sandwich problem, we are testing if there exists a sandwich graph G such that for every induced subgraph H of G, H is not in $\mathcal{C}$. The latter problem has an additional alternation, which is an indication that the not $\mathcal{C}$-free sandwich problem might always be "easy", or at least easier than the $\mathcal{C}$-free sandwich problem. Clearly, if the recognition problem for $\mathcal{C}$-free graphs is *NP*-hard (e. g. if $\mathcal{C}$ is the set of prisms), then the not $\mathcal{C}$-free sandwich problem is *NP*-hard. This leads to two open questions:

- Is there a set $\mathcal{C}$ such that recognition of $\mathcal{C}$-free graphs is in P, but the not $\mathcal{C}$-free sandwich problem is *NP*-hard?
- Is there a set $\mathcal{C}$ such that the $\mathcal{C}$-free sandwich problem is in P, but the not $\mathcal{C}$-free sandwich problem is *NP*-hard?

Three kinds of graphs we considered for the not $\mathcal{C}$-free sandwich problems were the Truemper configurations [34], prisms, thetas, and pyramids. In particular, [27] implies that the prism-free and not prism-free sandwich problems are *NP*-hard, because the recognition problem is *NP*-hard. However, the theta-free sandwich problem is *NP*-hard [16], but we proved that the not theta-free sandwich problem is in P. We also proved that the not pyramid-free sandwich problem is in P, but the complexity of the pyramid-free sandwich problem remains open.

We considered the hardness of probe problems for deciding if certain decompositions exist. Our results are summarized in Table 1. In particular, we gave an *NP*-hardness reduction for the clique cutset unpartitioned probe problem, and we generalized it to the full star cutset unpartitioned probe problem. This reduction is mainly based on the fact that in those probe problems, we make changes to a stable set (the set of non-probes) to create a cutset with a certain structure. This allows us to reduce the problem to a variant of the stable cutset problem. It is possible that a similar reduction can be used for the star cutset problem or the *skew cutset* problem, i. e. the problem of finding a cutset X of a graph G such that $G^c|X$ is not connected, which is a generalization of star cutsets. The skew cutset recognition problem is in P [17], and the skew cutset sandwich problem is *NP*-hard [33]. The fast skew partition recognition algorithm in [26] is based on the clique cutset recognition algorithm, and since we gave a polynomial-time algorithm for the partitioned probe clique cutset problem, similar ideas as in [26] might lead to a polynomial-time algorithm for the partitioned probe skew cutset problem.

We also showed that all probe problems are in P for the homogeneous set problem. For the sandwich problem, as well as all probe problems except the partitioned probe problem, it is open if the homogeneous pair problem, a generalization of the homogeneous set problem, can be solved in polynomial time. In general, our algorithms were based on showing that the non-probe vertices could only occur in certain ways in the decomposition, and then assigning a few vertices in key places and checking if these initial choices would lead to a full decomposition using Lemma 11 and Corollary 3. This approach seems useful in general for adapting algorithms for recognition problems to algorithms for the partitioned and unpartitioned probe problem.

Acknowledgements We are thankful to Paul Seymour for many helpful discussions. This material is based upon work supported in part by the U. S. Army Research Laboratory and the U. S. Army Research Office under Grant No. W911NF-16-1-0404.

References

1. Berge, C.: Färbung von Graphen, deren sämtliche bzw. deren ungerade Kreise starr sind. Wiss. Z. Martin-Luther-Univ. Halle-Wittenberg Math.-Natur. Reihe **10**(114), 114 (1961)
2. Berry, A., Golumbic, M.C., Lipshteyn, M.: Recognizing chordal probe graphs and cycle-bicolorable graphs. SIAM J. Discrete Math. **21**(3), 573–591 (2007)
3. Brandstädt, A., Dragan, F.F., Szymczak, T.: On stable cutsets in graphs. Discrete Appl. Math. **105**(1), 39–50 (2000)
4. Cerioli, M.R., Everett, H., de Figueiredo, C.M.H., Klein, S.: The homogeneous set sandwich problem. Inf. Process. Lett. **67**(1), 31–35 (1998)
5. Chudnovsky, M.: Berge trigraphs and their applications. PhD thesis, Princeton University (2003)
6. Chudnovsky, M., Cornuéjols, G., Liu, X., Seymour, P., Vušković, K.: Recognizing Berge graphs. Combinatorica **25**(2), 143–186 (2005)
7. Chudnovsky, M.: Berge trigraphs. J. Graph Theory **53**(1), 1–55 (2006)
8. Chudnovsky, M., Robertson, N., Seymour, P., Thomas, R.: The strong perfect graph theorem. Ann. Math. **164**(1), 51–229 (2006)
9. Chudnovsky, M., Kapadia, R.: Detecting a theta or a prism. SIAM J. Discrete Math. **22**(3), 1164–1186 (2008)

