Probe Distance-Hereditary Graphs

Maw-Shang Chang\textsuperscript{1} \quad Ling-Ju Hung\textsuperscript{1} \quad Peter Rossmanith\textsuperscript{2}

\textsuperscript{1}Department of Computer Science and Information Engineering
National Chiang Cheng University, Min-Hsiung, Chia-Yi 62102, Taiwan
Email: \{mschang, hunglc\}@cs.ccu.edu.tw
\textsuperscript{2}Department of Computer Science
RWTH Aachen University, Aachen 52056, Germany
Email: rossmani@informatik.rwth-aachen.de

Abstract

A graph $G = (V, E)$ is called a probe graph of graph class $\mathcal{G}$ if $V$ can be partitioned into two sets $\mathcal{P}$ (probes) and $\mathcal{N}$ (nonprobes), where $\mathcal{N}$ is an independent set, such that $G$ can be embedded into a graph of $\mathcal{G}$ by adding edges between certain nonprobes. A graph is distance hereditary if the distance between any two vertices remains the same in every connected induced subgraph. Distance-hereditary graphs have been studied by many researchers. In this paper, we give an $O(nm)$-time algorithm for recognizing probe graphs of distance-hereditary graphs.

1 Introduction

A probe graph $P$ is a two-tuple $(G, L)$ where $G$ is a graph and $L$ is a function from $V_G$ to the set $\{\mathcal{P}, \mathcal{N}, \mathcal{U}\}$ of labels. We use $P_P$, $P_N$, and $P_U$ for the first and second tuples of $P$, respectively, and use $P_P$ and $P_N$ for the sets of vertices and edges of $P_G$, respectively. We also use $P_P$, $P_N$, and $P_U$ for the sets of vertices $v \in V_P$ with $\mathcal{P}(v) = \mathcal{P}$, $\mathcal{P}(v) = \mathcal{N}$, and $\mathcal{P}(v) = \mathcal{U}$, respectively. A probe graph $P$ is fully (resp. partially) partitioned if $P_P = \emptyset$ (resp. $P_P \neq \emptyset$). A probe graph $P$ is unpartitioned if $P_P = P_N = \emptyset$. We call probe graph $P'$ a subgraph of a probe graph $P$ if $P_G'$ is a subgraph of $P_G$ and $\mathcal{P}(v) = \mathcal{P}(v)$ for $v \in P_V'$. Let $X$ be a subset of $P_V$. A subgraph of $P$ induced by $X$ is the subgraph $P'$ of $P$ with $P_G' = P_G[X]$, i.e., $P_E'$ is the subgraph of $P_E$ induced by $X$. For $v \in P_V$, use $P - v$ to denote the probe subgraph of $P$ induced by $P - v$. We also use $P - X$ for the subgraph of probe graph $P$ induced by $P - X$ for a subset $X$ of $P_V$. We call a vertex $v \in P_V$ a probe, a nonprobe, and a prime if $\mathcal{P}(v) = \mathcal{P}$, $\mathcal{P}(v) = \mathcal{N}$, and $\mathcal{P}(v) = \mathcal{U}$, respectively.

A probe graph $P$ is feasible if $P_N$ is an independent set of $P_G$. We say a probe graph $P'$ is an embedding of probe graph $P$ if $P_V' = P_V$, $P_E' \subseteq P_E$, $P_N \subseteq P_N'$, $P_P \subseteq P_P'$, $P'$ is fully partitioned, i.e., $P_U' = \emptyset$, $P_M'$ is an independent set of $P_G$, and for $(u, v) \in P_E - P_E'$ we have $P'_P(u) = P'_P(v) = \emptyset$. Let $\mathcal{G}$ be a class of graphs. We call probe graph $P$ a probe graph if there exists an embedding $P'$ of $P$ such that $P_G' \in \mathcal{G}$. If probe graph $P$ is not feasible, then it does not have any embedding by definition and hence it is not a probe graph for any graph class $\mathcal{G}$. The recognition of fully partitioned (resp. unpartitioned) probe graph $P$ has an embedding in $\mathcal{G}$.


The recognition of fully partitioned probe graph $\mathcal{G}$ graphs is a special case of the graph sandwich problem (Golumbic et al. 1995). Given $G^1 = (V, E^1)$ and $G^2 = (V, E^2)$ where $E^1 \subseteq E^2$, the graph sandwich problem asks whether there exists a graph $G = (V, E)$, $E^1 \subseteq E \subseteq E^2$, where $G$ is in a specific graph class $\mathcal{G}$. For example, the interval sandwich problem asks “Is there an interval graph $G = (V, E)$ where $E^1 \subseteq E \subseteq E^2$?”

Probes distance-hereditary graphs were introduced by Chandler et al. (2006). They gave an $O(n^3)$-time algorithm to recognize fully partitioned probe distance-hereditary graphs. Instead of studying the recognition of fully partitioned (or unpartitioned) probe distance-hereditary graphs directly, we study the recognition of partially partitioned probe distance-hereditary graphs. The recognition of partially partitioned probe distance-hereditary graphs is equivalent to the recognition of fully partitioned distance-hereditary graphs if $P_U = \emptyset$ and is equivalent to the recognition of unpartitioned probe distance-hereditary graphs if $P_P = P_N = \emptyset$. In this paper, we give an $O(nm)$-time algorithm to recognize partially partitioned probe distance-hereditary graphs.

2 Preliminaries

For a vertex $v$ of $G$, the open neighborhood of $v$, denoted by $N_G(v)$, consists of all vertices adjacent to $v$...
in $G$. We use $N_G[v]$ for $N_G(v) + v$, called the closed neighborhood of $v$. For a subset $X$ of $V$, we use $N_G(X) = \cup_{v \in X} N_G(v) - X$ to denote the neighborhood of $X$ in $G$. A subset $X$ of $V$ is called a module in $G$ if for every $x \in X$ $N_G(x) = X - N_G(x)$. Two vertices $u \neq v$ are false twins in $G$ if $N_G(u) = N_G(v)$ and are true twins if $N_G[u] = N_G[v]$. We say they are twins if $N_G(u) = v = N_G(v) - u$. A vertex $v$ in $G$ is called a pendant vertex if the degree of $v$ is one. A vertex $v$ in $G$ is called a universal vertex if the degree of $v$ is $|V| - 1$. A vertex $v$ in $G$ is simplicial if $N_G(v)$ induces a complete subgraph of $G$. In a graph $G = (V, E)$, two disjoint subsets $S$ and $T$ of $V$ are fully adjacent if every vertex of $S$ is adjacent to all vertices in $T$. Two sets $A$ and $B$ are incomparable if $A \cap B \neq \emptyset$, $A - B \neq \emptyset$, and $B - A \neq \emptyset$. For two vertices $u, v \in V$, we use $d_G(u, v)$ to denote the distance of $u$ and $v$ in a graph $G = (V, E)$.

We say a graph $G$ is a distance-hereditary graph (DHG for short) if the distance between any two vertices remains the same in every connected induced subgraph of $G$. It is a classical result that distance-hereditary graphs can be captured by forbidden induced subgraphs (Bandelt & Mulder 1986). For the house, hole, domino, and gem, we refer to Fig. 1. A hole is a $k$-cycle where $k \geq 5$.

**Theorem 1.** (Bandelt & Mulder 1986) Let $G$ be a graph. The following conditions are equivalent:

1. $G$ is distance hereditary.
2. $G$ contains no house, hole, domino, or gem as an induced subgraph.
3. Every connected induced subgraph of $G$ with at least two vertices has a pendant vertex or a twin.
4. For every pair of vertices $x$ and $y$ with $d(x, y) = 2$, there is no induced $x, y$-path of length greater than two.

Cographs are a subclass of distance-hereditary graphs. The following property of cographs is used in the paper.

**Theorem 2.** (Brandstädt et al. 1999) The following conditions are equivalent:

1. $G$ is a cograph;
2. Every induced subgraph of $G$ with at least two vertices has at least one pair of twins;
3. $G$ is $P_4$-free.

![Figure 1: A house, a hole, a domino, and a gem.](image)

In (D’Atri & Moscarini 1988) the notion of a hanging of $G$ by a vertex $v$ was introduced.

**Definition 1.** (D’Atri & Moscarini 1988) The hanging $\Phi$ of $G = (V, E)$ by $v$ is an $(\ell + 1)$-tuple $(L_0, L_1, \ldots, L_\ell)$ where $\ell = \max_{u \in V} d_G(u, v)$, $L_0 = \{v\}$, and $L_i = \{u \in V | d_G(u, v) = i\}$ for $1 \leq i \leq \ell$.

