Graph Inference from a Walk for Trees of Bounded Degree 3 is NP-Complete

Maruyama, Osamu
Research Institute of Fundamental Information Science Kyushu University

Miyano, Satoru
Research Institute of Fundamental Information Science Kyushu University

出版情報：RIFIS Technical Report. 94, 1994-09. Research Institute of Fundamental Information Science Kyushu University
バージョン：
権利関係：

Graph Inference from a Walk for Trees of Bounded Degree 3 is NP-Complete

Osamu Maruyama[†] Satoru Miyano
Research Institute of Fundamental Information Science
Kyushu University 33, Fukuoka, 812, Japan
E-mail: {maruyama,miyano}@rifis.kyushu-u.ac.jp.

Abstract

The graph inference from a walk for a class C of undirected edge-colored graphs is, given a string x of colors, finding the smallest graph G in C that allows a traverse of all edges in G whose sequence of edge-colors is x, called a walk for x. We prove that the graph inference from a walk for trees of bounded degree k is NP-complete for any $k \geq 3$, while the problem for trees without any degree bound constraint is known to be solvable in $O(n)$ time, where n is the length of the string. Furthermore, the problem for a special class of trees of bounded degree 3, called (1,1)-caterpillars, is shown NP-complete. This contrast with the case that the problem for linear chains is known to be solvable in $O(n \log n)$ time since a (1,1)-caterpillar is obtained by attaching at most one edge to each node of a linear chain. We also show the MAXSNP-hardness of these problems.

1 Introduction

A partial walk in an undirected edge-colored graph is a path in the graph. If a partial walk contains all the edges, it is called a *walk*. The trace of a partial walk w is the string of colors of the edges traversed in w. Let C be a class of graphs. The *graph inference from a walk for* C is defined as follows: Given a string x, find a graph G in C with the minimum number of edges such that G can realize a walk with trace x.

We can say that the trace of a walk in a graph provides a partial structural information of the graph. Thus the problem can be regarded as that of reconstructing an edge-colored graph only from the trace of a walk in the graph. Several kinds of such problems of reconstructing edge-colored graphs from their partial information have received attentions. Rudich [12] considered the problem of inferring a Markov chain from its output and showed that the smallest Markov chain for a

[†]Research Fellow of the Japan Society for the Promotion of Science (JSPS). This author's research is partly supported by Grants-in-Aid for JSPS research fellows from the Ministry of Education, Science and Culture, Japan.

given output can be produced in the limit. Angluin [1] and Gold [5] also considered the problem of identifying a finite automaton of the minimum size which is consistent with given input/output behaviors. They showed that the problem is, in general, NP-complete. Recently, Maruyama and Miyano [8] discussed the graph inference from partial walks, which is a variant of the graph inference from a walk, and proved that the problem is NP-complete even if the graphs are either trees or linear chains.

The graph inference problem was first deeply discussed by Aslam and Rivest [3]. They devised polynomial time algorithms for the graph inference from a walk for graphs of bounded degree 2, i.e., linear chains and cycles. Then Raghavan [11] developed a faster algorithm running in $O(n \log n)$ time. While only bounded degree graphs are considered in [3, 11], Maruyama and Miyano [8] focused on the case of unbounded degree graphs, and proved that the graph inference from a walk for trees without any degree bound constraint is solvable in $O(n)$ time. As a hardness result, Raghavan [11] considered the degree bound constraint on graphs and showed that the problem of finding a graph of bounded degree k with the minimum number of nodes that realizes a walk with trace x is NP-complete for all $k \geq 3$.

These results naturally raise a question whether the graph inference from a walk is tractable for trees with bounded degree 3. This paper settles this question. We prove that the graph inference from a walk for trees of bounded degree 3 is NP-complete even if the number of colors is restricted to 4. Generally, for any $k \geq 3$, we can show that the problem for trees of bounded degree k is NP-complete when the number of colors is $k+1$. The number $k+1$ of colors is optimal since the problem for trees of bounded degree k with at most k colors can be shown solvable in $O(n)$ time by applying the linear-time algorithm in [8].

Recall that the linear chain inference from a walk is solvable in polynomial time [3, 11]. We then consider a special class of graphs called (1,1)-caterpillars. A (1,1)-caterpillar is a graph obtained from a linear chain by attaching at most one edge, called a hair, to each node of the linear chain. Our next result asserts that the graph inference from a walk for (1,1)-caterpillars is, unfortunately, NP-complete. Namely, attaching at most one edge to each node of a linear chain makes the graph inference problem intractable.

The rest of the paper is organized as follows: Section 2 contains notations and definitions. In Section 3, we show the graph inference from a walk for trees of bounded degree 3 to be NP-complete. The NP-completeness of the problem for (1,1)-caterpillars is proven in Section 4. In Section 5, we give results on approximability of these problems.

2 Preliminaries

Let Σ be a finite alphabet. The set of all strings over Σ is denoted by Σ^*. The reversal of a string x is written as x^R, and the length of x is denoted by $|x|$. To represent the cardinality of a set S, we use the same notation $|S|$. The concatenation of strings x and y is written as $x \cdot y$, or simply xy. For strings $x_1, \ldots, x_n$, $\prod_{i=1}^{n} x_i$ denotes $x_1 x_2 \cdots x_n$. Especially, if $x = x_1 = \cdots = x_n$, we denote $\prod_{i=1}^{n} x_i$ by

$(x)^n$, or simply x^n. For convenience, x^0 is defined as the empty string ε.

A *color* is a symbol in an alphabet Σ. We consider undirected edge-colored graphs $G = (V, E, c)$, where $c : E \to \Sigma$ is called an *edge-coloring of* G. Hereafter a graph means an undirected edge-colored graph without any notice. For graphs G and G', if G and G' are isomorphic including edge labels, we identify G with G' without any notice. A *linear chain* is a graph $l = (V, E, c)$ with $V = \{v_i | i = 1, \ldots, m\}$ and $E = \{\{v_i, v_{i+1}\} | i = 1, \ldots, m-1\}$, and the *label* of l is $\prod_{i=1}^{m-1} c(\{v_i, v_{i+1}\})$.

A *partial walk* w in a graph G is a path in the graph and $w[i]$ denotes the ith node of G passed by w, where the start node of the partial walk is $w[0]$. If a partial walk contains all the edges, it is called a *walk*. We say that a partial walk w is *closed* if the start node of w coincides with the end node. Let $e_1, \ldots, e_n$ be the sequence of edges of a partial walk w in a graph $G = (V, E, c)$. The *trace* of w is $\prod_{i=1}^{n} c(e_i)$. Let x be a string in Σ^*. If w is a partial walk with trace x, w is called a *partial walk for* x. For a graph G, we say that G *realizes* a partial walk for x if there is a partial walk for x in G. Let $x = \prod_{i=1}^{n} s_i$ with s_i in Σ. For a partial walk w for x, the *subwalk* of w for substring $\prod_{i=j}^{k} s_i$ with $1 \leq j \leq k \leq n$ is the partial walk with the node sequence of $w[j-1], w[j], \cdots, w[k]$.

3 Inferring a tree of bounded degree 3 from a walk

Let C be a class of graphs. The *graph inference from a walk for* C, denoted by GIW(C), is defined as follows:

Instance: A string x over a finite alphabet Σ, and a positive integer K.

Problem: Is there a graph G in C with at most K edges such that G realizes a walk for x?

The class of trees of bounded degree k is denoted by k-Deg-Tree. At first, we consider GIW(3-Deg-Tree), the graph inference from a walk for trees of bounded degree 3.

Example. Let $K = 5$. For a string $x = abbccd$, the graph in Fig. 1 is a tree of bounded degree 3 with K edges which realizes a walk for x. On the other hand, for $y = abbccddccaad$, there is no tree of bounded degree 3 with at most K edges which realizes a walk for y.

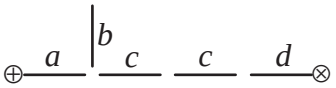


Figure 1: The nodes $\oplus$ and $\otimes$ are the start and end nodes of a walk for x, respectively.

Theorem 1 *The graph inference from a walk for trees of bounded degree 3 is NP-complete.*

The following fact is useful when we construct a tree realizing a closed walk.