10. Chudnovsky, M., Seymour, P.: The three-in-a-tree problem. Combinatorica **30**(4), 387–417 (2010)
11. Chvátal, V.: Recognizing decomposable graphs. J. Graph Theory **8**(1), 51–53 (1984)
12. Chvátal, V.: Star-cutsets and perfect graphs. J. Comb. Theory Ser. B **39**(3), 189–199 (1985)
13. Conforti, M., Cornuéjols, G., Kapoor, A., Vušković, K.: Even-hole-free graphs part II: recognition algorithm. J. Graph Theory **40**(4), 238–266 (2002)
14. Couto. F, Faria, L., Gravier, S., Klein, S.: On the forbidden induced subgraph probe and sandwich problems. Discrete Appl. Math. **234**, 56–66 (2016)
15. Cunningham, W.H.: Decomposition of directed graphs. SIAM J. Algebr. Discrete Methods **3**(2), 214–228 (1982)
16. Dantas, S., de Figueiredo, C.M.H., da Silva, M.V.G., Teixeira, R.B.: On the forbidden induced subgraph sandwich problem. Discrete Appl. Math. **159**(16), 1717–1725 (2011)
17. de Figueiredo, C.M.H., Klein, S., Kohayakawa, Y., Reed, B.A.: Finding skew partitions efficiently. J. Algorithms **2**(37), 505–521 (2000)
18. de Figueiredo, C.M.H., Klein, S., Vušković, K.: The graph sandwich problem for 1-join composition is NP-complete. Discrete Appl. Math. **121**(1), 73–82 (2002)
19. Dirac, G.A.: On rigid circuit graphs. In: Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg, vol. 25, no. 1, pp. 71–76. Springer, Berlin (1961)
20. Edwards, K.: The complexity of colouring problems on dense graphs. Theor. Comput. Sci. **43**, 337–343 (1986)
21. Everett, H., Klein, S., Reed, B.: An algorithm for finding homogeneous pairs. Discrete Appl. Math. **72**(3), 209–218 (1997)
22. Feder, T., Hell, P., Klein, S., Motwani, R.: List partitions. SIAM J. Discrete Math. **16**(3), 449–478 (2003)
23. Golumbic, M.C., Shamir, R.: Complexity and algorithms for reasoning about time: a graph-theoretic approach. J. ACM: JACM **40**(5), 1108–1133 (1993)
24. Golumbic, M.C., Kaplan, H.: Graph sandwich problems. J. Algorithms **19**(3), 449–473 (1995)
25. Golumbic, M.C., Lipshteyn, M.: Chordal probe graphs. Discrete Appl. Math. **143**(1), 221–237 (2004)
26. Kennedy, W.S., Reed, B.: Fast skew partition recognition. In: Ito, H., Kano, M., Katoh, N., Uno, Y. (eds.) Computational Geometry and Graph Theory, pp. 101–107. Springer, Berlin (2008)
27. Maffray, F., Trotignon, N.: Algorithms for perfectly contractile graphs. SIAM J. Discrete Math. **19**(3), 553–574 (2005)
28. McMorris, F.R., Wang, C., Zhang, P.: On probe interval graphs. Discrete Appl. Math. **88**(1), 315–324 (1998)
29. Moshi, A.M.: Matching cutsets in graphs. J. Graph Theory **13**(5), 527–536 (1989)
30. Ramsey, F. P.: On a problem of formal logic. In: Proceedings of the London Mathematical Society, 2nd Series, vol. 30, pp. 264–344 (1930)
31. Spinrad, J.: P_4-trees and substitution decomposition. Discrete Appl. Math. **39**(3), 263–291 (1992)
32. Teixeira, R.B., de Figueiredo, C.M.H.: The sandwich problem for cutsets: clique cutset, k-star cutset. Discrete Appl. Math. **154**(13), 1791–1798 (2006)
33. Teixeira, R.B., Dantas, S.: Skew partition sandwich problem is NP-complete. Electron. Notes Discrete Math. **35**, 9–14 (2009)
34. Truemper, K.: Alpha-balanced graphs and matrices and $GF(3)$-representability of matroids. J. Comb. Theory Ser. B **32**(2), 112–139 (1982)
35. Whitesides, S.H.: An algorithm for finding clique cut-sets. Inf. Process. Lett. **12**(1), 31–32 (1981)
36. Zhang, P., Schon, E.A., Fischer, S.G., Cayanis, E., Weiss, J., Kistler, S., Bourne, P.E.: An algorithm based on graph theory for the assembly of contigs in physical mapping of DNA. Comput. Appl. Biosci.: CABIOS **10**(3), 309–317 (1994)