**Definition 2.** Let $\Phi = (L_0, L_1, \ldots, L_\ell)$ be a hanging of $G$. For $x \in L_i$, $0 < i \leq \ell$, use $N_\Phi(x)$ for $N_G(x) \cap L_{i-1}$. Denote the subgraph of $G$ induced by $\cup_{0 \leq j \leq i} L_j$, by $G_\Phi$, for $0 \leq i \leq \ell$. By definition, $G = G_0$. Let $x$ and $y$ be vertices in $L_\ell$ with $1 \leq i \leq \ell$. We say that (i) $x$ properly contains $y$, denoted by $x \gg y$, if $N_\Phi(x)$ properly contains $N_\Phi(y)$; (ii) $x$ and $y$ are equivalent, denoted by $x \equiv y$, if $N_\Phi(x) = N_\Phi(y)$; and (iii) $x$ is minimal (resp. maximal) if there does not exist any other vertex $z \in L_i$ such that $x \gg z$ (resp. $z \gg x$).

**Remark 1.** Let $C$ be a component of $G_i$, where $0 < i \leq \ell$. By definition of hanging, $N_G(C) \subseteq L_{i-1}$.

**Theorem 3.** (D’Atri & Moscarini 1988) A connected graph $G$ is distance hereditary if and only if for every hanging $\Phi = (L_0, L_1, \ldots, L_\ell)$ of $G$ and every pair of vertices $x, y \in L_i$ $(1 \leq i \leq \ell)$ that are in the same component of $G_i$, we have $N_\Phi(x) = N_\Phi(y)$. In other words, for a component $C$ of $G_i$, $N_G(C)$ and $C \cap L_i$ are fully adjacent.

**Theorem 4.** (Hammer & Maffray 1990) Suppose $\Phi = (L_0, L_1, \ldots, L_\ell)$ is a hanging of a connected distance-hereditary graph $G$. For any two vertices $x, y \in L_i$ with $1 \leq i \leq \ell$, $N_\Phi(x)$ and $N_\Phi(y)$ are disjoint or $N_\Phi^{-}(x) \subseteq N_\Phi^{-}(y)$ or $N_\Phi^{+}(y) \subseteq N_\Phi^{+}(x)$.

The following corollary can be seen from the above two theorems.

**Corollary 1.** Suppose $\Phi = (L_0, L_1, \ldots, L_\ell)$ is a hanging of a connected distance-hereditary graph $G$. For any two components $C_1$ and $C_2$ of $G_i$ with $1 \leq i \leq \ell$, $N_G(C_1)$ and $N_G(C_2)$ are disjoint or $N_G(C_1) \subseteq N_G(C_2)$ or $N_G(C_2) \subseteq N_G(C_1)$.

**Theorem 5.** (Hammer & Maffray 1990) Suppose $\Phi = (L_0, L_1, \ldots, L_\ell)$ is a hanging of a connected distance-hereditary graph $G$. For each $1 \leq i \leq \ell$, there exists a minimal vertex $v$. In addition, if $v$ is minimal then $N_G(x) - N_\Phi^{-}(v) = N_G(y) - N_\Phi^{+}(v)$ for every pair of vertices $x$ and $y$ in $N_\Phi(v)$.

By Theorem 3 and Theorem 5, we get the following corollary.
Corollary 2. Suppose $\Phi = (L_0, L_1, \ldots, L_t)$ is a hanging of a connected distance-hereditary graph $G$. For each $1 \leq i \leq t$, $G_i$ has a minimal component $C$, i.e., $N_D(C)$ does not properly contain $N_D(C')$ for any component $C'$ of $G_i$. In addition, if $C$ is a minimal component of $G$, then $N_D(C)$ is a module of $G$.

Corollary 3. Suppose $G$ is a biconnected distance-hereditary graph and $\Phi = (L_0, L_1, \ldots, L_t)$ is a hanging of $G$. Let $C$ be a component of $G$, where $1 < i \leq t$. Then $N_D(C)$ contains a pair of twins.

**Proof.** For every component $C$ of $G_i$, there exists a minimal component $C^*$ such that $N_D(C^*) \subseteq N_D(C)$. For every component $C$ of $G_i$, $N_D(C)$ induces a cograph. Otherwise by Theorem 2 there exists a $P_4$ in $G[N_D(C)]$. The $P_4$ with any $x \in C$ induces a gem, a contradiction. Because $G$ is biconnected, $|N_D(C^*)| > 1$. By Corollary 2, $N_D(C^*)$ is a module of $G$. Since $N_D(C^*)$ induces a cograph in $G$, there exists two vertices $N_D(C^*)$ that are twins in the subgraph induced by $N_D(C^*)$. They are twins in $G$.

In the rest of this section, we give some observations on a probe distance-hereditary graph and its distance-hereditary embedding.

**Proposition 1.** Suppose $P$ is a probe distance-hereditary graph and $P^*$ is a distance-hereditary embedding of $P$. Then the following statements are true:

1. Any two probes in $P^*$ that are false twins in $P^*$ are false twins in $P$.
2. Any two probes in $P^*$ that are true twins in $P^*$ are true twins in $P$.
3. Any two nonprobes in $P^*$ that are false twins in $P^*$ are false twins in $P$.
4. Any two nonprobes in $P^*$ that are true twins in $P^*$ are false twins in $P$.

**Lemma 1.** If $P$ is a probe distance-hereditary graph, then the universal vertex of any induced gem of $P_G$ is a probe in any distance-hereditary embedding of $P$.

**Proof.** Suppose the universal vertex of any induced gem is a nonprobe. Then all the other vertices in the gem are probes. Hence the same five vertices will induce a gem in any embedding of $P$, a contradiction.

The next lemma shows that also any induced house can be assigned a designated vertex.

**Lemma 2.** If $P$ is a probe distance-hereditary graph, then the simplicial vertex of any induced house of $P_G$ is a nonprobe in any distance-hereditary embedding of $P$.

**Proof.** An induced house in $P_G$ can have only two nonprobes in any distance-hereditary embedding of $P$. If they are two vertices of the square, then adding an edge creates a gem, which is a contradiction.

**Theorem 6.** If $P$ is a probe distance-hereditary graph and the subgraph of $P_G$ induced by a set $D$ of six vertices is a domino, then in any distance-hereditary embedding of $P$, $D$ has exactly two nonprobes which are at distance three in the subgraph of $P_G$ induced by $D$.

**Proof.** Any maximal independent set in a domino has either two or three vertices. Assume there are three nonprobes in a distance-hereditary embedding of $P$. Then the three nonprobes have pairwise distance two in $P_G$. If only one edge is added, this creates a house. If two or three edges are added in the embedding it creates a house or a gem. Hence a domino has two nonprobes in any distance-hereditary embedding of $P$. If they are at distance two, a house is created in the embedding. Hence there are exactly two nonprobes in any distance-hereditary embedding of $P$ and they are at distance three in $P_G$.

**Corollary 4.** If $P$ is probe distance-hereditary and the subgraph of $G$ induced by a set $D$ of six vertices is a domino, then the two vertices that have degree three in the subgraph of $P_G$ induced by $D$ are probes.

**Definition 3.** Two disjoint vertex sets $X$ and $Y$ are called probe adjacent if $X$ can be partitioned into two nonempty sets $X_1$ and $X_2$ and $Y$ can be partitioned into two nonempty sets $Y_1$ and $Y_2$ such that every vertex in $X_1$ (resp. $Y_1$) is adjacent to all vertices of $Y$ (resp. $X$) and every vertex in $X_2$ (resp. $Y_2$) is adjacent to all vertices of $Y_1$ (resp. $X_1$) but not adjacent to any vertex of $Y_2$ (resp. $X_2$).

**Lemma 3.** Let $P$ be a probe graph and $P^*$ be a distance-hereditary embedding of $P$. Suppose $X$ and $Y$ are two disjoint vertex sets of $P_G$ of size greater than one and $X$ and $Y$ are fully adjacent in $P_G$. If both $X$ and $Y$ have vertices with labels both $\mathbb{P}$ and $\mathbb{N}$ in $P^*$, then $X$ and $Y$ are probe adjacent in $G$. Besides, a vertex $x \in X$ (resp. $Y$) is a probe in $P^*$ if and only if $x$ is adjacent to all vertices of $Y$ (resp. $X$).