Fact 1 *Suppose that a tree t realizes a closed walk w for a string x. If a symbol a occurs in x exactly twice, then t includes exactly one edge labeled a and the edge is traversed exactly twice by w in distinct directions.*

Proof of Theorem 1 Obviously, GIW(3-Deg-Tree) is in NP. To show that the problem is NP-hard, we give a reduction from the vertex cover problem (VC) [4], which is to decide if, given a graph $G = (V, E)$ and a positive integer K, there is a vertex cover of size at most K for G, that is, a subset $U \subseteq V$ with $|U| \leq K$ such that for each edge $\{u, v\} \in E$ at least one of u and v belongs to U. For an integer k, let k-DEGREE VERTEX COVER be the VC restricted to graphs of bounded degree k without any self-loop. It is known that 3-DEGREE VERTEX COVER remains NP-complete [4]. Let $K \leq |V|$ be a positive integer and $G = (V, E)$ be a graph of bounded degree 3 without any self-loop, where $V = \{v_1, \ldots, v_n\}$ and $E = \{e_1, \ldots, e_m\}$. We will define an alphabet Σ and construct a string x over Σ and a positive integer K' such that there is a tree of bounded degree 3 with at most K' edges which realizes a walk for x if and only if G has a vertex cover of size K or less. Hereafter a tree means a tree of bounded degree 3. The alphabet Σ is defined as follows:

$$\begin{aligned}\Sigma \;=\; & \{a_1, a_2\} \cup \{g_1, g_2, g_3\} \cup \{h_1, h_2\} \cup \{r_1, r_2\} \cup \{s_1, s_2\} \\ & \cup \{0, 1, \#\} \\ & \cup \{\alpha_i, \overline{\alpha}_i \mid i = 1,2,3\} \\ & \cup \{\beta_{i,j} \mid i = 1, \ldots, n \text{ and } j = 1,2,3\} \\ & \cup V.\end{aligned}$$

In order to define the string x, we introduce the four kinds of strings as follows:

(1) W and W': The string W occurs in x many times and W' once. Let $\mu = 53n + m + 2$, where $n = |V|$ and $m = |E|$. Then we define

$$\begin{aligned}W \;&=\; \textstyle\prod_{i=1}^{\mu} b_i. \\ W' \;&=\; \textstyle\prod_{i=1}^{\mu} (\#\# b_i),\end{aligned}$$

where $b_i = 0$ if i is even and $b_i = 1$ if i is odd. It is trivial that only the linear chain with label W realizes a walk for W, and the tree T_W in Fig. 2 is the smallest tree without any degree bound that realizes a walk for W'. Notice that these graphs have 2μ and μ edges, respectively. It will be seen that any tree realizing a walk for x with at most K' edges has exactly one subgraph isomorphic to T_W in Fig. 2 and all partial walks for W and W' are realized in this unique T_W.

(2) $X_{i,j}$ with $i \in \{1, \ldots, n\}$ and $j \in \{1,2,3\}$: First let

$$x_{i,j} \;=\; 0 g_1 h_1 h_2 h_2 h_1 g_2 g_2 h_1 \; \overline{\alpha}_j \overline{\alpha}_j \; h_2 g_3 g_3 h_2 \; \alpha_j \alpha_j \; h_1 g_1 \; \beta_{i,j} \beta_{i,j} \; 0,$$

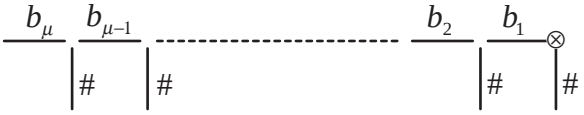
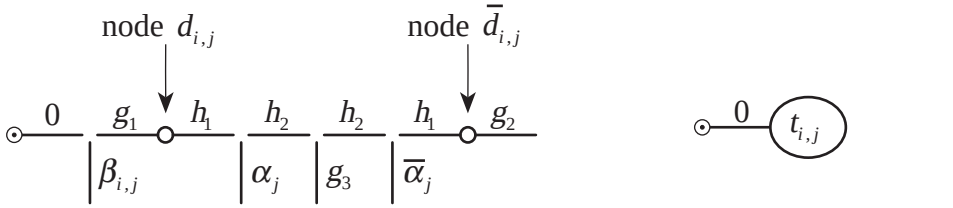


Figure 2: T_W

for $i \in \{1,\ldots,n\}$ and $j \in \{1,2,3\}$. We define $X_{i,j}$ by using $x_{i,j}$ as

$$\begin{aligned} X_{i,j} \;=\; & (a_1\#\#a_1)^i\ 1\ (r_1r_2r_2r_1)^{\lfloor (3-j)/2 \rfloor}\ (a_2\#\#a_2)^j \\ & x_{i,j}\ (s_1s_1)^{\lfloor j/3 \rfloor}\ (a_2a_2)^j\ 1\ (s_2s_2)^{\lfloor i/n \rfloor}\ (a_1a_1)^i. \end{aligned}$$

Note that $\lfloor (3-j)/2 \rfloor$ is equal to 1 if $j = 1$ and 0 otherwise. Consider the tree $t_{i,j}$ in Fig. 3. Then it is clear from Fact 1 that if a walk w for $X_{i,j}$ is closed then the subwalk of w for the substring $x_{i,j}$ of $X_{i,j}$ must be a closed walk in the tree $t_{i,j}$. A node of a tree is said to be *free* if its degree is less than 3. A prefix of x is a concatenation of the strings $X_{1,1}, X_{1,2}, X_{1,3}, X_{2,1}, \ldots, X_{n,3}$, W' and W's, which is called *TEMPLATE*. It will be seen that a tree realizing a walk for *TEMPLATE* with at most K' edges is unique up to isomorphism.



(a) $t_{i,j}$ (b) abbreviation for $t_{i,j}$

Figure 3: The tree $t_{i,j}$ has two free internal nodes, which are denoted by $d_{i,j}$ and $\overline{d}_{i,j}$. The node $\odot$ is the start and end node of the closed walk for $x_{i,j}$.

(3) $\langle v,j \rangle$ with $v \in V$ and $j \in \{1,2,3\}$: The string vv appears in $\langle v,j \rangle$ twice. We define

$$\langle v,j \rangle \;=\; (a_1a_1)^n\ 1r_1\ vv\ r_1\ (a_2a_2)^j\ 0g_1\ vv\ h_1\ \alpha_j\alpha_j\ h_1g_10\ (a_2a_2)^j\ 1\ (a_1a_1)^n.$$

The tree in Fig. 4 realizes a closed walk for $\langle v,j \rangle$ starting and ending at node $\otimes$.

(4) $\langle e \rangle$ with $e \in E$: Let $e = \{u,v\} \in E$. The strings uu and vv appear in $\langle e \rangle$ once, respectively. We define

$$\begin{aligned} \langle e \rangle \;=\; & (a_1a_1)^K\ 1\ (a_2a_2)^3\ 0g_1h_1h_2h_2h_1\ uu\ h_1h_2h_2h_1\ vv\ h_1h_2h_2h_1g_10 \\ & (a_2a_2)^3\ 1\ (a_1a_1)^K. \end{aligned}$$

Notice that $(a_1a_1)^n$ is not a prefix of $\langle e \rangle$, but $(a_1a_1)^K$ is a prefix. The tree in Fig. 5 realizes a closed walk for $\langle e \rangle$ starting and ending at node $\otimes$.

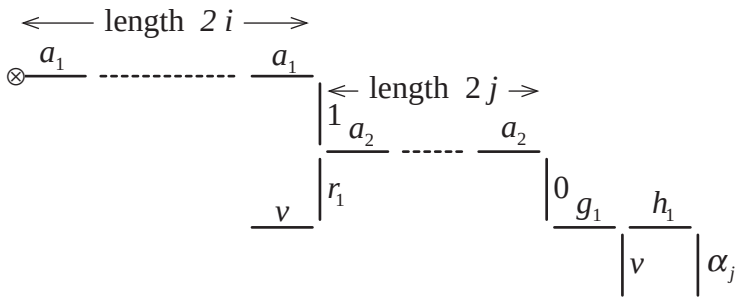


Figure 4: $i \in \{1, \ldots, n\}$.

