Vol.2009-AL-126 No.5
2009/9/15

IPSJ SIG Technical Report

# Reconstruction Algorithms for Permutation Graphs and Distance-Hereditary Graphs

Masashi Kiyomi,[†1] Toshiki Saitoh [†1]
and Ryuhei Uehara[†1]

PREIMAGE CONSTRUCTION problem by Kratsch and Hemaspaandra naturally arose from the famous graph reconstruction conjecture. It deals in the algorithmic aspects of the conjecture. We present an $O(n^6)$ time algorithm for PREIMAGE CONSTRUCTION on permutation graphs. A simplified algorithm can be applied for PREIMAGE CONSTRUCTION on distance-hereditary graphs. There are polynomial time isomorphism algorithms for permutation graphs. However the number of permutation graphs obtained by adding a vertex to a permutation graph may be exponentially large. Thus exhaustive checking of these graphs does not achieve any polynomial time algorithm. Therefore reducing the number of preimage candidates is the key point.
**Keywords:** the graph reconstruction conjecture, permutation graphs, distance-hereditary graphs, polynomial time algorithm

## 1. Introduction

The graph reconstruction conjecture proposed by Ulam and Kelly[⋆1] has been studied by many researchers intensively. We call the multi-set $\{G-v \mid v \in V\}$ the *deck* of a graph $G = (V, E)$, where $G - v$ is a graph obtained from $G$ by removing $v$ and incident edges. More precisely the graphs in a deck are vertex-unlabeled. Otherwise the argument below has no mean. A graph $G$ is a *preimage* of a deck of a graph $G'$ if $G$ and $G'$ has the same deck. We also say that a graph $G$ is a preimage of the $n$ graphs when the deck of $G$ exactly consists of them. The graph reconstruction conjecture is that there is at most one preimage of given $n$ graphs ($n \geq 3$). No one has given a positive nor a negative proof of this conjecture, while small graphs are checked positively[15]. Kelly showed the following lemma.

**Lemma 1** (Kelley's Lemma[11])**.** *Let $G$ be any preimage of the given deck, and let $H$ be a graph whose number of vertices is smaller than that of $G$. Then we can uniquely determine the number of subgraphs in $G$ isomorphic to $H$ from the deck.*

Greenwell and Hemminger extended this lemma to a more general form[8]. We can know the degree sequence of a preimage from these lemmas. Kelly also showed that the conjecture is true on regular graphs, trees, and disconnected graphs. Tutte proved that the dichromatic rank and Tutte polynomials are reconstructible (i.e. looking at the deck, they are uniquely determined)[20]. Bollobás showed that almost all graphs are reconstructible from three well-chosen graphs in its deck[2]. About permutation graphs, Rimscha showed that permutation graphs are recognizable in the sense that looking at the deck of $G$ one can decide whether or not $G$ belongs to permutation graphs[17]. To be precise Rimscha showed in the paper that comparability graphs are recognizable. Even's result[6] directly gives a proof in the case of permutation graphs. Rimscha also showed in the same paper that many subclasses of perfect graphs including perfect graphs themselves are recognizable, and some of subclasses are reconstructible. There are jillion of papers about the conjecture, and many good surveys about this conjecture. See for example[3),9)].

There are several kinds of algorithmic problems related to the graph reconstruction conjecture. We consider algorithmic problems proposed by Kratsch and Hemaspaandra[13] described below.

- Given a graph $G$ and a multi-set of graphs $D$, check whether $D$ is the deck of $G$ (DECK CHECKING).
- Given a multi-set of graphs $D$, determine whether there is a graph whose deck is $D$ (LEGITIMATE DECK).
- Given a multi-set of graphs $D$, construct a graph whose deck is $D$ (PREIMAGE CONSTRUCTION).
- Given a multi-set of graphs $D$, compute the number of (pairwise nonisomorphic) graphs whose decks are $D$ (PREIMAGE COUNTING).

Kratsch and Hemaspaandra showed that these problems are solvable in polynomial time for graphs of bounded degree, partial $k$-trees for any fixed $k$, and graphs

†1 Japan Advanced Institute of Science and Technology, Asahidai 1-1, Nomi, Ishikawa 923-1292, Japan. {mkiyomi,toshikis,uehara}@jaist.ac.jp

⋆1 Determining the first person who proposed the graph reconstruction conjecture is difficult, actually. See[9] for the detail.

of bounded genus, in particular for planner graphs[13]. In the same paper they proved many GI related complexity results. Hemaspaandra et al. extended the results[10]. The authors presented a polynomial time PREIMAGE CONSTRUCTION algorithm for interval graphs[12].