**Proof.** By definition.

**Theorem 7.** Suppose $P$ is a probe graph and $u$ is a universal vertex of $P_G$. Then $P$ is a probe distance-hereditary graph if and only if one of the following conditions is satisfied:

(i) $P_L(u) = \mathbb{P}$ and $P - u$ is a probe cograph.
(ii) $P_L(u) = \mathbb{N}$ and $P - u$ is a cograph.
(iii) $P_L(u) = \mathbb{U}$ and $P - u$ is a probe cograph.

**Proof.** If $P$ has a distance-hereditary embedding $P^*$, then $P^* - u$ is a cograph, i.e., $P - u$ is a probe cograph. Otherwise there exists an induced $P_4$ in $P - u$ and the $P_4$ with $u$ induces a gem in $P^*$, a contradiction. If $u$ is a nonprobe, then all vertices in $P - u$ are probes. Hence $P - u$ must be a cograph.

Suppose $P - u$ is a probe cograph. Let $P^*$ be a cograph embedding of $P - u$. We create a new graph $P^*$ from $P^*$ by letting $u$ be adjacent to all vertices of $P^*$. It cannot introduce any induced $P_4$ in $P^*$ including $u$ since $u$ is a universal vertex. Hence $P^*$ is a cograph embedding of $P$ and a distance-hereditary embedding of $P$. Suppose $P - u$ is a cograph. There is no induced $P_4$ including $u$ since $u$ is a universal vertex. Thus $P$ is a cograph and also a distance-hereditary graph if $u$ is a universal vertex and $P - u$ is a cograph.

**Corollary 5.** Suppose $P$ is a probe graph and $u$ is a universal vertex of $P_G$. Then $P$ is a probe distance-hereditary graph if and only if $P$ is a probe cograph.
3 The algorithm

In this section, we give an $O(nm)$-time algorithm to recognize probe distance-hereditary graphs. This algorithm is a recursive one. We denote the input probe graph by $P$. The algorithm first checks whether $P$ is feasible. If $P$ is not feasible, then it is not a probe distance-hereditary graph. Let $P_{L}(u) = \mathbb{P}$ for all vertices of $u \in P_{L}$ that are adjacent to $v \in P_{L}$ with $P_{L}(v) = \mathbb{N}$. This can be done in linear time. In the following assume $P$ is feasible, i.e., all neighbors of a nonprobe must be probes. The algorithm checks to which of the following classes the input probe graph $P$ does belong and takes action accordingly:

P 1. $P_{L}$ has twins. If it has one, reduce the size of $P$ according to Lemma 4 and 5 and solve the problem recursively. The two lemmas and the reduction steps will be described in Section 3.1.

P 2. $|P_{L}| \leq c$ for some constant $c$. Solve the problem by brute force in $O(1)$ time.

P 3. $P$ is fully partitioned. Use the $O(n^{2})$-time algorithm in (Chandler et al. 2006).

P 4. $P_{L}$ is biconnected and without twins. Call Algorithm B, to be given in Section 3.2, to solve the problem.

P 5. $P_{L}$ is not biconnected and without twins. Call Algorithm R, to be given in Section 3.3 to solve the problem recursively.

It is easy to see the correctness of the algorithm if the algorithm for each class of input is correct. We analyze the time complexity of the algorithm in Section 3.4.

3.1 Twins

In this subsection we first prove two lemmas and show how to use them to solve the problem recursively.

Lemma 4. Suppose $P$ is a probe graph and $u$ and $v$ are true twins in $P_{L}$ satisfying one of the following conditions.

1. $P_{L}(u) = P_{L}(v) = \mathbb{P} \text{ or } \mathbb{U}$.
2. $P_{L}(u) = \mathbb{P}$, $P_{L}(v) = \mathbb{N}$ or $\mathbb{U}$.

Let $P^{*} = P - v$ if $P_{L}(u) = \mathbb{P}$. If $P_{L}(u) = P_{L}(v) = \mathbb{U}$, let $P^{*}$ be the probe graph obtained from $P - v$ by changing the label of $u$ from $\mathbb{U}$ to $\mathbb{P}$, i.e., $P^{*}_{L}(u) = P_{L}(v) = \mathbb{P}$, and $P^{*}_{L}(x) = P_{L}(x)$ for $x \in P_{L} - u$. Then $P^{*}$ is a probe distance-hereditary graph if and only if $P^{*}$ is a probe distance-hereditary graph.

Proof. If $P$ has an embedding $P^{*}$, then $P^{*} = v$ is an embedding of $P - v$. Thus $P^{*} = v$ is an embedding of $P - u$. Next we show that $P$ has a distance-hereditary embedding if $P - v$ has one. Assume that $P^{*}$ is a distance-hereditary embedding of $P$. Suppose that $P_{L}(u) = \mathbb{P}$. We obtain $P^{*}$ from $P^{*}$ by attaching $v$ as a true twin of $u$ and relabel $v$ as a probe in $P^{*}$ if $P_{L}(v) = \mathbb{U}$. Since $u$ is a probe, we have $N_{P^{*}}(u) = N_{P^{*}}(v)$ and hence $N_{P^{*}}(v) = N_{P^{*}}(v)$. By Theorem 1, $P^{*}$ is a distance-hereditary embedding of $P$.

Suppose that $P_{L}(u) = P_{L}(v) = \mathbb{U}$. Since $u$ and $v$ are adjacent in $P$, one of $u$ and $v$ is not a nonprobe in any embedding of $P$. Without loss of generality assume that $u$ is a probe. Since $P_{L}(u) = \mathbb{P}$ and $P_{L}(u) = \mathbb{N}$, we obtain $P^{*}$ from $P^{*}$ by attaching $v$ as a true twin of $u$ and relabel $v$ as a probe in $P^{*}$. By Theorem 1, $P^{*}$ is a distance-hereditary embedding.

Lemma 5. Suppose $P$ is a probe graph and $u$ and $v$ are false twins in $P_{L}$ satisfying one of the following conditions.

1. $P_{L}(u) = P_{L}(v) = \mathbb{P}, \mathbb{N}$, or $\mathbb{U}$.
2. $P_{L}(u) = \mathbb{P}$, $P_{L}(v) = \mathbb{N}$ or $\mathbb{U}$.
3. $P_{L}(u) = \mathbb{N}$, $P_{L}(v) = \mathbb{U}$.

Then $P$ is a probe distance-hereditary graph if and only if $P - v$ is a probe distance-hereditary graph.

Proof. If $P$ has a distance-hereditary embedding $P^{*}$, then $P^{*} = v$ is an embedding of $P - v$. Next we show that $P$ has a distance-hereditary embedding if $P - v$ has one. Suppose $P - v$ has an embedding $P^{*}$. We then obtain $P^{*}$ from $P^{*}$ by attaching $v$ as a false twin of $u$. Let $P_{L}(v) = P_{L}(u)$ if $P_{L}(v) = \mathbb{U}$. If $P_{L}(u) = \mathbb{P}$, we see $N_{P^{*}}(u) = N_{P^{*}}(u)$ and $N_{P^{*}}(v) = N_{P^{*}}(v)$. Suppose $P_{L}(u) = \mathbb{P}$ and $P_{L}(u) = \mathbb{N}$ or $\mathbb{U}$. Assume $N_{P^{*}}(u) = N_{P^{*}}(u) + X$, all vertices in $X$ are nonprobes. We obtain $P^{*}$ from $P^{*}$ by attaching $v$ as a false twin of $u$ and letting $P_{L}(v) = N$ if $P_{L}(v) = \mathbb{U}$. Hence $N_{P^{*}}(v) = N_{P^{*}}(u) = N_{P^{*}}(u) + X$. By Theorem 1, we see $P^{*}$ is a distance-hereditary embedding of $P$.

The proofs of the above two lemmas explicitly point out how to reduce the size of input probe graph $P$ and imply the problem can be solved recursively. In (Lanlignel & Thierry 2000), an $O(n^{2})$-time algorithm was developed for removing all twins in a given graph. The following lemma summarizes the result of this subsection.

Lemma 6. Given a graph, removing vertices that have a twin until it is not possible can be done in $O(n^{2})$ time.