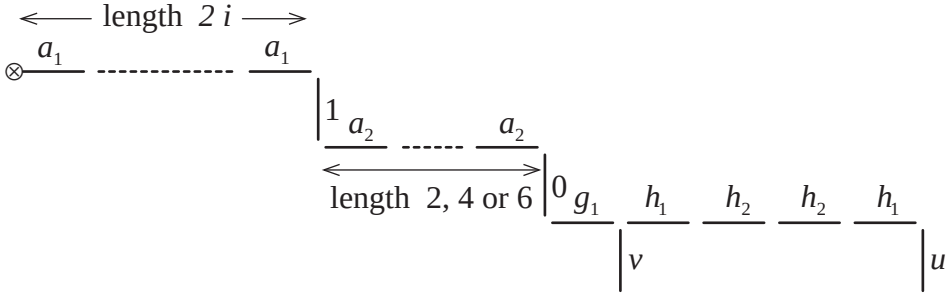


Figure 5: $i \in \{1, \ldots, K\}$.

Let

$$\begin{aligned} TEMPLATE &= W^R W' W^R \ \prod_{i=1}^{n}(\prod_{j=1}^{3}(X_{i,j} W W^R)), \\ VERTEX &= \prod_{j=1}^{3}(\prod_{i=1}^{n}(\langle v_i, j\rangle W W^R)), \\ EDGE &= \prod_{i=1}^{m}(\langle e_i\rangle W W^R). \end{aligned}$$

We then define

$$x = TEMPLATE \cdot VERTEX \cdot EDGE.$$

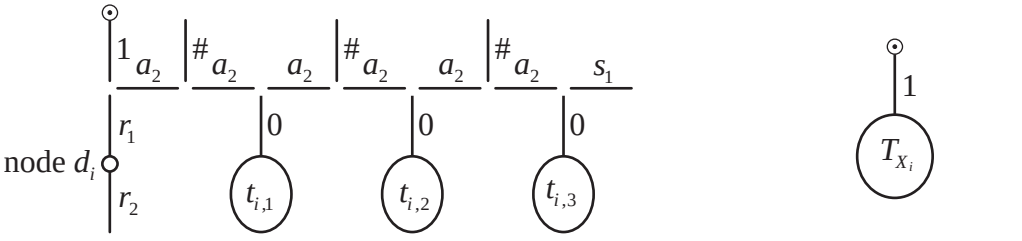
Finally, let $K' = 53n + m + 1 + 2\mu$, which is equal to $3\mu - 1$ since $\mu = 53n + m + 2$.

It is easy to see how the construction can be accomplished in polynomial time. We claim that there is a vertex cover of G with size at most K if and only if there is a tree of bounded degree 3 with at most K' edges which realizes a walk for x.

Suppose that G has a vertex cover U with $|U| \leq K$. At first, for each $i \in \{1, \ldots, n\}$, we construct the tree T_{X_i} in Fig. 6. It is obvious that for each $j \in \{1,2,3\}$, the tree T_{X_i} realizes a closed partial walk for the substring of $X_{i,j}$,

$$1\ (r_1 r_2 r_2 r_1)^{\lfloor (3-j)/2 \rfloor}\ (a_2 \# \# a_2)^j\ x_{i,j}\ (11)^{\lfloor j/3 \rfloor}\ (a_2 a_2)^j\ 1,$$

such that the closed partial walk starts and ends at the node $\odot$ in Fig. 6.



(a) T_{X_i} (b) abbreviation for T_{X_i}

Figure 6: T_{X_i} has one internal free node d_i in addition to the nodes $d_{i,j}, \overline{d}_{i,j}$ in $t_{i,j}$ for $j = 1,2,3$.

We next construct the tree T_{TEM} in Fig. 7 from all the trees $T_{X_1}, \ldots, T_{X_n}$. It is easy to check that T_{TEM} realizes a walk for the string $TEMPLATE$ with the start node $\oplus$ and the end node $\otimes$.

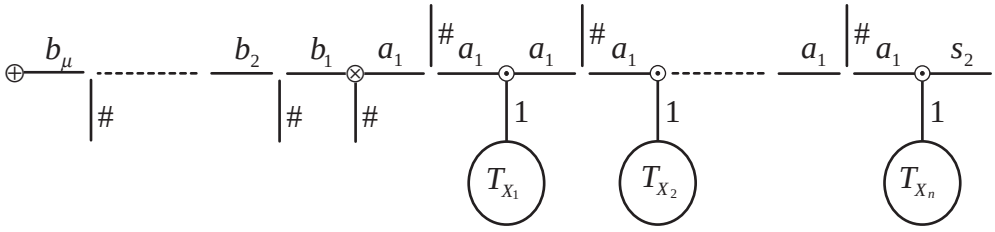


Figure 7: T_{TEM}

We here consider an assignment from V to the set of integers from 1 to n. Let b be a bijection $b : V \to \{1, \ldots, n\}$ such that $b(v) \leq K$ for all $v \in U$. For each

$v \in V$, if $b(v) = i$, four new edges labeled v are attached to T_{TEM} such that they are adjacent to the four nodes $d_i, d_{i,1}, d_{i,2}, d_{i,3}$ of T_{TEM}, respectively.

Then the resulting tree realizes a walk for $TEMPLATE \cdot VERTEX$ since for $v \in V$ and $j \in \{1,2,3\}$, a closed partial walk for $\langle v, j \rangle$ is realized in the tree such that the start and end node is $\otimes$ in Fig. 7, which is also the end node of the partial walk for $TEMPLATE$. We call the tree the *vertex-selection tree for b*.

We finally consider a partial walk for the string $EDGE$. Since U is a vertex cover, for an edge $e = \{u, v\} \in E$, at least one of u and v belongs to U. Let $f : E \to U$ be a function which for each $e \in E$, returns one endpoint of e in U. For each edge $e = \{u, v\} \in E$, if $f(e) = u$, a new edge e_{new} labeled v is attached to the vertex selection tree for b such that e_{new} is adjacent to a free node in $\{\overline{d}_{b(u),1}, \overline{d}_{b(u),2}, \overline{d}_{b(u),3}\}$. If $f(e) = v$, a new edge labeled u becomes adjacent to a free node in $\{\overline{d}_{b(v),1}, \overline{d}_{b(v),2}, \overline{d}_{b(v),3}\}$. Such a free node must exist since for any $v \in U$, $|f^{-1}(v)|$ is less than or equal to 3. For example, consider nodes $v, u_1, u_2, u_3 \in V$ and edges $\{v, u_1\}, \{v, u_2\}, \{v, u_3\} \in E$. If $b(v) = i \leq K$ and $f(\{v, u_1\}) = f(\{v, u_2\}) = f(\{v, u_3\}) = v$, then Fig. 8 becomes a subgraph of the resulting tree.

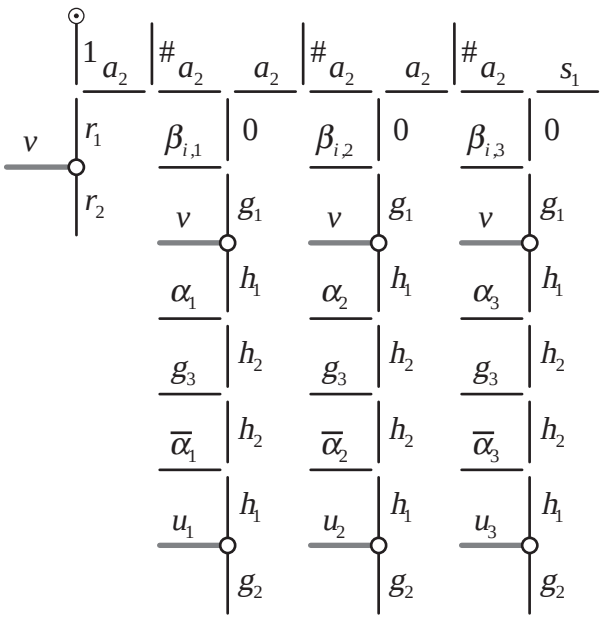


Figure 8: This tree is obtained from the tree T_{X_i} by attaching new edges to internal free nodes of T_{X_i}.