We present an $O(n^6)$ time algorithm for PREIMAGE CONSTRUCTION on permutation graphs. A simplified algorithm can be applied for PREIMAGE CONSTRUCTION on distance-hereditary graphs. In order to make things clear, we introduce a new graph class *weakly distance-hereditary graphs* as graphs whose each connected component is distance-hereditary. It is easy to see that every graph in the deck of a permutation graph is a permutation graph, and that every graph in the deck of a weakly distance-hereditary graph is weakly distance-hereditary. It is also easy to see that a deck of a connected graph has at least two connected graphs. Since a distance-hereditary graphs is equivalently a connected weakly distance-hereditary graph, every graph in the deck of a distance-hereditary graph is weakly distance-hereditary, and at least two of them are connected. We propose PREIMAGE CONSTRUCTION algorithm for a deck consisting of permutation graphs, and that consisting of weakly distance-hereditary graphs at least two of which are connected. We state our main theorems below.

**Theorem 1.** *There is an $O(n^6)$ time PREIMAGE CONSTRUCTION algorithm for a deck $D$ consisting of $n$ permutation graphs.*

**Theorem 2.** *There is an $O(n^3m)$ time PREIMAGE CONSTRUCTION algorithm for a deck $D$ consisting of $n$ weakly distance-hereditary graphs at least two of which are connected, where $m$ is the number obtained by dividing the total number of edges in $D$ by $n-2$.* ⋆1

## 2. Notations and definitions

All the graphs in this paper are simple unless stated otherwise. A direct graph $G = (V, E)$ is *oriented* if there are no two vertices $v$ and $u$ in $V$ such that both $(v, u)$ and $(u, v)$ are in $E$. An oriented graph $G = (V, E)$ is *transitively oriented* if there is an edge $(u, w)$ in $E$ for every three vertices $u, v$ and $w$ in $V$ such that $(u, v)$ and $(v, w)$ are in $E$. An undirected graph $G$ is *transitively orientable* if it is an underlying graph of some transitively oriented graph $G'$, and we call $G'$ a *transitive orientation* of $G$. A transitively orientable graph is *uniquely orientable* if it has exactly two transitive orientations. Note that if there is a transitive orientation of an undirected graph $G$, then the graph obtained by reversing every direction of the orientation is also a transitive orientation. Thus if $G$ is transitively orientable, $G$ has at least two transitive orientations. This definition of uniquely orientability is equivalent that an undirected graph $G = (V, E)$ is uniquely orientable if and only if for every pair of edges $e$ and $f$ in $G$ there is an edge sequence $(e_1, \ldots, e_k)$ such that $e_1$ is equal to $e$, $e_k$ is equal to $f$, and $e_i$ and $e_{i+1}$ are $\Gamma$-related for all $i \in \{1, \ldots, k-1\}$, where edge $\{p, q\}$ and $\{r, s\}$ are $\Gamma$-related if $q = r$ and $\{p, s\} \notin E$ hold[5]. See Fig. 1 for an example. Note that two identical edges $\{p, q\}$ and $\{q, p\}$ in a simple graph are $\Gamma$-related.

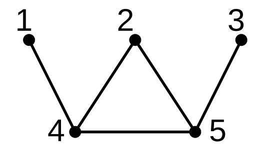


**Fig. 1** An example of a uniquely orientable permutation graph. It is a bit confusing to find a $\Gamma$-related edge sequence beginning from $\{2, 4\}$ and ending at $\{2, 5\}$. There is $(\{2, 4\}, \{1, 4\}, \{4, 5\}, \{3, 5\}, \{2, 5\})$ as such an edge sequence.

We denote by $N(v)$ the neighbor set of vertex $v$, and by $N[v]$ the closed neighbor set of vertex $v$. "Closed" means that $N[v]$ contains $v$ itself. Vertices $u$ and $v$ are called *strong twins* if $N[u]$ is equal to $N[v]$, and *weak twins* if $N(u)$ is equal to $N(v)$. A vertex $v$ is called a *pendant* if $v$ is a degree one vertex.

We call a graph $G$ is *reducible* if there are twins in $G$. The graph obtained by repeatedly contracting twins from a graph $G$ until it has no twin is the *irreducible reduction* of $G$, and we denote it by $R(G)$. We call a graph $G$ is *prunable* if there are twins or a pendant in $G$. The graph obtained by repeatedly contracting twins or removing a pendant from a graph $G$ until it has no twin and pendant is the *unprunable reduction* of $G$, and we denote it by $R'(G)$. Note that $R(G)$ and $R'(G)$ are well-defined[16].