3.2 Kernel probe graphs and Algorithm B

In the subsection we deal with the case that input graph $P$ is of class $P$. 4. This is the most crucial part of the algorithm. We will show that whether such a probe graph $P$ is a probe distance-hereditary graph can be recognized in $O(n^{2})$ time. There are two stages in Algorithm B. At the first stage of Algorithm B, we check whether $P$ is a probe cograph. A probe cograph can be recognized in linear time (Le & de Ridder 2007). If $P$ is not a probe cograph, we do the second stage of Algorithm B. First arbitrarily pick an edge $(x, y)$ of $P_{E}$. In any distance-hereditary embedding of $P$, either $x$ is a probe or $y$ is a probe. Hence we reduce the problem to the case that there is a vertex $p \in P_{V}$ with $P_{L}(p) = \mathbb{P}$. We call a probe graph $P$ satisfying the following four conditions a kernel probe graph: (i) $P_{E}$ is biconnected, (ii) $P_{C}$ has no twins, (iii) there is a vertex $p \in P_{V}$ with $P_{L}(p) = \mathbb{P}$, and (iv) $P$ is not a probe cograph. Given a kernel probe graph $P$ and a probe $p$, our goal is to determine whether $P$ is a probe distance-hereditary graph. We say that a kernel probe graph $P$ is well-labeled if there is a vertex $p$ such that $P_{L}(p) = \mathbb{P}$ and...
contains probes and nonprobes in $G$. This contradicts the assumption that

$$P \text{ is not a probe cograph,}$$

by Corollary 5 there is no universal vertex in $P$. For clarity of the notation, use $G$ for $P$. The second stage of Algorithm B checks to which class of probe graphs the input kernel probe graph $P$ does belong and takes action accordingly:

**C.1.** $P$ is well-labeled.

**C.2.** $P$ is not well-labeled.

To handle the case that $P$ is of class C.2, i.e., not well-labeled, the algorithm again checks to which class of probe graphs the input kernel probe graph $P$ does belong and takes action accordingly:

**D.1.** there is a component $C$ in $G_2$ with $|C \cap L_2| \geq 2$.

If $\ell > 2$, there must be a component $C$ in $G_2$ with $|C \cap L_2| \geq 2$ since $G$ is biconnected.

**D.2.** $\ell = 2, |C| = 1$ for every component $C$ in $G_2$ and there is a vertex $y \in L_2$ with $P_1(y) = \overline{P}$.

**D.3.** $\ell = 2, |C| = 1$ for every component $C$ in $G_2$ and every vertex in $L_2$ is not a probe.

In the following assume that $P$ is a kernel probe graph and $P^*$ is a minimal distance-hereditary embedding of $P$. For simplicity, use $G$ and $G^*$ for $P_2$ and $P^*_2$, respectively. Let $p$ be a probe of $P^*$, $\Phi = (L_0, L_1, \ldots, L_\ell)$ and $\Psi = (Z_0, Z_1, \ldots, Z_\ell)$ be the hangings of $G$ and $G^*$ by vertex $p$, respectively. The above notation will be used in lemmas and theorems in the rest of this subsection. Now we give some observations on both the hangings of $G$ and $G^*$ by a probe $p$.

**Lemma 7.** Suppose $C$ is a component of $G^*_i$ with $1 < i \leq \ell$. Then $N_{G^*_i}(C)$ contains probes and nonprobes in $P^*$. In addition, if $|C \cap Z_i| > 1$ then $C \cap Z_i$ also contains probes and nonprobes in $P^*$.

**Proof.** First we prove that $N_{G^*_i}(C)$ contains probes and nonprobes in $P^*$. By Corollary 3, $N_{G^*_i}(C)$ contains a pair of twins, $u$ and $v$. If the labels of $u$ and $v$ in $P^*$ are the same, then $u$ and $v$ are twins of $G$. This contradicts the assumption that $G$ has no twins. Hence one of $u$ and $v$ is a probe and the other is a nonprobe in $G^*$. Next we show that $C \cap Z_i$ also contains probes and nonprobes in $G^*$ if $|C \cap Z_i| > 1$. There are two cases:

1. $i = \ell$ or $i < \ell$ and $C \cap Z_{i+1} = \emptyset$. Notice that $C \cap Z_i = C$ in this case. Since a vertex in $N_{G^*_i}(C)$ is adjacent to all vertices in $C$, $C$ induces a co-graph in $G^*$. There is a pair of twins in $G^*(C)$. Since $C$ and $N_{G^*_i}(C)$ are fully adjacent in $G^*$ they are also twins of $G^*$. $\Box$

2. $i < \ell$ and $C \cap Z_{i+1} \neq \emptyset$. Let $C'$ be a component of $G^*_{i+1}$ with $C' \subset C$. Then $N_{G^*_i}(C') \subseteq (C \cap Z_i)$. By the first statement of this lemma, $N_{G^*_i}(C')$ contains probes and nonprobes in $P^*$. Hence $C \cap Z_i$ also contains probes and nonprobes in $P^*$. \hfill $\Box$

**Corollary 6.** For $x \in Z_i$ where $1 < i \leq \ell$, $N_{G^*_i}(x)$ contains probes and nonprobes in $P^*$.

**Proof.** Since $G^*$ is biconnected and distance hereditary, $|N_{G^*_i}(x)| > 1$. Obviously, $x \in C$ for some component $C$ of $G^*_i$. By Lemma 7, $N_{G^*_i}(C)$ contains probes and nonprobes in $P^*$. By Theorem 3, $x$ is adjacent to all vertices in $N_{G^*_i}(C)$ in $G^*$. Hence $N_{G^*_i}(x)$ contains probes and nonprobes in $P^*$.

**Lemma 8.** $\ell = \ell_1 > 1$ and $L_i = Z_i$ for $0 \leq i \leq \ell_1$.

**Proof.** Since $P$ is not a probe cograph, there is no universal vertex in $G^*$. Because $p$ is not a universal vertex of $G^*$, $\ell > 1$. Since $G^*$ is obtained from $G$ by adding edges, $\ell \geq h$. Clearly, $L_0 = Z_0 = \{p\}$. Because $p$ is a probe, $L_1 = Z_1$. By induction hypothesis, assume that $L_i = Z_i$ for $0 \leq i < j$. By the assumption, $d_G(y, p) = d_{G^*}(y, p)$ for every $y \in L_j$. We complete the proof by showing that every vertex $x \in Z_i$ is also in $L_i$. Let $x$ be a vertex in $Z_i$. By definition, $d_G(x, p) = d_{G^*}(x, p)$ since $G^*$ is obtained from $G$ by adding edges between nonprobes. By Corollary 6, $x$ is adjacent to a probe $y$ in $Z_{i-1} = L_{i-1}$. Hence $d_G(x, p) = d_{G^*}(x, p) + 1 = i$ and $x \in L_i$. \hfill $\Box$

**Theorem 8.** For $1 < i \leq \ell$ and $x \in L_i, x$ is a probe in $P^*$ if and only if in $G$ $x$ is adjacent to some vertices in $Z_{i-1}$ that are nonprobes in $P^*$.

**Proof.** By Lemma 8, $x \in Z_i$ since $x \in L_i$. By Corollary 6, in $G^*$ every vertex in $Z_i$ is adjacent to vertices in $Z_{i-1}$ that are nonprobes in $P^*$. Since $x$ is a probe, $N_{G^*}(x) = N_{G^*}(L_1)$. By Lemma 8, $L_{i-1} = Z_{i-1}$. Hence in $G$ $x$ is adjacent to vertices in $L_{i-1}$ that are nonprobes in $P^*$. On the other hand, $x$ must be a probe in $P^*$ if in $G$ $x$ is adjacent to some vertices that are nonprobes in $P^*$. \hfill $\Box$

**Algorithm W.** Now we are ready to show the algorithm for the case that $P$ is of class C.1, i.e., a well-labeled kernel probe graph. We refer to the algorithm for handling this case as Algorithm W. We will see that Algorithm W serves as a major subroutine to be used later. The algorithm is as follows. By definition, the labels of vertices in $N_{G^*_i}(p)$ are either $P$ or $N$. Compute $P^*$ from $P$ as follows. Let $P_G = P_G$ and let $P_G^* (y) = P_L(y)$ for all $y \in N_{G_p}(p)$. For every $i$ from $i = 2$ to $i = \ell$ and every $y \in L_i$ with $P_L(y) = \overline{P}$, let $P_G^* (y) = P$ if in $G$ $y$ is adjacent to some vertex $z \in L_{i-1}$ with $P_L^* (z) = N$; and let $P_G^* (y) = N$ otherwise. By Theorem 8, we see that $P$ is a probe distance-hereditary graph if and only if $P^*$ is a probe distance-hereditary graph. Apparently $P^*$ is fully partitioned. Use the $O(n^2)$-time algorithm in (Chandler et al. 2006) to determine whether $P^*$ is a probe distance-hereditary graph. It is not hard to see that Algorithm W runs in $O(n^2)$ time.