It is obvious from the construction that the resulting tree is a tree of bounded degree 3 with at most K' edges which realizes a walk for $x = TEMPLATE \cdot VERTEX \cdot EDGE$.

Conversely, suppose that there is a tree T of bounded degree 3 with at most K' edges which realizes a walk w for x. Hereafter, for a substring y of x, the subwalk for y means the subwalk of w for y without any notice, and the *induced subgraph of the subwalk for y* is the subgraph of T that is induced by all the edges

in the subwalk for y. At first, we consider the induced subgraph of the subwalk for $W^R W' W^R$, which is a prefix of x.

Claim 1 *Any tree of bounded degree 3 with at most K' edges that realizes a walk for $W^R W' W^R$ is isomorphic to the tree T_W.*

Proof Let t be a tree of bounded degree 3 with at most K' edges that realizes a walk for $W^R W' W^R$. If t contains adjacent edges labeled #, t must have at least $3\mu + 1$ edges (see Fig. 9 (a)), a contradiction, since $K' = 3\mu - 1$. Thus W' must be realized in T_W. If t is a proper supergraph of T_W, then t must be a tree with either 3μ or 4μ edges (see Fig. 9 (b)), which is a contradiction. □

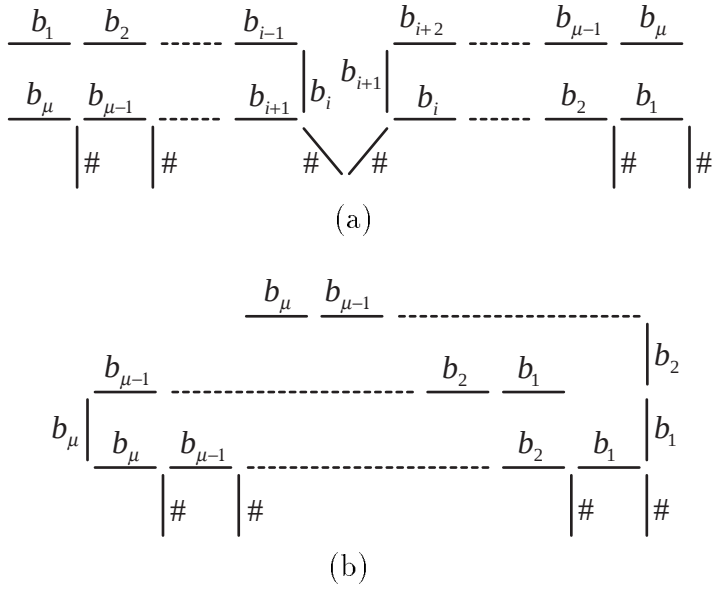


Figure 9: (a) T_W is not a subgraph of t. (b) T_W is a subgraph of t.

By Claim 1, the induced subgraph of the subwalk for $W^R W' W^R$ is isomorphic to T_W. Moreover, this subgraph isomorphic to T_W contains exactly one linear chain with label W. Let us denote this linear chain by l_W. Notice that string WW^R occurs $6n + m$ times in x. We denote the ith WW^R occurring in x by $(WW^R)_i$.

Claim 2 *For any $i \in \{1, 2, \ldots, 6n + m\}$, the induced subgraph of the subwalk for $(WW^R)_i$ coincides with the linear chain l_W.*

Proof If the induced subgraph of the subwalk for $(WW^R)_i$ and the induced subgraph of the subwalk for $W^R W' W^R$, which is isomorphic to T_W, are disjoint for some i, T must contain at least 3μ ($> K'$) edges, which is a contradiction. Therefore these induced subgraphs share nodes for any i.

Consider the substring $(WW^R)_1$. By the following three reasons, we can see that the start node of the subwalk for $(WW^R)_1$ coincides with the end node of the subwalk for $W^R W' W^R$, which corresponds to the node $\otimes$ of T_W in Fig. 2.

1. The node of T corresponding to the node $\otimes$ has one adjacent edge labeled a_1 since the next symbol of the prefix $W^R W' W^R$ of x is a_1. We denote this edge by e_{a_1}.

2. It is easy to check that the subwalk for the substring $X_{1,1}$ of x, which occurs between the prefix $W^R W' W^R$ and the substring $(WW^R)_1$, does not have any common edge with the induced subgraph of the subwalk for $W^R W' W^R$.

3. Notice that the last symbol of $X_{1,1}$ is also a_1. If the subwalk for the symbol a_1 does not coincide with the edge e_{a_1}, the subwalk for $(WW^R)_1$ does not have any common nodes with the induced subgraph of the subwalk for $W^R W' W^R$, which is a contradiction.

If the induced subgraph of the subwalk for $(WW^R)_1$ does not coincide with the linear chain l_W, T must have a node with degree exceeding 3, a contradiction. Thus the claim holds for $i = 1$. In a similar way, we can see that the claim holds in a consecutive manner for $i = 2, 3, \ldots, 6n + m$. □

Notice that for each $i \in \{1, 2, \ldots, 6n + m\}$, the subwalks for $(WW^R)_i$ is a closed partial walk starting and ending at the node corresponding to the node $\otimes$ in Fig. 2. Let

$$S = \{X_{i,j}, \langle v_i, j\rangle \mid i = 1, \ldots, n \text{ and } j = 1, 2, 3\} \cup \{\langle e_i\rangle \mid i = 1, \ldots, m\},$$

which consists of $X_{1,1}$ and the strings Z_i such that Z_i occurs in x as $x = \cdots (WW^R)_{i-1} Z_i (WW^R)_i \cdots$ for $i = 2, 3, \ldots, 6n + m$. The following is clear from Claim 2:

Claim 3 *For each string Z in S, the subwalk for Z must be closed.*

We next consider the prefix $W^R W' W^R X_{1,1} W W^R$ of x. By carefully folding the prefix $W^R W' W^R X_{1,1} W W^R$, we can see the following claim:

Claim 4 *Let $T_{X_{1,1}}$ be the tree in Fig. 10. Any tree of bounded degree 3 with at most K' edges that realizes a walk for $W^R W' W^R X_{1,1} W W^R$ is isomorphic to the tree $T_{X_{1,1}}$.*

Consider the substring $X_{1,2}$ appearing after the prefix $W^R W' W^R X_{1,1} W W^R$. For convenience, we denote the prefix $W^R W' W^R \prod_{j=1}^{2} (X_{1,j} W W^R)$ of x by $X_{1,(1,2)}$. See Fig. 11. The nonshaded part of the tree is isomorphic to $T_{X_{1,1}}$. For each $i \in \{1, 2, 3, 4\}$, let $t^{(i)}$ be the tree consisting of the shaded part $p^{(i)}$ and the nonshaded part. Then it is not hard to see that $t^{(1)}$, $t^{(2)}$, $t^{(3)}$, $t^{(4)}$ are the only trees of bounded degree 3 with at most K' edges which can realize a walk for $X_{1,(1,2)}$. By considering the substrings $X_{1,3}$, $X_{n,1}$ that appear after $X_{1,(1,2)}$ in x, we will see that the possible form of the induced subgraph of the subwalk for $X_{1,(1,2)}$ is only $t^{(4)}$.

If the induced subgraph of the subwalk for $X_{1,(1,2)}$ is either $t^{(1)}$ or $t^{(2)}$, then we can see that there is no possibility to realize any closed partial walk for the substring $X_{n,1}$ in T starting and ending at the node corresponding to the node $\otimes$

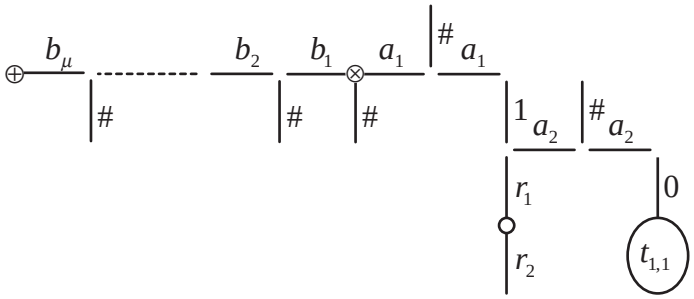


Figure 10: $T_{X_{1,1}}$

in Fig. 11. Next consider the case that the induced subgraph of the subwalk for $X_{1,(1,2)}$ is $t^{(3)}$. Let us denote a tree realizing a closed walk for the string

$$1\ (r_1r_2r_2r_1)^{\lfloor(3-j)/2\rfloor}\ (a_2\#\#a_2)^j\ x_{i,j}\ (s_1s_1)^{\lfloor j/3\rfloor}\ (a_2a_2)^j\ 1$$

by $t'_{1,3}$, which is a substring of $X_{1,3}$. We consider the subwalk for $X_{1,3}$ in T. In this case, the induced subgraph of the subwalk for the prefix $X_{1,(1,2)}\ X_{1,3}$ of x must be isomorphic to a tree obtained by attaching $t'_{1,3}$ to $t^{(3)}$ such that the leaf of the edge labeled 1 of $t'_{1,3}$ is identified with the node $\oslash$ of $t^{(3)}$. In the same way as the cases of $t^{(1)}$ and $t^{(2)}$, this implies no possibility to realize any closed partial walk for the substring $X_{n,1}$ in T.