An $n$-vertex undirected graph $G = (V, E)$ is a *permutation graph* if $G$ has a permutation representation, that is, there exist a labeling $\{v_1, v_2, \ldots, v_n\}$ of the

⋆1 By Lemma 1 the number of edges in every preimage is $\sum_{i=1}^{n} |E(G_i)|/(n-2)$[11].

vertices in $V$ and a permutation $\pi$ of $\{1, 2, \ldots, n\}$ such that $(i-j)(\pi(i)-\pi(j)) < 0$ if and only if $\{v_i, v_j\}$ is in $E$. An undirected graph $G$ is *distance-hereditary* if and only if $G$ is connected, and for any two distinct vertices $x$ and $y$ in $G$ the length of induced paths between $x$ and $y$ are the same. It is convenient to define a new graph class not to care if graphs are connected. We define an undirected graph $G$ is *weakly distance-hereditary* if and only if for any two distinct vertices $x$ and $y$ in $G$ the length of induced paths between $x$ and $y$ are the same.

Given two graphs $G_1$ and $G_2$, we define the disjoint union $G_1 \dot\cup G_2$ of $G_1$ and $G_2$ as $(V_1 \dot\cup V_2, E_1 \dot\cup E_2)$ such that $(V_1, E_1)$ is isomorphic to $G_1$, and $(V_2, E_2)$ is isomorphic to $G_2$, where $\dot\cup$ means the disjoint union.

We denote by $u(G)$ the graph obtained by adding one universal vertex to the graph $G$ such that the vertex connects to every vertex in $G$.

Let $P$ be an ordering of elements in a set $V$. For $v \in V$ we denote by $P - v$ the ordering of the elements $V \setminus \{v\}$ in which all the elements appear in the same order in $P$. We denote by $P +_i v$ the orderings obtained by inserting $v$ to the $i$th position of $P$. We denote by $\overline{P}$ the reverse ordering of $P$.

## 3. Previous works

It is known that an undirected graph $G$ is a permutation graph if and only if $G$ and the complement of $G$ is transitively orientable[6]. Moreover it is known that permutation graphs are precisely the comparability graphs of 2-dimensional partial orders[1]. That is, for any permutation graph $G = (V, E)$ there are two orderings of the vertices in $V$ such that $\{u, v\}$ is in $E$ if and only if $u$ proceeds $v$ in both the orderings (we say the two orderings identify $G$). And it turns out that if $G' = (V', E')$ is a transitive orientation of a permutation graph, there is a pair of vertex orderings such that $(u, v)$ is in $E'$ if and only if $u$ proceeds $v$ in both the orderings. Specifically for an uniquely orientable permutation graph $G$, $G$ has just two pairs of vertex orderings, one pair $P$ and $Q$ of which corresponds to one of the two transitive orientations of $G$, and the other, $\overline{P}$ and $\overline{Q}$, corresponds to the other transitive orientation.

Colbourn showed that $R(G)$ is uniquely orientable if $G$ is a connected permutation graph[5]. [⋆1] Bandelt and Mulder showed that $R'(G)$ consists of a singleton if and only if $G$ is distance-hereditary[4].

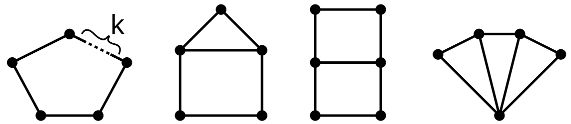


**Fig. 2** Forbidden graphs of distance-hereditary graphs. The part described $k$ contains $k$ vertices ($k \geq 0$).

Nakano et al. proposed a DH-tree for a distance-hereditary graph[16]. We can see the tree as a uniquely defined canonical form of distance-hereditary graphs isomorphic to each other.

Bandelt and Mulder[4] showed that a graph $G$ is distance-hereditary if and only if $G$ is connected, and (hole, house, domino, gem)-free, that is, $G$ has none of the graphs in Fig. 2 as an induced subgraph. Gallai characterized comparability graphs with the forbidden subgraphs[7]. Since permutation graphs are equivalent to comparability and co-comparability graphs[6], the characterization of permutation graphs is easily obtained. A graph $G$ is a permutation graph if and only if $G$ is ($\mathrm{C}_{k+6}$, $\mathrm{T}_2$, $\mathrm{X}_2$, $\mathrm{X}_3$, $\mathrm{X}_{30}$, $\mathrm{X}_{31}$, $\mathrm{X}_{32}$, $\mathrm{X}_{33}$, $\mathrm{X}_{34}$, $\mathrm{X}_{36}$, $\mathrm{XF}_1^{2k+3}$, $\mathrm{XF}_2^{k+1}$, $\mathrm{XF}_3^{k}$, $\mathrm{XF}_4^{k}$, $\mathrm{XF}_5^{2k+3}$, $\mathrm{XF}_6^{2k+3}$, co-$\mathrm{C}_{k+6}$, co-$\mathrm{T}_2$, co-$\mathrm{X}_2$, co-$\mathrm{X}_3$, co-$\mathrm{X}_{30}$, co-$\mathrm{X}_{31}$, co-$\mathrm{X}_{32}$, co-$\mathrm{X}_{33}$, co-$\mathrm{X}_{34}$, co-$\mathrm{X}_{36}$, co-$\mathrm{XF}_1^{2k+3}$, co-$\mathrm{XF}_2^{k+1}$, co-$\mathrm{XF}_3^{k}$, co-$\mathrm{XF}_4^{k}$, co-$\mathrm{XF}_5^{2k+3}$, co-$\mathrm{XF}_6^{2k+2}$, and odd-hole)-free. See Fig. 3.