In the following we give observations to be used for handling probe graphs of class C.2.

**Lemma 9.** Suppose $P^*$ is a minimal distance-hereditary embedding of $P$. Then the following statements hold:

1. A component of $G_i$ is a component of $G^*_i$ for $0 \leq i \leq \ell$.

2. For any component $C$ of $G_i$ with $1 \leq i \leq \ell$ and $|C \cap L_i| > 1$, $N_{G^*_i}(C) = N_{G^*_i}(C)$.

3. For any component $C$ of $G_i$ with $1 \leq i \leq \ell$ and $|C \cap L_i| > 1$, $N_{G^*_i}(C) = N_{G^*_i}(C)$.  

59
Proof. First we prove Statement (1). By Lemma 8, $h = \ell$ and $L_1 \subseteq Z_i$ for $0 \leq i \leq h = \ell$. In addition, $G^*$ is obtained from $G$ by adding edges. Hence a component of $G^*_i$, $0 \leq i \leq h$, is a component of $G_i$ or the union of some components of $G_i^*$. Since both $G$ and $G^*$ are biconnected, all $G_0, G_1, G^*_i$, and $G^*_i$ have only one component. Hence the lemma holds for $i = 0$ and $i = 1$. For $1 < i \leq \ell$, we prove the statement by contradiction showing that if some component $C$ of $G^*_i$ is not a component of $G_i$ then $P^*$ is not a minimal distance-hereditary embedding of $P$. Suppose $C$ is a component of $G^*_i$ that properly contains a component $D$ of $G_i$. Let $P'$ be an embedding of $P$ obtained from $P^*$ by removing edges connecting a vertex in $C - D$ and another vertex in $D$. Use $G'$ for $P'$. Clearly $N_{G^*}(C) = N_{G^*}(D) \cap Z_i = N_{G^*}(C - D) \cap Z_i = N_{G^*}(C - D)$ and $P'$ is a distance-hereditary graph, contradiction again. Assume that $P'$ is not a distance-hereditary embedding of $P$, i.e., $G'$ is not a distance-hereditary graph. There is an induced forbidden subgraph in $G'$. Let $F$ be the set of vertices that induces a hole or a domino or a gem or a house in $G'$. Because the induced forbidden subgraph is formed by removing edges connecting a vertex in $F$ and another vertex in $F$ such that $(F - D, \emptyset)$ and $P'$ is a distance-hereditary graph, contradiction again.

Next we prove Statement (2). By Statement (1), $C$ is also a component of $G^*_i$. By Lemma 8, $Z_i = L_i$ for $0 \leq i \leq \ell$. Hence $C \cap Z_i \subseteq C \cap L_i$. Since $G^*$ is obtained from $G$ by adding edges, $C \cap L_i \subseteq C \cap Z_i$. Therefore $C \cap Z_i = C \cap L_i$.

Finally, we prove Statement (3). Clearly the statement is true if $i = 1$. In the following assume $1 < i \leq \ell$. By Statement (1) of this lemma, $C$ is also a component of $G^*_i$. By Statement (2) of this lemma, $Z_i \subseteq C \cap L_i$. Since $C \cap Z_i > 1$, by Lemma 7 both $N_{G^*}(C)$ and $C \cap Z_i$ contains probes and nonprobes. Let $x \in C \cap Z_i$ be a probe in $G^*$. Since $G^*$ is distance hereditary, $N_{G^*}(C) = N_{G^*}(x)$ by Theorem 3. Because $Z_i = L_i$, for $0 \leq i \leq \ell$ (see Lemma 8) and $x$ is a probe in $P^*$, $N_{G^*}(x) = N_{G^*}(x)$. Since $G^*$ is obtained from $G$ by adding edges, we have $N_G(C) \subseteq N_{G^*}(C)$. Thus $N_{G^*}(C) = N_{G^*}(x) = N_{G^*}(x) \subseteq N_{G^*}(C) \subseteq N_{G^*}(C)$. This proves the theorem.

**Theorem 9.** Suppose $P^*$ is a minimal distance-hereditary embedding of $P$ and $C$ is a component of $G$, with $|C \cap L_1| > 1$ and $1 < i \leq \ell$. A vertex $x \in C \cap L_i$ (resp. $N_{G^*}(C)$) is a probe in $P^*$ if and only if $x$ is adjacent to all vertices in $N_{G^*}(C)$ (resp. $C$).

**Proof.** By Statement (2) of Lemma 9, $C \cap Z_i = C \cap L_i$. Hence $|C \cap Z_i| > 1$. By Statement (3) of Lemma 9, $N_{G^*}(C) = N_{G^*}(C)$. Since $G^*$ is biconnected, $|N_{G^*}(C)| > 1$. Since $G^*$ is distance hereditary, $N_{G^*}(C)$ and $C \cap Z_i$ are fully adjacent by Theorem 3. By Lemma 3, the theorem holds.

Next we show how to use the above lemmas and theorems to handle the case that $P$ is of class C2.

**Algorithm for D 1.** In this case there is a component $C$ in $G_2$ with $|C \cap L_1| \geq 2$. By Lemma 9 and Theorem 9, a vertex $x \in C \cap L_2$ (resp. $N_{G^*}(C)$) is a probe in $P^*$ if and only if $x$ is adjacent to all vertices in $N_{G^*}(C)$ (resp. $C$). Compute $P'$ from $P$ as follows. Let $P'_G = P_G$ and $P'_L(y) = P_L(y)$ for every $y \in P_Y - (C \cup N_{G^*}(C))$. For every $y \in (C \cup N_{G^*}(C))$, let $P'_L(y) = P_L(y)$ if $P_L(y) \neq \emptyset$. For every $y \in N_{G^*}(C)$ with $P_L(y) = \emptyset$, let $P'_L(y) = P$ if $y$ is adjacent to all vertices $z \in C \cap L_2$ and let $P'_L(y) = \emptyset$ otherwise. If we let $P'_L(y) = P_L(y)$ for all primes $y \in C$, we see that $P$ is a probe distance-hereditary graph if and only if $P'$ is a probe distance-hereditary graph by Theorem 9. But we will go further. Clearly all vertices in $N_{G^*}(C)$ are not primes now. From $i = 2$ to $i = \ell$, for every $y \in C \cap L_i$, with $P_L(y) = \emptyset$, let $P'_L(y) = P$ if $G$ is adjacent to some nonprobes in $L_i$ and let $P'_L(y) = \emptyset$ otherwise. By Theorem 9, we see that $P$ is a probe distance-hereditary graph if and only if $P'$ is a probe distance-hereditary graph after we relabel primes of $P$ in $C$. In $P'$, there must be a probe $p' \in C \cap L_2$. Besides $P'_L(y) \neq \emptyset$ for every $y \in N_{G^*}(p')$. Thus $P'$ is a well-labeled kernel probe graph. We then call Algorithm W to determine whether $P'$ is a probe distance-hereditary graph. It takes linear time to find a component $C$ of $G_2$ with $|C \cap L_2| > 1$ and obtain $P'$ in linear time. Thus the algorithm for D 1 runs in $O(n^2)$ time.

**Algorithm for D 2.** In this case $\ell = 2$ and there is a component $C$ of $G_2$ such that $C = \{y\}$ and $P_L(y) = P$. Theorem 9
If \( N_G(q) = L_1 \), then \( q \) is a false twin of \( p \), a contradiction. Thus \( L_1 - N_G(q) \neq \emptyset \). Let \( (L'_0, L'_1, \ldots, L'_k) \) be the hanging of \( G \) by \( q \). Then \( p \) and all vertices in \( L_1 - N_G(q) \) are in \( L'_0 \) and are in the same component of \( G - N_G(q) \). Hence \( P \) is also of class \( \text{D}1 \) and the algorithm is finished by calling the algorithm for \( \text{D}1 \). Thus the algorithm for \( \text{D}2 \) runs in \( O(n^2) \) time.