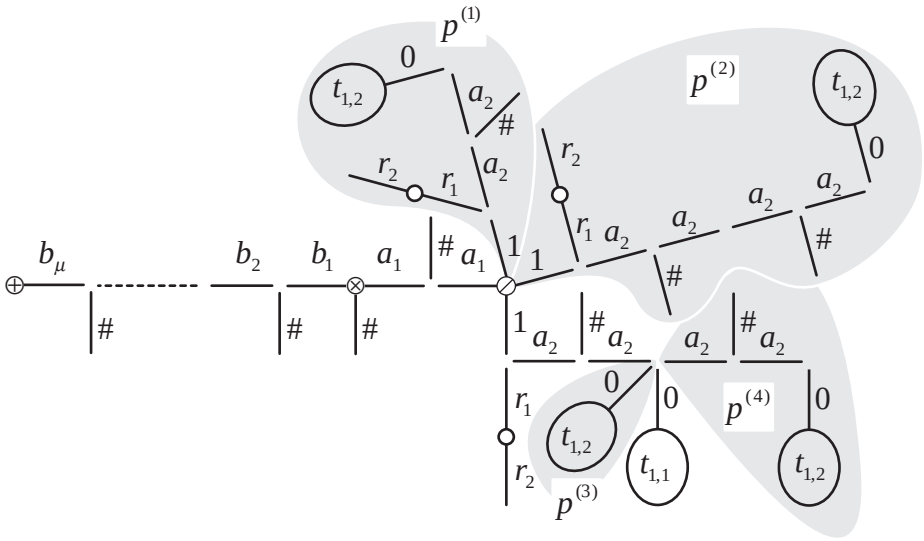


Figure 11: The four shaded parts are denoted by $p^{(1)}, p^{(2)}, p^{(3)}$ and $p^{(4)}$, respectively.

For convenience, we denote by X_1 the prefix $W^RW'W^R\ \prod_{j=1}^{3}(X_{1,j}WW^R)$. Let $T_{X_{1,(1,2,3)}}$ be the tree in Fig. 12. In a way similar to the above argument, it can be seen that any tree of bounded degree 3 with at most K' edges that realizes a walk for X_1 is isomorphic to the tree $T_{X_{1,(1,2,3)}}$.

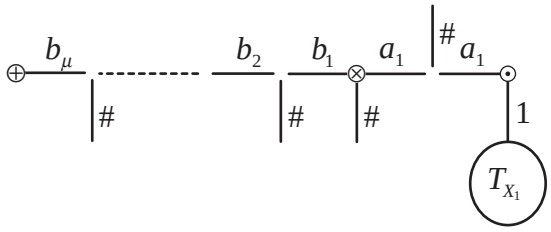


Figure 12: $T_{X_{1,(1,2,3)}}$.

We next consider the substring $X_{2,1}$ appearing after the prefix X_1. There are two possibilities to realize a walk for $X_1\ X_{2,1}WW^R$ by a tree of bounded degree 3 with at most K' edges. See the tree in Fig. 13. The nonshaded part of the tree is isomorphic to $T_{X_{1,(1,2,3)}}$. One is the tree consisting of the shaded part $\tilde{p}^{(1)}$ and the nonshaded part. The other is the tree consisting of the shaded part $\tilde{p}^{(2)}$ and the nonshaded part. It is clear that if the former is a subgraph of T then T cannot realize any closed partial walk for $X_{n,1}$. Therefore any tree of bounded degree 3 with at most K' edges that realizes a walk for $X_1\ X_{2,1}WW^R$ must be isomorphic to the latter.

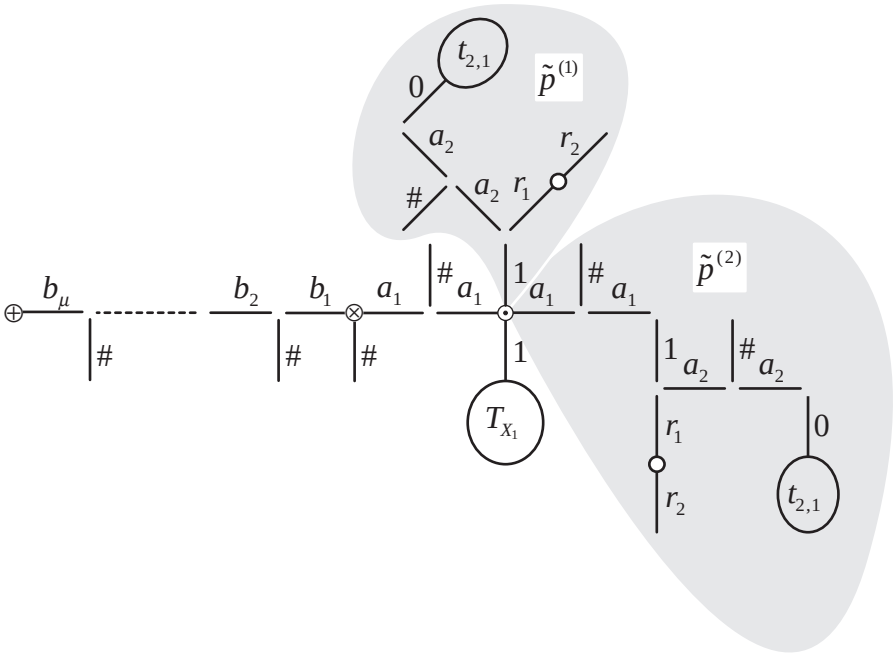


Figure 13: The two shaded parts are denoted by $\tilde{p}^{(1)}$ and $\tilde{p}^{(2)}$, respectively.

By continuing this argument for $X_{2,2}, X_{2,3}, X_{3,1}, \ldots, X_{n,3}$, we can obtain the following:

Claim 5 *Any tree of bounded degree 3 with at most K' edges that realizes a walk for TEMPLATE* $= W^R W' W^R \prod_{i=1}^{n} (\prod_{j=1}^{3} (X_{i,j} W W^R))$ *is isomorphic to the tree* T_{TEM}.