## 4. Algorithm

Our algorithm outputs preimages that are permutation graphs (or distance-hereditary graphs). However it is possible that a non-permutation (non-distance-hereditary) graph has a deck that consists of permutation graphs (distance-hereditary graphs), though it is exceptional. Since considering this case all the time in the main algorithm makes it complex, we attempt to get done with this special case in subsection 4.1.

Then we present our main algorithms. The reconstruction algorithm for per-

⋆1 This sentense is not correct, but we use it in the algorithm. We will fix the algorithm before the presentation.

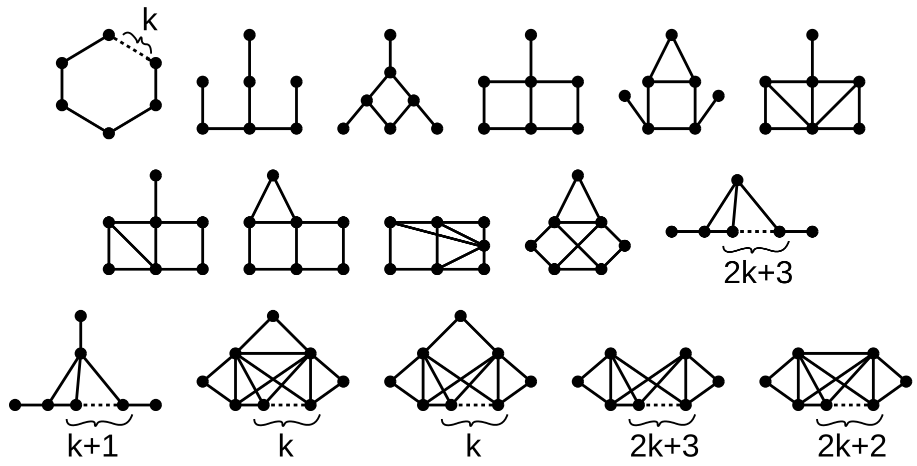


**Fig. 3** Forbidden graphs of permutation graphs are these graphs, the complements of them, and odd-holes.

mutation graphs consists of two phases. In the first phase we assume that the preimages are reducible. The second phase is for the case that the preimages are irreducible and hence are uniquely orientable. The reconstructions of distance-hereditary graphs are done only using a slightly modified first phase algorithm. In both phases the algorithm lists up the preimage candidates, and checks if they are really the preimages by the DECK CHECKING algorithm which we describe in subsection 4.2.

Our main algorithms assume that the preimages are connected graphs. This assumption is true on distance-hereditary graphs. However there are disconnected permutation graphs that are preimages of some decks. Thus we consider the case that the preimages are disconnected permutation graphs in subsection 4.5.

### 4.1 non-permutation, non-distance-hereditary graph preimage case

Graph $G$ is distance-hereditary if and only if it is connected, and it has no cycle of length more than five, no house, no domino, and no gem as an induced subgraph[4)]. This means that the forbidden graphs of weakly distance-hereditary graphs are cycles of length more than five, a house, a domino, and a gem. If a connected graph $G$ is not distance-hereditary (it turns out that $G$ has some forbidden graph as an induced subgraph), and if $G$ has a deck consisting of weakly distance-hereditary graphs, and at least two of them are connected, then $G$ must be the one of a cycle of length more than five, a house, a domino, or a gem, since otherwise some graphs in the deck have the forbidden induced subgraphs. We can check if the input deck is a deck of a house, of a domino, or of a gem in constant time, since the size of these graphs are constant. We can check if the input deck is a deck of a cycle in $O(n^2)$ time, since the deck of a cycle of length $n$ consists of $n$ paths of length $n-2$. Thus we have the theorem below.

**Theorem 3.** *If $n$ weakly distance-hereditary graphs $G_1, G_2, \ldots, G_n$ including at least two connected graphs have a non-distance-hereditary preimage $G$, we can reconstruct $G$ from $G_1, G_2, \ldots, G_n$ in $O(n^2)$ time.*

Since the forbidden graphs of permutation graphs are more complicated than those of weak distance-hereditary graphs, we need to be more careful in the case of deck consisting of permutation graphs.