**Algorithm for \( \text{D}3 \).** In this case \( \ell = 2, |C| = 1 \) for every component of \( G_{2b} \), and every vertex in \( L_2 \) is not a probe. Let \( q \) be a vertex in \( L_2 \) and be of minimum degree among vertices in \( L_2 \). By definition, \( P_G(q) = U \) or \( P_L(q) = \mathbb{N} \). Let \( P \) be the probe graph \( (P_G, P_L) \) where \( P_G(q) = P \) and \( P_L(x) = P_L(x) \) for \( x \in P_V - q \). Let \( P \) be the probe graph \( (P_G, P_L) \) where \( P_G(q) = \mathbb{N} \), \( P_L(q) = P \) for \( q \in N_G(q) \), and \( P_L(x) = P_L(x) \) for \( x \in P_V - N_G(q) \). If \( P_G(q) = U \), then \( P \) is a probe distance-hereditary graph if and only if one of \( P \) and \( P \) is a probe distance-hereditary graph. It is easy to see that we can use the algorithm for \( \text{D}2 \) to test whether \( P \) is a probe distance-hereditary graph. In the following we focus on checking whether \( P \) is a probe distance-hereditary graph. Notice that \( P_G = G \). In the following we use \( G \) to refer to \( P_G \). Since \( G \) has no twins, \( N^*_G(q) \neq N^*_G(q') \) for any \( q' \in L_2 \) and \( q' \neq q \). We distinguish the following two classes of the input graph:

**N 1.** There exists a vertex \( q' \in L_2, q' \neq q \), such that \( N^*_G(q') \) and \( N^*_G(q) \) are incomparable, i.e., \( N^*_G(q') \cap N^*_G(q) \neq \emptyset, N^*_G(q') - N^*_G(q) \neq \emptyset, \) and \( N^*_G(q) - N^*_G(q') \neq \emptyset \).

**N 2.** For all \( q' \in L_2, q' \neq q \), either \( N^*_G(q) \subseteq N^*_G(q') \), or \( N^*_G(q') \subseteq N^*_G(q) \) are disjoint.

To which of the above two classes \( P \) does belong can be determined in \( O(n + m) \) time. We refer to the algorithms for input probe graphs of classes \( \text{N}1 \) and \( \text{N}2 \) as Algorithm \( \text{N}1 \) and Algorithm \( \text{N}2 \), respectively.

**Algorithm N 1.** Let \( y_1, y_2 \in N^*_G(q) \) where \( y_2 \in N^*_G(q') \) and \( y_1 \notin N^*_G(q') \). Notice that \( y_1 \) and \( y_2 \) must be probes in any distance-hereditary embedding of \( G \). Suppose \( y_1 \) and \( y_2 \) are not adjacent in \( G \). Consider the hanging \( (L_0, L_1, \ldots, L_k) \) of \( G \) by \( y_1 \). By definition, \( d_C(y_1, y_2) \geq 2 \). Clearly \( d_C(y_1, y_2) = 2 \) if \( d_C(y_1, y_2) = 2 \), then both \( y_2 \) and \( q' \) are in \( L_2 \) and in the same components of the graph obtained by removing \( N_G(y_1) \). If \( d_C(y_1, y_2) > 2 \), then \( d_C(y_1, z) = 2 \). Hence \( z \) and \( q' \) are in \( L_2 \) and in the same components of the graph obtained by removing \( N_G(y_1) \). Clearly \( P \) is of class \( \text{D}1 \) and we can finish this case in \( O(n^2) \) time. In addition, assume \( z \) and \( y_1 \) are adjacent in the following. Consider the following subcases:

(a) \( z \) is adjacent to \( y_2 \) in \( G \). \( \{y_1, y_2, z, q, q'\} \) induces another gem, where \( y_2 \) is the universal vertex and \( qy_1, zq' \) is the \( P_4 \) in the embedding. There is a probe \( \{q, q'\} \) in the embedding. Besides \( z \) must be a probe. Then \( \{y_2, p, y_1, q, q'\} \) induces another gem, where \( y_2 \) is the universal vertex and \( py_1y_2q'q' \) is the \( P_5 \) in the embedding and we have no way to destroy it by adding edges between nonprobes. Thus \( q' \) must be a probe in any distance-hereditary embedding of \( P \). Let \( P' \) be the probe graph \( (P_G', P_L') \) where \( P_G'(q') = P \) and \( P_L'(x) = P_L'(x) \) for \( x \in P' - q' \). It is easy to see that \( P' \) is of class \( \text{D}2 \). Hence this case can be done in \( O(n^2) \) time by calling the algorithm for \( \text{D}2 \).

(b) \( z \) is not adjacent to \( y_2 \) in \( G \). \( \{q, y_1, y_2, q', z\} \) induces a house as shown in Fig. 2(b) where \( y_2q'y_1z = y_1 \) is the \( C_4 \) of the house. Suppose \( q' \) is a nonprobe in the embedding. Similar to the above case, edge \( (q, q') \) must be added to destroy this house in the embedding and hence creating a gem induced by \( \{y_2, p, y_1, q, q'\} \), where \( y_2 \) is the universal vertex and \( py_1q'q' \) is the \( P_4 \) in the embedding and we have no way to destroy it by adding edges between nonprobes. Thus \( q' \) must be a probe in any distance-hereditary embedding of \( P \). By arguments similar to those given in the above case, we can finish this case in \( O(n^2) \) time.

**Algorithm N 2.** Let \( Y \) denote the set of vertices in \( L_1 - N^*_G(q) \) that are adjacent to some but not all vertices of \( N^*_G(q) \). Let \( X \) denote the set of vertices in \( L_1 - N^*_G(q) \) that are adjacent to all vertices of \( N^*_G(q) \). Notice that all vertices in \( N^*_G(q) \) are probes in any distance-hereditary embedding. Hence \( N^*_G(q) \) induces a cograph in \( G \). There is a pair of twins in the subgraph of \( G \) induced by \( N^*_G(q) \). If \( Y = \emptyset \), then they are also twins of \( G \), a contradiction. Assume \( Y \neq \emptyset \) in the following. Let \( y \) be any vertex in \( Y \). By definition there are \( y_1, y_2 \in N^*_G(q) \) such that \( y \) is adjacent to \( y_2 \) but not adjacent to \( y_1 \). The subgraph of \( G \) induced by \( \{p, y_1, y_2, y, q\} \) is either a house or a gem as shown in Fig. 2(c) depending on whether \( y_1 \) and \( y_2 \) are adjacent in \( G \). The only way to destroy the house or the gem is to make \( y \) a nonprobe and to add edge \( (y, q) \) in the embedding. Thus all vertices in \( Y \) must be nonprobes in any distance-hereditary embedding. Let \( x \) be a vertex in \( X \). If \( x \) is adjacent to some vertex in \( Y \), then \( x \) must be a probe in any embedding. Suppose \( x \) is not adjacent to any vertex in \( Y \). After adding edge \( (y, q) \), \( \{y_2, x, p, y, q\} \) induces a gem, where \( y_2 \) is the universal vertex and \( xpyq \) is the \( P_4 \) in the embedding. For destroying the gem, \( x \) must be a nonprobe in any distance-hereditary embedding. Let \( Q = \{q' \mid q' \in L_2, q' \neq q, N^*_G(q) \subseteq N^*_G(q')\} \). If \( q' \in Q \) is adjacent to some vertex in \( Y \), then it must be a probe in any distance-hereditary embedding. Suppose \( q' \in Q \) is not adjacent to any vertex of \( Y \). Similar to the case for vertex \( x \in X \) not adjacent to any vertex in \( Y \), we can show that \( q' \) must be a nonprobe in any distance-hereditary embedding. From the arguments above we see that every vertex in \( N_G(N^*_G(q)) \) has a unique label among all distance-hereditary embeddings of \( P \). Let \( P^* \) be the probe graph \( (P_G^*, P_L^*) \) where \( P_G^*(q) = \mathbb{N} \) for \( y \in Y \), \( P_L^*(x) = \mathbb{N} \) for \( x \in X \) not adjacent to any vertex in \( Y \), \( P_L^*(x) = P \) for \( x \in X \) adjacent to some vertex in \( Y \), \( P_L^*(q) = \mathbb{N} \) for \( q' \in Q \) not adjacent to any vertex in \( Y \), \( P_L^*(q') = \mathbb{N} \) for \( q' \in Q \) adjacent to some vertex in \( Y \), and \( P_L^*(u) = P_L(u) \) for
\(u \in P_v - (X + Y + Q)\). From the above arguments, we see that \(\hat{P}\) is a probe distance-hereditary graph if and only if \(P^*\) is a probe distance-hereditary graph. Let \(p'\) be some vertex in \(N_G(q)\). We see that \(p'\) is a probe in \(P^*\) and \(P^*_L(u) \neq U\) for every \(u \in N_G(p')\). Thus \(P^*\) is well-labeled with respect to \(p'\) (C 1). Hence we can call Algorithm \text{W} to complete the job in \(O(n^2)\) time.