Consider the substring $\langle v, 1\rangle$ for $v \in V$. By Claim 3, the subwalk for $\langle v, 1\rangle$ is closed. Then the start and end node of the subwalk for $\langle v, 1\rangle$ coincides with the node corresponding to the node $\otimes$ in Fig. 7. We next consider the subwalk for the prefix $(a_1 a_1)^n$ of $\langle v, 1\rangle$. By Claim 5, the induced subgraph of the subwalk for *TEMPLATE* is isomorphic to T_{TEM}. Moreover, this subgraph contains exactly one linear chain with label $(a_1 a_1)^n$. Let us denote this linear chain by $l_{(a_1 a_1)^n}$. It can be easily seen that the induced subgraph of the subwalk for the prefix $(a_1 a_1)^n$ of $\langle v, 1\rangle$ is a subgraph of $l_{(a_1 a_1)^n}$. Notice that the subwalk for the substring

$$1\ (r_1 r_2 r_2 r_1)^{\lfloor (3-j)/2 \rfloor}\ (a_2 \# \# a_2)^j\ x_{i,j}\ (s_1 s_1)^{\lfloor j/3 \rfloor}\ (a_2 a_2)^j\ 1$$

of $X_{i,j}$, denoted by $sub_{X_{i,j}}$, is closed and the induced subgraph of the subwalks for $sub_{X_{i,1}}$, $sub_{X_{i,2}}$ and $sub_{X_{i,3}}$ is isomorphic to T_{X_i}. The start and end nodes of these subwalks coincide and correspond to the node $\odot$ of T_{X_i} in Fig. 7. We denote this induced subgraph isomorphic to T_{X_i} by T_i and the node of T_i corresponding to the node $\odot$ of T_{X_i} by $\odot_i$. The end node of the subwalk for the prefix $(a_1 a_1)^n$ of $\langle v, 1\rangle$ must coincide with $\odot_i$ for some $i \in \{1, \ldots, n\}$, but does not coincide with the node corresponding to the node $\otimes$ in Fig 7 since the next symbol occurring after the prefix $(a_1 a_1)^n$ in $\langle v, 1\rangle$ is 1 and $\otimes$ does not have any adjacent edges labeled 1. We say that the subwalk for $\langle v, 1\rangle$ *selects* T_i if the end node of the subwalk for the prefix $(a_1 a_1)^n$ of $\langle v, 1\rangle$ coincides with $\odot_i$.

Suppose that the subwalk for $\langle v, 1\rangle$ selects T_i. Consider the substring

$$1 r_1\ vv\ r_1 a_2 a_2 0 g_1\ vv\ h_1\ \alpha_1 \alpha_1\ h_1 g_1 0 a_2 a_2 1,$$

denoted by $sub_{\langle v,1\rangle}$, of $\langle v, 1\rangle$ appearing after the prefix $(a_1 a_1)^n$. By Fact 1, the subwalk for $sub_{\langle v,1\rangle}$ must be closed, and then it is trivial that any tree of bounded degree 3 realizing a walk for $sub_{\langle v,1\rangle}$ is the tree in Fig. 14. This implies that the

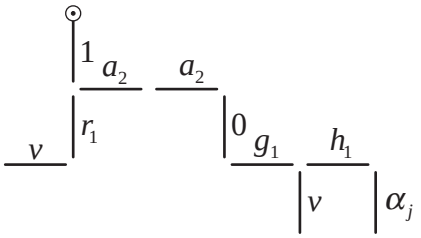


Figure 14: The node $\odot$ is the start and end node of a walk for $sub_{\langle v,1\rangle}$.

induced subgraph of the subwalks for *TEMPLATE* and $\langle v, 1\rangle$ is isomorphic to the tree obtained from T_{TEM} by attaching two new edges labeled v to the nodes d_i and $d_{i,1}$, respectively. Note that the remainder $(a_1 a_1)^n$ of $\langle v, 1\rangle$ finishes the subwalk for $\langle v, 1\rangle$ in T.

We should notice that if the subwalk for $\langle v,1\rangle$ has selected T_i then no other subwalk for $\langle v',1\rangle$ with $v' \neq v$ can select T_i since node d_i is not free and any of the labels of the three adjacent edges of d_i is not v'.

For an arbitrary bijection $b : V \to \{1,\ldots,n\}$, let $T_{TEM,b}$ be the tree obtained from T_{TEM} by attaching two new edges labeled v to $d_{b(v)}$ and $d_{b(v),1}$ of T_{TEM} for all $v \in V$. It is easy to see that the following holds:

Claim 6 *For a tree T', T' is a tree of bounded degree 3 with at most K' edges that realizes a walk for $TEMPLATE \cdot \prod_{i=1}^{n}(\langle v_i,1\rangle WW^R)$ if and only if T' is isomorphic to $T_{TEM,b}$ for some bijection $b : V \to \{1,\ldots,n\}$.*

Consider the substring $\langle v,2\rangle$. In the same way as the subwalk for $\langle v,1\rangle$, we also say that the subwalk for $\langle v,2\rangle$ *selects* T_i if the end node of the subwalk for the prefix $(a_1a_1)^n$ of $\langle v,2\rangle$ coincides with $\odot_i$. The string vv occurs in $\langle v,2\rangle$ twice as a substring. If the subwalk for $\langle v,1\rangle$ selects T_i then the subwalk for $\langle v,2\rangle$ must also select T_i, since the subwalk for the first vv coincides with the edge labeled v adjacent to the node corresponding to d_i. Moreover, the subwalk for the second occurrence of vv must be the edge labeled v adjacent to the node corresponding to $d_{b(v),2}$.

By a similar argument on $\langle v,3\rangle$, the following claim holds:

Claim 7 *For a tree T', T' is a tree of bounded degree 3 with at most K' edges realizing a walk for $TEMPLATE \cdot VERTEX$ if and only if T' is isomorphic to the vertex-selection tree for some bijection $b : V \to \{1,\ldots,n\}$.*

Fix a vertex-selection tree for a bijection $b : V \to \{1,\ldots,n\}$. We consider the substring $\langle e\rangle$ for $e = \{u,v\} \in E$. In the same way as the subwalks for $\langle v,1\rangle$ and $\langle v,2\rangle$, we also say that the subwalk for $\langle e\rangle$ *selects* T_i if the end node of the subwalk for the prefix $(a_1a_1)^K$ of $\langle e\rangle$ coincides with $\odot_i$. It is clear that the subwalk for $\langle e\rangle$ must select T_i with $i \leq K$.

Suppose that the subwalk for $\langle e\rangle$ selects T_i for some $i \in \{1,\ldots,K\}$. Consider the substring $(a_2a_2)^3$ of $\langle e\rangle$ appearing after the prefix $(a_1a_1)^K1$. We denote the induced subgraph of the subwalk for $x_{i,j}$ by $t'_{i,j}$, which is isomorphic to $t_{i,j}$, and denote the start and end node of the subwalk for $x_{i,j}$ by $\odot_{i,j}$, which corresponds to the node $\odot$ of $t_{i,j}$ in Fig. 3. If the end node of the subwalk for $(a_1a_1)^K1(a_2a_2)^3$ coincides with the node $\odot_{i,j}$, we say that the subwalk for $\langle e\rangle$ *selects* $t'_{i,j}$.

Suppose that the subwalk for $\langle e\rangle$ selects $t'_{i,j}$. We denote the substring

$$0g_1h_1h_2h_2h_1\ uu\ h_1h_2h_2h_1\ vv\ h_1h_2h_2h_1g_10$$

of $\langle e\rangle$ by $sub_{\langle e\rangle}$, which follows after the prefix $(a_1a_1)^K1(a_2a_2)^3$. By Fact 1, the subwalk for $sub_{\langle e\rangle}$ must be closed. Note that $b^{-1}(i)$ is either u or v. If not, we can easily see that T must have a node with degree exceeding 3, which is a contradiction. If $b^{-1}(i) = v$, the subwalk for $sub_{\langle e\rangle}$ must include the edge labeled u being adjacent to the node corresponding to $\overline{d}_{i,j}$ in the vertex-selection tree for b (see Fig. 15). If $b^{-1}(i) = u$, the subwalk for $sub_{\langle e\rangle}$ must include the edge labeled v being adjacent to the node corresponding to $\overline{d}_{i,j}$ in the vertex-selection tree for b (see Fig. 16).

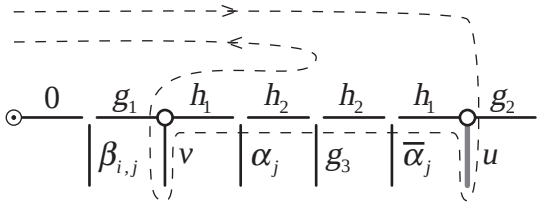


Figure 15: The broken line is the subwalk for $sub_{\langle e \rangle}$.

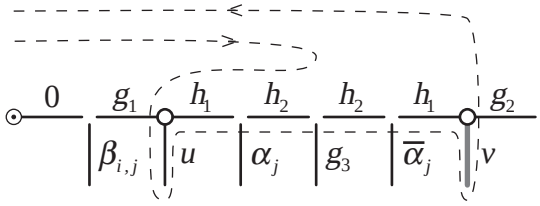


Figure 16: The broken line is the subwalk for $sub_{\langle e \rangle}$.