Let $D$ be a deck consisting of $n$ graphs $G_1, G_2, \ldots, G_n$. It is clear that $G_1, G_2, \ldots, G_n$ have the same number of vertices $n-1$, and that the number of vertices in a preimage $G$ is $n$. Since the number of the forbidden graphs of size $n$ is $O(1)$, we can check if one of them is a preimage of the input graphs in the polynomial time with DECK CHECKING algorithm which we will describe in the next subsection. The time complexity is $O(n^4)$, since the time complexity of the DECK CHECKING algorithm is $O(n^4)$.

**Theorem 4.** *If $n$ permutation graphs $G_1, G_2, \ldots, G_n$ have a preimage $G$ that is not a permutation graph, we can reconstruct $G$ from $G_1, G_2, \ldots, G_n$ in $O(n^4)$ time.*

### 4.2 DECK CHECKING

Since an $O(n^2)$ time isomorphism algorithm for permutation graphs[19)], and an $O(m)$ time isomorphism algorithm for distance-hereditary graphs[16)] are known, developing polynomial time DECK CHECKING algorithms for permutation graphs and distance-hereditary graphs are not very difficult.

Given a deck $D$ that consists of permutation graphs, and given a preimage candidate $G=(V,E)$ which is a permutation graph, we first prepare the deck $\hat{D}$ of $G$ in $O(|V|(|V|+|E|))$ time. We then add a universal vertex to every graph in $D$ and $\hat{D}$ in order to make each graph connected. Note that for any permutation graph $G$, $u(G)$ is also a permutation graph. Since the disjoint union of permutation graphs is clearly a permutation graph, we can check if $D$ and $\hat{D}$ are equivalent in $\mathrm{O}((|V|(|V|+1))^2)=\mathrm{O}(|V|^4)$ time by applying the isomorphism algorithm for permutation graphs to the disjoint union of graphs in $D$ and the disjoint union of graphs in $\hat{D}$. Now we obtain the theorem below.

**Theorem 5.** *There is $O(|V|^4)$ time DECK CHECKING algorithm for a deck that consists of permutation graphs, and a preimage candidate $G = (V, E)$ which is a permutation graph.*

We have to be more careful in the case of distance-hereditary graphs, since a distance-hereditary graph must be connected, and adding a universal vertex breaks (weakly) distance-hereditariness.

**Lemma 2.** *For two weakly distance-hereditary graphs $G_1$ and $G_2$, we can check if $G_1$ and $G_2$ are isomorphic in $O(n+m)$ time, where $n$ is the number of vertices in $G_1$ (and of course in $G_2$), and $m$ is the number of edges in $G_1$.*

*Proof.* The $\mathrm{O}(m)$ isomorphism algorithm in[16] does not explicitly use the property that distance-hereditary graphs are connected. It makes two DH-trees corresponding to the two input distance-hereditary graphs, and compare them. Each node of a DH-tree corresponds to an operation of adding twins or adding pendants, and the root corresponds to $\mathrm{K}_2$. We only have to replace the root $\mathrm{K}_2$ by $\mathrm{K}_1$. Since adding $k-1$ weak twins to $\mathrm{K}_1$ results in $k$ isolated vertices, we can generate any disconnected weakly distance-hereditary graphs from $\mathrm{K}_1$. It is straightforward to modify the algorithm in[16] to handle such a case without affecting the time complexity. □

Moreover the following lemma is useful.

**Lemma 3.** *Given two sets of weakly distance-hereditary graphs $S_1 = \{G_1, \ldots, G_k\}$ and $S_2 = \{G'_1, \ldots, G'_k\}$, we can determine if $S_1$ is equal to $S_2$ in $O(k(n+m))$ time, where $n$ is the maximum number of vertices in $G_1, \ldots, G_k$ and $G'_1, \ldots, G'_k$, and $m$ is the maximum number of edges in $G_1, \ldots, G_k$ and $G'_1, \ldots, G'_k$.*

*Proof.* We extend the DH-tree for a weakly distance-hereditary graph described above to the DH-tree for a set $S$ of weakly distance-hereditary graphs. The root corresponds to an empty graph, and the DH-trees of all the elements in $S$ are the children of the root. Then we can use the similar algorithm to that in[16]. □

Now we describe DECK CHECKING algorithm for distance-hereditary graphs. Given a deck $D$ that consists of weakly distance-hereditary graphs at least two of which are connected, and given a distance-hereditary preimage candidate $G = (V, E)$, we prepare the deck $\hat{D}$ of $G$ in $\mathrm{O}(|V| \cdot |E|)$ time. We can check if $D$ and $\hat{D}$ are equivalent in $\mathrm{O}(|V| \cdot |E|)$ time by Lemma 3. We thus obtain the theorem below.