![Figure 2: Some induced subgraphs in P where a dotted line denotes two vertices are possibly adjacent or possibly not adjacent.](image)

The following lemma summarizes the results of this subsection.

**Lemma 10.** Whether a probe graph of class \(P_4\) is a probe distance-hereditary graph can be determined in \(O(n^2)\) time.

### 3.3 Non-biconnected probe graphs without twins and Algorithm \(R\)

In this subsection we show how to solve the problem recursively when the input probe graph \(P\) has no twins and is not biconnected. Our algorithm is based upon the following two lemmas.

**Lemma 11.** Suppose \(P\) is a connected probe graph and \(P^*\) is a minimal distance-hereditary embedding of \(P\). Then a vertex is a cut vertex of \(P^*\) if and only if it is a cut vertex of \(P\).

**Proof.** Suppose \(P\) has \(k\) biconnected components \(C_1, C_2, \ldots, C_k\). Let \(P^*_G\) be the graph \((P^*_V, \bigcup_{i=1}^{k} P_i^*[C_i])\). It is easy to see that a vertex is a cut vertex of \(P^*_G\) if and only if it is a cut vertex of \(P\). We then prove the lemma by showing that \(P^*\) is indeed a distance-hereditary embedding of \(P\). If \(P^*\) is a distance-hereditary embedding of \(P\) and \(P^* \neq P^*\), then \(P^*\) is not minimal, a contradiction. Thus \(P^* = P^*\) if \(P^*\) is a distance-hereditary embedding of \(P\).

Now we prove that \(P^*\) is a distance-hereditary embedding of \(P\). Suppose that \(P^*\) is not a distance-hereditary embedding of \(P\). That is, \(P^*\) has a forbidden induced subgraph of distance-hereditary graphs. Let \(F\) be the vertex set of a forbidden induced subgraph. Since \(P^*[C_i]\) is a distance-hereditary embedding of \(P[C_i]\), \(F\) is not a subset of any \(C_i\) for \(1 \leq i \leq k\). Notice that \(F\) induces a hole, a gem, a house, or a domino. All these four forbidden induced subgraphs are biconnected. Thus \(F\) must be a subset of some \(C_i\), a contradiction. This completes the proof.

**Lemma 12.** Let \(P\) be a probe graph. If there exists a cut vertex \(v\) in \(P\) and \(C\) is a component of \(P - v\), then \(P\) is a probe distance-hereditary graph if and only if \(P - C\) has an embedding \(P^*\) and \(P^*[C + v]\) has an embedding \(P^*\) where either \(P^*_L(v) = \overline{P}^*_L(v) = \mathbb{P}\) or \(P^*_L(v) = \overline{P}^*_L(v) = \overline{\mathbb{N}}\).

**Proof.** If \(P\) has a distance-hereditary embedding \(P^*\), then \(P^*[C + v]\) is a distance-hereditary embedding of \(P\) \(P^* - C\) is a distance-hereditary embedding of \(P - C\).

Suppose that \(P^*_L(v) = \mathbb{P}\) or \(\overline{\mathbb{N}}\). By Lemma 11, \(P^* = P^* + P^*\) is a distance-hereditary embedding of \(P\).

Suppose that \(P^*_L(v) = U\). If \(P[C + v]\) has no distance-hereditary embedding that \(v\) is a probe or a nonprobe, then \(P\) is not a probe distance-hereditary graphs. If \(P[C + v]\) has a distance-hereditary embedding \(P\) that \(v\) is a probe but has no distance-hereditary embedding that \(v\) is a nonprobe, then \(P\) has a distance-hereditary embedding if and only if \(P - C\) has a distance-hereditary embedding that \(v\) is a nonprobe. Conversely, if \(P[C + v]\) has a distance-hereditary embedding \(P\) that \(v\) is a probe and \(P[C + v]\) has a distance-hereditary embedding \(P\) that \(v\) is a nonprobe, then \(P\) has a distance-hereditary embedding if and only if \(P - C\) has a distance-hereditary embedding.

The proof of the above lemma is constructive. It points out a recursive way to solve the problem. We now describe Algorithm \(R\) in detail. Let \(v\) be a cut vertex of \(P_G\) and \(C\) be a component of \(P_G - v\) such that \(C\) does not contain any other cut vertex of \(P_G\).

In other words, \(C + v\) induces a biconnected component of \(P_G\). There are two cases:

1. \(P_L(v) = \mathbb{P}\) or \(\overline{\mathbb{N}}\). By Lemma 12, \(P\) is a probe distance-hereditary graph if and only if both \(P[C + v]\) and \(P - C\) are probe distance-hereditary graphs. Call Algorithm \(C\) to check whether \(P[C + v]\) has an embedding and recursively call the main algorithm to check whether \(P - C\) has an embedding.

2. \(P_L(v) = U\). Let \(P\) be the probe graph \((P_G[C + v], P_L)\) where \(P_L(v) = \mathbb{P}\) and \(P_L(x) = P(x)\) for \(x \in C\). Let \(P\) be the probe graph \((P_G[C + v], P_L)\) where \(P_L(v) = \mathbb{N}\), \(P_L(x) = \mathbb{P}\) for \(x \in N_G(v) \cap C\), and \(P_L(x) = P_L(x)\) for \(x \in C - N_G(v)\). Let \(P^*\) be the probe graph \((P_G[V - C], P^*_L)\) where \(P^*_L(v) = \mathbb{P}\) and \(P^*_L(x) = P^*_L(x)\) for \(x \in V - C - v\). Let \(P^*_L\) be the probe graph \((P_G[V - C], P^*_L)\) where \(P^*_L(v) = \mathbb{N}\), \(P^*_L(x) = \mathbb{P}\) for \(x \in (P - C) \cap N_G(v)\), and \(P^*_L(x) = P^*_L(x)\) for \(x \in V - C - N_G(v)\). Call Algorithm \(C\) to check whether \(P\) has embeddings. There are four subcases:

(a) If neither \(\hat{P}\) nor \(\hat{P}\) is a probe distance-hereditary graph, then \(P\) is not a probe distance-hereditary graph.

(b) If both \(\hat{P}\) and \(\hat{P}\) are probe distance-hereditary graphs, then \(P\) is a probe distance-hereditary graph if and only if \(P - C\) is a probe distance-hereditary graph. Recursively call the main algorithm to check whether \(P - C\) has an embedding.

(c) If \(\hat{P}\) is a probe distance-hereditary graph but \(\hat{P}\) is not, then \(P\) is a probe distance-hereditary graph if and only if \(P^*\) is a probe distance-hereditary graph. Recursively call the main algorithm to
check whether $P'$ has a distance-hereditary embedding.

(d) If $P$ is not a probe distance-hereditary graph but $\bar{P}$ is, then $P$ is a probe distance-hereditary graph if and only if $P''$ is a probe distance-hereditary graph. Recursively call the main algorithm to check whether $P''$ has a distance-hereditary embedding.

We call a probe graph $P$ a pseudo-kernel probe graph if it satisfies one of the following three conditions: (i) $P$ is biconnected without twins, (ii) $P$ is biconnected and has only one pair of twins. One of the pair of twins is not a prime, (iii) $P$ is non-biconnected without twins and has only one cut vertex. The cut vertex is not a prime.

Suppose $v$ is the cut vertex of $P_G$, used to decompose $P_G$ into $P_{G[C + v]}$ and $P_G - C$ in Algorithm $R$. We have the following observations. For simplifying the notation, we use $G$ and $G_{C + v}$ to denote $P_G$ and $P_{G[C + v]}$ respectively.

Theorem 10. Suppose $G$ is a non-biconnected graph without twins. There exists a biconnected component $G_{C + v}$ of $G$ that only contains a cut vertex $v$ of $G$. Then one of the following statements holds.

(i) There are no twins in $G_{C + v}$.

(ii) There is only one pair of twins in $G_{C + v}$, $v$ is one of the pair of twins. After removing one of the pair of twins from $G_{C + v}$, the resulting graph has no twins and either it is biconnected or it has only one cut vertex which is one of the pair of twins in $G_{C + v}$.

Proof. Since $G$ has no twins, no $x, y \in G_C$ are twins in $G_{C + v}$. After the decomposition, only the neighborhood of the vertex used to decompose the graph is changed. Hence $x$ must be one of the pair of twins, and there exists only one vertex $u$ in $G_{C}$ that $u$ and $v$ are twins in $G_{C + v}$.