The suffix $(a_2a_2)^3 1 (a_1a_1)^K$ of $\langle e \rangle$ finishes the subwalk for $\langle e \rangle$ in T. Notice that if the subwalk for $\langle e \rangle$ has selected $t'_{i,j}$ then no other subwalk for $\langle e' \rangle$ can select $t'_{i,j}$.

Let $U = \{b^{-1}(i) \mid i \in \{1, \ldots, K\}\}$. It should be clear from the form of T that U is a vertex cover of G with size K.

□

Theorem 2 *The graph inference from a walk for trees of bounded degree* 3 *is NP-complete even if the alphabet size is restricted to* 4*.*

If a string x is long and written over a few symbols, Fact 1 does not seem to be helpful to determine the tree realizing a closed walk for the string x. The following fact is useful in such a case:

Fact 2 *Suppose that t is an arbitrary tree without any degree bound constraint realizing a closed walk for a string x and t' is the smallest tree without any degree bound constraint realizing a walk for x. Let w and w' be those walks realized by t and t', respectively.*

1. *For an edge e of t' traversed in w' exactly twice, let $e = \{w'[i], w'[i+1]\} = \{w'[j], w'[j+1]\}$ with $0 \leq i < j \leq |x|-1$. Then, $\{w[i], w[i+1]\} = \{w[j], w[j+1]\}$.*

2. *For an edge e of t' traversed in w' exactly four times, let*

$$\begin{aligned} e &= \{w'[i_1], w'[i_1+1]\} \\ &= \{w'[i_2], w'[i_2+1]\} \\ &= \{w'[i_3], w'[i_3+1]\} \\ &= \{w'[i_4], w'[i_4+1]\} \end{aligned}$$

with $0 \leq i_1 < i_2 < i_3 < i_4 \leq |x| - 1$. *Then, one of the following three cases holds:*

Case 1: $\{w[i_1], w[i_1+1]\} = \{w[i_2], w[i_2+1]\}$ $(= e_0)$,
$\{w[i_3], w[i_3+1]\} = \{w[i_4], w[i_4+1]\}$ $(= e_1)$
and $e_0 \neq e_1$.

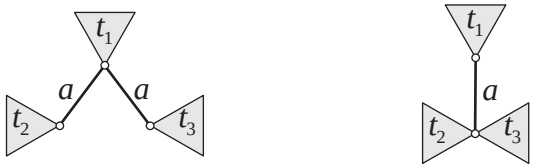
Case 2: $\{w[i_1], w[i_1+1]\} = \{w[i_4], w[i_4+1]\}$ $(= e_0)$,
$\{w[i_2], w[i_2+1]\} = \{w[i_3], w[i_3+1]\}$ $(= e_1)$
and $e_0 \neq e_1$.

Case 3: $\{w[i_1], w[i_1+1]\} = \{w[i_2], w[i_2+1]\} =$
$\{w[i_3], w[i_3+1]\} = \{w[i_4], w[i_4+1]\}$.

Fact 2 can be proven by Lemma 1 (4) and the fact that the shortest path between two nodes in a tree is unique.

Definition 1 *1. Let* $\rightarrow$ *be a binary relation on a set* S *and* $\stackrel{*}{\rightarrow}$ *be the transitive and reflexive closure of* $\rightarrow$. *An element* $y \in S$ *is irreducible if there is no* $y \in S$ *such that* $x \rightarrow y$. *For* $x, y \in S$, *if* $x \stackrel{*}{\rightarrow} y$ *and* y *is irreducible, then* y *is called a* $\rightarrow$*-normal form of* x.

2. Let T_1 *be a tree which includes adjacent edges* $e_1 = \{v_1, v_2\}$ *and* $e_2 = \{v_2, v_3\}$ *with* $c(e_1) = c(e_2)$ *(see Fig. 17 (a)). Let* T_2 *be the tree obtained from* T_1 *by identifying* v_3 *with* v_1 *together with the adjacent edges* $\{v_1, v_2\}$ *and* $\{v_2, v_3\}$ *(see Fig. 17 (b)). Then we say that* T_2 *is an edge-folding of* T_1. *The binary relation* $\rightarrow_F$ *on the set of trees is defined to be the set of pairs* (T_1, T_2) *such that* T_2 *is an edge-folding of* T_1.



(a) T_1 (b) T_2

Figure 17: t_1, t_2 and t_3 in (a) and (b) are arbitrary trees and a is an arbitrary color.

Lemma 1 (1) *A tree* T *realizes a walk for a string* x *if and only if* $l \stackrel{*}{\rightarrow}_F T$, *where* l *is a linear chain with label* x.

(2) *The binary relation* $\rightarrow_F$ *on the set of trees is confluent, namely, for any trees* x, y_0, y_1, $x \stackrel{*}{\rightarrow}_F y_0$ *and* $x \stackrel{*}{\rightarrow}_F y_1$ *imply that there is a tree* z *such that* $y_0 \stackrel{*}{\rightarrow}_X z$ *and* $y_1 \stackrel{*}{\rightarrow}_X z$.

(3) *Let* x *be a string and* l *be a linear chain with label* x. *Then, a* $\rightarrow_F$*-normal form of* l *is unique and it is the smallest tree that realizes a walk for* x.

(4) *Suppose that t is an arbitrary tree realizing a walk for a string x and t' is the smallest tree realizing a walk for x. We denote those walks by w and w', respectively. Then, for any integers $0 \leq i < j \leq |x|-1$, if $\{w'[i], w'[i+1]\} \neq \{w'[j], w'[j+1]\}$, then $\{w[i], w[i+1]\} \neq \{w[j], w[j+1]\}$.*

Proof (1) We can show the necessary condition by induction on the length of x. The converse is trivial.
(2) It is sufficient to show that $\rightarrow_F$ is locally confluent [7, Lemma 2.4] or [3, Theorem 1], which can be easily seen.
(3) It can be shown from (1) and (2) in the same way as the proof of Theorem 4 in [3].
(4) It is trivial from (1) and (3). □

Proof of Theorem 2 We give a reduction to show the NP-hardness by coding the string x defined in the previous proof into a string x' over the alphabet $\Sigma' = \{0, 1, \#, \$\}$.

We first make a string over Σ', which is denoted by $code(i,j)$. For integers $j, k \geq 0$, the notation j_k denotes the kth bit of the binary representation of j, i.e., $j_0 + j_1 \cdot 2 + j_2 \cdot 2^2 + \cdots + j_m \cdot 2^m$ for some integer $m \geq \lfloor \log j \rfloor$. For integers $i \geq j \geq 1$, the string $code(i,j)$ is defined as follows:

$$code(i,j) = \begin{cases} j_0 j_0 & \text{if } i = 1 \\ j_0 \cdot \prod_{k=1}^{\lfloor \log i \rfloor} (\#\#\$\#\#j_k) \cdot \prod_{k=1}^{\lfloor \log i \rfloor} (j_{\lfloor \log i \rfloor - k + 1}\$) \cdot j_0 & \text{if } i \geq 2 \end{cases}$$

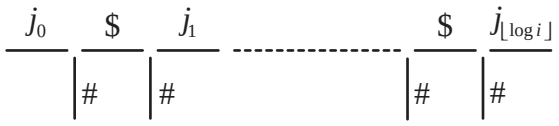


Figure 18: The tree of bounded degree 3 that realizes a closed walk for $code(i,j)$.

Recall that the alphabet Σ defined in the proof of Theorem 1 is

$$\begin{aligned} \Sigma \;=\; & \{a_1, a_2\} \cup \{g_1, g_2, g_3\} \cup \{h_1, h_2\} \cup \{r_1, r_2\} \cup \{s_1, s_2\} \\ & \cup \{0, 1, \#\} \\ & \cup \{\alpha_i, \overline{\alpha}_i \mid i = 1,2,3\} \\ & \cup \{\beta_{i,j} \mid i = 1, \ldots, n \text{ and } j = 1,2,3\} \\ & \cup V. \end{aligned}$$

The string x' is obtained from x by replacing the symbols and substrings of x which are listed in the left side of Table 1 with those in the right side.