**Theorem 6.** *There is $O(|V| \cdot |E|)$ time DECK CHECKING algorithm for a deck that consists of weakly distance-hereditary graphs at least two of which are connected, and for a preimage candidate $G = (V, E)$ which is a distance-hereditary graph.*

### 4.3 reducible or prunable preimage

A reducible permutation graph has twins. A distance-hereditary graph has twins or a pendant[4]. It is easy to develop a polynomial time PREIMAGE CONSTRUCTION algorithm for the deck of a graph that has twins or a pendant. If a preimage has twins, we can reconstruct it by copying every vertex in the deck and checking if the resulting graph is a preimage by DECK CHECKING algorithm. If a preimage has a pendant, we can reconstruct it by adding a degree one vertex to every vertex in the deck and checking if it is a preimage by DECK CHECKING algorithm. Thus we have the theorems below.

**Theorem 7.** *Given a deck $D = \{G_1, \ldots, G_n\}$ consisting of permutation graphs, we can list up every reducible permutation graph whose deck is $D$, if any, in $O(n^6)$ time, where $n$ is the number of graphs in $D$ (equivalently the number of vertices in a preimage).*

*Proof.* Copying every vertex in every graph in $D$ requires $\mathrm{O}(nm)$ time, where $m$ is the number of edges in a preimage, and is thus $\mathrm{O}(n^2)$. Each DECK CHECKING costs $\mathrm{O}(n^4)$ time. The maximum number of DECK CHECKING executions is $\mathrm{O}(n^2)$. Hence we need $\mathrm{O}(nm + n^4 \cdot n^2)=\mathrm{O}(n^6)$ time. □

**Theorem 8.** *Given a deck $D = \{G_1, \ldots, G_n\}$ consisting of weakly distance-hereditary graphs at least two of which are connected, we can list up every* *distance-hereditary graph whose deck is $D$, if any, in $O(n^3m)$ time, where $n$ is the number of graphs in $D$, and $m$ is the number of edges in a preimage.*

*Proof.* Copying every vertex in every graph in $D$ requires $\mathrm{O}(nm)$ time. Adding a pendant to each vertex in every graph in $D$ requires $\mathrm{O}(n^2)$ time. Each DECK CHECKING costs $\mathrm{O}(nm)$ time. The maximum number of DECK CHECKING executions is $\mathrm{O}(n^2)$. Hence we need $\mathrm{O}(nm + n^2 + nm \cdot n^2)$=$\mathrm{O}(n^3m)$ time. □

### 4.4 uniquely orientable preimage

We consider the case that a preimage of the deck is uniquely orientable in this subsection. It is possible that a graph obtained by removing a vertex from a uniquely orientable permutation graph is not uniquely orientable. Consider the graph obtained by removing the vertex 1 from the graph in Fig. 1. It seems difficult to reconstruct a uniquely orientable permutation graph owing to this fact. First we show that we can get over this difficulty.

**Lemma 4.** *There is a connected uniquely orientable permutation graph in the deck of a connected uniquely orientable permutation graph.*

*Proof.* Let $G$ be a connected uniquely orientable permutation graph. We assume that there is no connected uniquely orientable permutation graph in the deck $D$ of $G$. It is easy to see that every graph in $D$ is a permutation graph, and there exist at least two connected graphs in $D$ (considering graphs that are obtained from $G$ by removing vertices that are leaves of a spanning tree of $G$). Let $G'$ be a connected permutation graph in $D$. Since $G'$ is not uniquely orientable, $G'$ must have twins. In $G$, the pair of vertices that are twins in $G'$ are almost twins to each other but one of them has one another neighbor, since otherwise $G$ has twins and hence is not uniquely orientable. Denote such vertices by $u$ and $v$ (we assume that the degree of $u$ is one greater than that of $v$). We show that the graph $G''$ obtained by removing $v$ from $G$ is uniquely orientable. Since $G$ is uniquely orientable, for every pair of edges $e$ and $f$ in $G''$ there is an edge sequence $P = (e_1 = e, e_2, \ldots, e_k = f)$ such that $e_i$ is an edge in $G$, and $e_i$ and $e_{i+1}$ are $\Gamma$-related in $G$. Since $G''$ is not uniquely orientable, some of edges in $P$ must be incident to $v$. There are two cases.

(1) *$u$ and $v$ are weak twins.* In this case, there is an index $i \in \{2, \ldots, k-2\}$ such that $e_i$ and $e_{i+1}$ are incident to $v$. Denote $e_i$ by $\{p, v\}$ and $e_{i+1}$ by $\{v, q\}$. Since $\{p, v\}$ and $\{v, q\}$ are $\Gamma$-related, so are $\{p, u\}$ and $\{u, q\}$. Thus considering an edge sequence $(e_1, \ldots, e_{i-1}, \{p, u\}, \{u, q\}, e_{i+2}, \ldots, e_k)$, $G''$ is uniquely orientable.