Suppose there exists a pair of twins $x$ and $y$ in $G_C$. Removing $v$ from $G_{C + v}$ only changes the neighborhood of vertices in $N_{G_{C}}[u]$, where $u$ and $v$ are twins in $G_{C + v}$. Hence one of $x, y \notin N_{G_C}[u]$. If $u = x$ and $y \notin N_{G_C}[u]$ and $v$ are twins in $G_{C + v}$, a contradiction. If $u = x$ and $y \notin N_{G_C}[u]$ and $y$ are twins in $G_{C + v}$, a contradiction. Suppose $x \neq u$ and $y \neq u$. If $x, y \notin N_{G_C}(u)$, they are twins in $G_{C + v}$, a contradiction. If $x \in N_{G_C}(u)$ but $y \notin N_{G_C}[u]$, then they are not twins in $G_C$ since $y$ is not adjacent to $u$, $u$ is not adjacent to $u$, $y$ is not adjacent to $u$, $u$ is not adjacent to $u$, $y$ is not adjacent to $u$, $y$ is not adjacent to $u$.

Suppose $G_{C + v}$ is non-biconnected. If $u$ is not a cut vertex, let $x \neq u$ be a cut vertex of $G_{C}$. Let $C_1$ and $C_2$ be two components of $G_{C} - x$. Since $G_{C + v}$ is biconnected, $v$ is adjacent to some vertex of $C_1$ and some vertex of $C_2$ in $G_{C + v}$. Since $u$ and $v$ are twins in $G_{C + v}$, $v$ is adjacent to some vertex of $C_1$ and some vertex of $C_2$, a contradiction to the assumption that $x \neq u$ is a cut vertex. Hence $u$ is the only cut vertex in $G_{C}$. Similarly, we can show $v$ is the only cut vertex of $G_{C + v} - u$.

Corollary 7. The probe graph $P[C + v]$ produced in Case 1. of Algorithm $R$ and the probe graphs $\bar{P}$ and $\bar{P}$ produced in Case 2. of Algorithm $R$ are pseudo-kernel probe graphs. In addition, if $u$ and $v$ are the only pair of twins in $P_G[C + v]$, after removing a twin according to Lemma 4 and from $P[C + v]$, $\bar{P}$, and $\bar{P}$, the resulting probe graph $R$ is a pseudo-kernel probe graph.

Proof. Note that $P_G[C + v]$ satisfies one of the conditions of Theorem 10 and $v$ is not a prime. Assume $u$ is the twin of $v$. By the steps of removing twins according to Lemma 4 and 5, if $v$ is a probe, we remove $u$; if $v$ is a nonprobe and $u$ is a nonprobe or a prime, we remove $u$; if $v$ is a nonprobe and $u$ is a probe, we remove $v$. After removing twins, if the resulting probe graph $R$ is biconnected, by Theorem 10 it is a kernel probe graph. Assume $R$ is non-biconnected, by Theorem 10 one of $u$ and $v$ is the only cut vertex of $R$. Moreover, the only cut vertex in $R$ is not a prime.

Now we are ready to describe Algorithm $C$. The input of Algorithm $C$ is a two-tuple $(P, v)$ where $P$ is a pseudo-kernel probe graph and $v$ is a vertex in $P$ with $P_L(v) = \mathbb{P}$ or $\mathbb{N}$. Note that if $P_G$ is biconnected, it has at most one pair of twins and $v$ is one of the twins. If $P_G$ is non-biconnected, $v$ is the only cut vertex in $P_G$.

Algorithm $C$. We distinguish the following four classes of the input graphs.

E 1 $|P_V| \leq c$ for some constant $c$. Solve the problem by brute force in $O(1)$ time.

E 2 $P$ is biconnected without twins. Call Algorithm $B$ to solve the problem in $O(n^2)$ time.

E 3 $P_G$ is biconnected and $v$ has a twin $u$. It is easy to see that $u$ can be found in linear time by simply checking whether the neighborhood of the other vertices is the same as $v$. By Lemma 4 and 5, we can remove one of $u$ and $v$ from $P$, and check whether the resulting probe graph is a probe distance-hereditary graph. We have the following two cases:

(1) $P_L(v) = P$. Recursively call Algorithm $C$ to check $(P - u, v)$.

(ii) $P_L(v) = \mathbb{N}$. If $P_L(u) = \mathbb{N}$ or $\mathbb{U}$, recursively call Algorithm $C$ to check $(P - u, v)$. If $P_L(u) = \mathbb{P}$, recursively call Algorithm $C$ to check $(P - v, u)$.

E 4 $P_G$ is non-biconnected without twins, $v$ is the only cut vertex in $P_G$. Let $C_1, C_2, \ldots, C_r$ be biconnected components of $P_G$. Since $v$ is the only cut vertex in $P_G$, $C_i \cap C_j = \{v\}$ for $1 \leq i < j \leq r$. For each $C_i$, $i = 1, 2, \ldots, r$, call Algorithm $C$ to check $(P[C_i], v)$.

Lemma 13. Whether a pseudo-kernel probe graph is a probe distance-hereditary graph can be checked in $O(nm)$ time.

Proof. Let $g(n)$ denote the time complexity of Algorithm $C$. We claim that $g(n) \leq c_1nm$. The input graph of class E 1 can be recognized in $O(1)$ time. The input graph of class E 2 can be recognized in $O(n^2)$ time. The input graph of class E 3 can be recognized in $g(n) + c_0(n + m)$ time where $c_0(n + m)$ is the time spent for decomposing the input graph into biconnected components and removing a twin from it. It is easy to see that $g(n - 1) + c_0(n + m) \leq c_1nm$ if $c_1 \geq 2c_0$. The input of class E 4 can be recognized in $2^{2}g(n) + c_0(n + m)$ time where $n_e = |C_i|$ and $C_1, C_2, \ldots, C_r$ are biconnected components in $P_G$. 63
This completes the proof. □

3.4 Time complexity

In this subsection we analyze the time complexity of the algorithm.

Theorem 11. There exists an \( O(nm) \)-time algorithm to check if a probe graph \( P \) is a probe distance-hereditary graph.

Proof. By using the data structure described in (Lanlignel & Thierry 2000), we can repeat the step of removing twins until input probe graph \( P \) has no twins in \( O(n^2) \) time. If \( P \) is biconnected after removing all twins, then call Algorithm B to complete the algorithm. Suppose \( P \) is not biconnected after removing all twins. We go on performing the recursive step that decomposes \( P \) into two subgraphs \( P_C[\{C+v\}] \) and \( P_C - C \) (Algorithm R). The algorithm calls Algorithm C to test \( P[C+v] \) (Case 1 of Algorithm R) or \( P \) and \( P \) (Case 2 of Algorithm R) and goes on testing the subgraph \( P - C \) obtained recursively. Note that removing all twins from \( P - C \) only takes \( O(n+m) \) time. After removing \( C \) from \( P \) only the neighborhood of the cut vertex \( v \) is changed. Since \( P \) has no twins, there is only one pair of twins in \( P - C \) and \( v \) is one of the twins. Let \( t(n) \) be the time of the whole algorithm. Then \( t(n) = t_1(n') + O(n^2) \) where \( n' \) is the number of vertices in the input probe graph after removing all twins and \( t_1(n') \) is the time spent by the algorithm after removing all twins. Let \( C_1, C_2, \ldots, C_k \) be the biconnected components produced by Algorithm R in each recursive call. For each \( C_i \) we call Algorithm C at most two times. Assume \( |C_i| = n_i \) for \( i = 1, \ldots, k \). Let \( m_i \) denote the number of edges in \( P_{C_i[\{C\}]}. \) We use \( g(n_i) \) to denote the time spent by Algorithm C for \( i = 1, \ldots, k. \)

\[
t_1(n') = 2g(n_1) + t_1(n' - n_1 - 1) + c_0(n' + m')
\]
\[
= 2g(n_1) + \cdots + 2g(n_k) + c_0(k(n' + m'))
\]
\[
= 2\sum_{i=1}^k g(n_im_i) + c_0k(n' + m')
\]
\[
= 2\sum_{i=0}^k c_1m_i + c_0k(n' + m')
\]
\[
\leq c_1\sum_{i=1}^k m_i + c_0k(n' + m')
\]
\[
\leq c_2njm' + c_0(n' + m')
\]
\[
= O(n'm')
\]

Since \( t(n) = t_1(n') + O(n^2) \) and \( t_2(n') = O(n'm') \), we have \( t(n) = O(nm) \). This completes the proof of the theorem. □

References