(2) *$u$ and $v$ are strong twins.* If $u$ and $v$ are connected, it seems possible that some edge $e_i$ is incident to $v$ (we denote $e_i$ by $\{p, v\}$), and $e_{i-1}$ or $e_{i+1}$ is equal to $\{u, v\}$. However this is not the case, since there exists an edge $\{p, u\}$, and hence $\{p, v\}$ and $\{u, v\}$ are not $\Gamma$-related. Therefore the same argument to the case 1 gives a proof that $G''$ is uniquely orientable.

It is clear that $G''$ is a connected permutation graph in $D$. Thus the fact that $G''$ is uniquely orientable contradicts the assumption. □

Now we consider an algorithm which is given a connected uniquely orientable permutation graph $G'$ as the input and outputs all the connected uniquely orientable permutation graphs whose decks include $G'$. From the definition of permutation graph, the lemma below is clear.

**Lemma 5.** *Let $G = (V, E)$ be a permutation graph, and $G'$ be a permutation graph obtained by removing a vertex $v \in V$ from $G$. Denote a pair of vertex orderings that identifies $G$ by $P$ and $Q$. Then $P - v$ and $Q - v$ identify $G'$.*

Thus we have the lemma below.

**Lemma 6.** *Given a connected uniquely orientable permutation graph $G' = (V', E')$, we can list up all the connected uniquely orientable permutation graphs whose deck include $G'$ in $O(|V'|^2)$ time.*

*Proof.* Let $G = (V, E)$ be a connected uniquely orientable permutation graph whose deck includes $G'$. Denote by $v$ the vertex in $G$ and not in $G'$. Let $P$ and $Q$ be a pair of vertex orderings of $G$ that identifies $G$. Since $G$ is uniquely orientable, vertex orderings that identify $G$ are only $P$ and $Q$, and $\overline{P}$ and $\overline{Q}$. By lemma 5 $P - v$ and $Q - v$ identify $G'$, and $\overline{P} - v = \overline{P - v}$ and $\overline{Q} - v = \overline{Q - v}$ identify $G'$. Since $G'$ is uniquely orientable, no other pair of vertex orderings identify $G'$. Thus if $P'$ and $Q'$ are vertex orderings that identify $G'$, one of the two pairs of the vertex orderings that identify $G$ must be in the form $P' +_i v$ and $Q' +_j v$ for some $i \in \{1, \ldots, |V|\}$ and $j \in \{1, \ldots, |V|\}$. Given $G'$ we can calculate $P'$ and $Q'$ in $\mathrm{O}(|V'| + |E'|)$ time[14]. Hence we can list up $P' +_i v$ and $Q' +_j v$

in $O(|V'|+|E'|+|V|^2)$=$O(|V'|^2)$ time. Note that each output graph is in the form of edge difference from the graph previously output. Otherwise we cannot output each graph in amortized constant time. Note that the graph identified by $P' +_i v$ and $Q' +_j v$, and the graph identified by $P' +_{i+1} v$ and $Q' +_j v$ differ in one edge. □

From the lemma above, we have a polynomial time algorithm for listing all connected uniquely orientable permutation graphs whose deck is the input.

**Theorem 9.** *There is an $O(n^6)$ time algorithm for listing all connected uniquely orientable permutation graphs whose deck is the input, where $n$ is the number of vertices in a preimage.*

*Proof.* We can check if a graph $G = (V, E)$ has twins in $O(|V| + |E|)$ time[18]. Hence we can find a connected uniquely orientable permutation graph in a deck in $O(n(n + m))$ time, where $m$ is the number of edges in a preimage, and is $O(n^2)$. Once such a graph $G'$ is found, the remaining thing to do is only listing graphs whose deck include $G'$ and checking if each of them is a preimage with DECK CHECKING algorithm. Note that we only need one $G'$, not all the connected uniquely orientable permutation graphs in the deck. Thus executing DECK CHECKING at most $O(n^2)$ times is enough. □

### 4.5 disconnected preimage case

Disconnected graphs are forbidden by the definition of distance-hereditary graphs. Thus we only treat the case that the preimage $G$ is a disconnected permutation graph in this subsection.

It is clear that if there are at most one connected graph in a given deck, it is possible that the deck has disconnected preimage, and it is impossible that the deck has connected preimage. If there is at least two connected graphs in the deck, the deck has no disconnected preimage.

When $G$ is reducible, since our algorithm described in subsection 4.3 does not use the property that $G$ is connected, we simply execute our previously described algorithm.

Now we mention in the case that $G$ is irreducible. Since every connected component of $G$ must be uniquely orientable, there is a uniquely orientable graph $G''$ obtained by removing some vertex $v$ from a connected component of $G$. There must be a graph $G'$ in the deck of $G$ that has $G''$ as a connected component. Of course every connected component of $G'$ is uniquely orientable. The number of ways for inserting a vertex to a pair of vertex orderings to some connected component in $G'$ is $O(n^2)$. Thus we can use the similar algorithm to that described in subsection 4.4.

**Theorem 10.** *There is an $O(n^6)$ time algorithm for finding the disconnected permutation graph whose deck is the input, where $n$ is the number of vertices in the preimage.*

Note that if the preimage is disconnected, the graph reconstruction conjecture is true. There is thus at most one preimage in this case.

## 5. Concluding remarks

We have the main theorems by joining theorems in the previous sections. Remember that the size of the input is $O(nm)$, not $O(n + m)$. Since we can use PREIMAGE CONSTRUCTION algorithms for LEGITIMATE DECK and PREIMAGE COUNTING, we also have the LEGITIMATE DECK and PREIMAGE COUNTING algorithms running in the same time complexity for permutation graphs. In the case of distance-hereditary graphs the time complexity for LEGITIMATE DECK is the same as that for PREIMAGE CONSTRUCTION. However PREIMAGE COUNTING may cost $O(n^4m)$ time, since we have to check if we count an identical graph twice. Our argument in this paper only ensures that the number of preimages are at most $O(n^2)$. Thus the time complexity for the checking is $O(n^2 \cdot n^2 \cdot m)$, where $m$ is the time complexity for isomorphism. Note that in the case of permutation graphs the time complexity for this checking is $O(n^6)$. Also note that if the graph reconstruction conjecture is true, we do not have to execute these checkings for PREIMAGE COUNTING. These theorems do not help directly the proofs of the graph reconstruction conjecture on permutation graphs, and distance-hereditary graphs. The conjecture on these classes still remains to be open.

Note: After concluding this manuscript, we found a bug; as noted in page 3, we use a claim that $R(G)$ is uniquely orientable if $G$ is a connected permutation graph. But this claim is not correct. We will fix the bug in the representation.

## References

1) K. A. Baker, P. C. Fishburn, and F. S. Roberts: Partial orders of dimension 2. *Networks*, vol. 2 (1971) 11–28.
2) B. Bollobás: Almost every graph has reconstruction number three. *Journal of Graph Theory*, vol. 14 (1990) 1–4.
3) J. A. Bondy: A graph reconstructor's manual. *Surveys in Combinatorics, London Mathematical Society Lecture Note Series*, vol. 166 (1991) 221–252.
4) H. Bandelt, and H. M. Mulder: Distance-hereditary graphs. *Journal of Combinatorial Theory, Series B*, vol. 41 (1986) 182–208.
5) C. J. Colbourn: On testing isomorphism of permutation graphs. *Networks*, vol. 11 (1981) 13–21.
6) S. Even: *Algorithmic Combinatorics*. Macmillan, New York, 1973.
7) T. Gallai: Transitiv orientierbare graphen. *Acta Mathematica Hungarica*, vol. 18 (2005) 25–66.
8) D. L. Greenwell, and R. L. Hemminger: Reconstructing the $n$-connected components of a graph. *Aequationes Mathematicae*, vol. 9 (1973) 19–22.
9) F. Harary: A survey of the reconstruction conjecture. Graphs and Combinatorics, *Lecture Notes in Mathematics*, vol. 406 (1974) 18–28.
10) E. Hemaspaandra, L. Hemaspaandra, S. Radziszowski, and R. Tripathi: Complexity Results in Graph Reconstruction. *Discrete Applied Mathematics*, vol. 152 (2007) 103–118.
11) P. J. Kelly: A congruence theorem for trees. *Pacific Journal of Mathematics*, vol. 7 (1957) 961–968.
12) M. Kiyomi, T. Saitoh, and R. Uehara: Reconstruction of interval graphs. *COCOON 2009*, to appear.
13) D. Kratsch and L. A. Hemaspaandra: On the complexity of graph reconstruction, *Mathematical Systems Theory*, vol. 27 (1994) 257–273.
14) R. M. McConnell, and J. Spinrad: Modular decomposition and transitive orientation. *Discrete Mathematics*, vol. 201 (1999) 189–241.
15) B. D. McKay: Small graphs are reconstructible. *Australasian Journal of Combinatorics*, vol. 15 (1997) 123–126.
16) S. Nakano, R. Uehara, and T. Uno: A new approach to graph recognition and applications to distance-hereditary graphs. *Journal of computer science and technology*, vol. 24 (2009) 517–533.
17) M. von Rimscha: Reconstructibility and perfect graphs. *Discrete Mathematics*, vol. 47 (1983) 283–291.
18) J. P. Spinrad: *Efficient Graph Representations*, AMS, 2003.
19) J. Spinrad, and J. Valdes: Recognition and isomorphism of two-dimensional partial orders. *Lecture Notes in Computer Science*, vol. 154 (1983) 676–686.
20) W. T. Tutte: On dichromatic polynomials. *Journal of Combinatorial Theory*, vol. 2 (1967) 310–320.