Thus the substrings of x which are $x_{i,j}, X_{i,j}, \langle v_i, j \rangle, \langle e \rangle, W$ and W' are transformed as follows:

(1) For $i \in \{1, 2, \ldots, n\}$ and $j \in \{1,2,3\}$,

$$\begin{aligned} x_{i,j} \;=\; & 0\#\$\#\#\$\#\#\$ \; code(1, (j+1) mod 2) \; \#00\# \; code(3,j) \; \$\# \\ & (\$\#\# \; code(3n, 3(i-1)+j) \; \$) \; 0. \\ X_{i,j} \;=\; & (\$\#\#\$)^i \; 1 \; (\#\$\$\#)^{\lfloor (3-j)/2 \rfloor} \; (\$\#\#\$)^j \; x_{i,j} \; (11)^{\lfloor j/3 \rfloor} \; (\$\$)^j \; 1 \\ & (00)^{\lfloor i/n \rfloor} (\$\$)^i. \end{aligned}$$

a_1	\$
a_2	\$
g_1	#
g_2	#
g_3	0
h_1	\$
h_2	#
r_1	#
r_2	\$
s_1	1
s_2	0
0	0
1	1
#	#
$\alpha_i\alpha_i$	$code(3,i)$
$\overline{\alpha}_i\overline{\alpha}_i$	$code(1,(i+1)mod2)$
$\beta_{i,j}\beta_{i,j}$	\$## $code(3n,3(i-1)+j)$ \$
$v_iv_i(v_i \in V)$	$code(n,i)$.

Table 1:

(2) For $v_i \in V$,

$$\begin{aligned}\langle v_i, j\rangle \;=\; & (\$\$)^n\ 1\#\ code(n,i)\ \#\ (\$\$)^j\ 0\#\ code(n,i)\ \$\ code(3,j)\ \$\#0\\ & (\$\$)^j\ 1\ (\$\$)^n.\end{aligned}$$

(3) For $e = \{v_i, v_j\} \in E$,

$$\begin{aligned}\langle e\rangle \;=\; & (\$\$)^K\ 1\ (\$\$)^3\ 0\#\$\#\#\$\ code(n,i)\ \$\#\#\$\ code(n,j)\ \$\#\#\$\#0\\ & (\$\$)^3\ 1\ (\$\$)^K.\end{aligned}$$

(4) $W = \prod_{i=1}^{\mu'} b_i$ and $W' = \prod_{i=1}^{\mu'}(\#\#b_i)$, where

$$\mu' = 58n + 2 + 3n(4\lfloor \log 3n\rfloor + 1) + (4n+m)(4\lfloor \log n\rfloor + 1).$$

and b_i is defined in the former proof.

Then x' is also written as follows:

$$x' = TEMPLATE \cdot VERTEX \cdot EDGE$$

where

$$\begin{aligned} TEMPLATE &= W^R W' W^R \ \prod_{i=1}^{n}(\prod_{j=1}^{3}(X_{i,j} W W^R)),\\ VERTEX &= \prod_{j=1}^{3}(\prod_{i=1}^{n}(\langle v_i, j\rangle W W^R)),\\ EDGE &= \prod_{i=1}^{m}(\langle e_i\rangle W W^R). \end{aligned}$$

Let

$$K'' = 58n + 3n(4\lfloor \log 3n \rfloor + 1) + (4n + m)(4\lfloor \log n \rfloor + 1) + 1 + 2\mu'.$$

This transformation can be also done in polynomial time. We claim that there is a vertex cover of G with size at most K if and only if there is a tree of bounded degree 3 with at most K'' edges which realizes a walk for x'. We leave the verification of the claim to the reader. □

By a slight modification of the reduction in the proof of Theorem 2, we can obtain the following:

Theorem 3 *For any integer $k \geq 3$, the graph inference from a walk for trees of bounded degree k is NP-complete even if the alphabet size is restricted to $k + 1$.*

On the other hand, if the number of symbols in a string x is at most k, the linear-time algorithm in [8] produces the smallest tree of bounded degree k that realizes a walk for x. This is because for any string x of k colors, the smallest tree realizing a walk for x has no node with degree exceeding k.

4 Inferring a (1,1)-caterpillar from a walk

A caterpillar is a tree which is created by a linear chain, called the *backbone*, and various other appendage linear chains attached to the nodes of the backbone, called *hairs* [6]. For integers k and $l \geq 0$, a caterpillar is called a (k, l)-caterpillar if the number of hairs of a node in the backbone is at most k and the maximum length of hairs is at most l. We can say that (1,1)-caterpillars are the simplest trees of bounded degree 3 since at most one edge is attached to each node of a linear chain. The class of (1,1)-caterpillars is denoted by (1,1)-Caterpillar. We next consider the graph inference from a walk for (1,1)-caterpillars, GIW((1,1)-Caterpillar). Recall that the problem for linear chains is solvable in polynomial time [3, 11]. However, if each node of a linear chain is allowed to append at most one edge to itself, the graph inference problem turns to be NP-complete.

Theorem 4 *The graph inference from a walk for (1,1)-caterpillars is NP-complete.*

Proof We also give a reduction from 3-DEGREE VERTEX COVER. Let $K \leq |V|$ be a positive integer and $G = (V, E)$ be a graph of bounded degree 3 without any self-loop, where $V = \{v_1, \ldots, v_n\}$ and $E = \{e_1, \ldots, e_m\}$. We must construct a string x over an alphabet Σ and a positive integer K' such that there is a $(1, 1)$-caterpillar T with at most K' edges which realizes a walk for x if and only if G has a vertex cover of size K or less.

The basic idea of the reduction is the same as the reduction in the proof of Theorem 1. For example, we will define a string W, which is similar to the string W defined in the proof of Theorem 1, so that if a string y occurs in x as $x = \cdots W^R y W \cdots$ then the subwalk for y must be closed.

Let

$$\begin{aligned}\Sigma &= \{a_1, a_2, a_3\} \\ &\cup \{0, 1\} \\ &\cup \{g\} \\ &\cup \{\alpha_1, \alpha_2, \alpha_3\} \\ &\cup \{\beta_{i,j} \mid i = 1, \ldots, n \text{ and } j = 1, \ldots, 12\}.\end{aligned}$$

We define $X_i, \langle v \rangle, \langle e \rangle$ and W as follows:

(1) For $i \in \{1, 2, \ldots, n\}$,

$$\begin{aligned}X_i &= g\ (\beta_{i,1})^2\ g\ (\beta_{i,2})^2\ gg\ (\beta_{i,3})^2 \\ &\quad \textstyle\prod_{j=1}^{3}(a_1\ \alpha_j\alpha_j\ a_2a_3\ (\beta_{i,3j+1})^2\ a_3a_2\ (\beta_{i,3j+2})^2\ a_1\ (\beta_{i,3j+3})^2).\end{aligned}$$

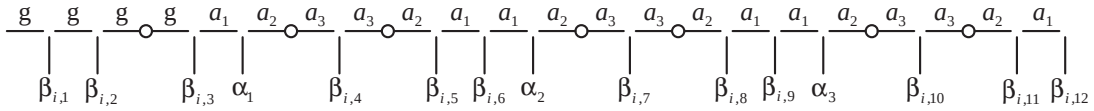


Figure 19: A (1,1)-caterpillar which realizes a walk for X_i.

(2) For $v \in V$,

$$\begin{aligned}\langle v \rangle &= (g^4\ (a_1a_2a_3a_3a_2a_1)^3\ )^{n-1}\ g^4g\ vv\ g\ \textstyle\prod_{j=1}^{3}(a_1\ \alpha_j\alpha_j\ a_2\ vv\ a_3a_3a_2a_1) \\ &\quad ((a_1a_2a_3a_3a_2a_1)^3\ g^4\ )^n.\end{aligned}$$

(3) For $e = \{u, v\} \in E$,

$$\begin{aligned}\langle e \rangle &= (g^4\ (a_1a_2a_3a_3a_2a_1)^3\ )^K\ (a_1a_2a_3a_3a_2a_1)^2 \\ &\quad a_1a_2a_3a_3\ uu\ a_3a_3\ vv\ a_3a_3a_2a_1 \\ &\quad (a_1a_2a_3a_3a_2a_1)^2\ ((a_1a_2a_3a_3a_2a_1)^3\ g^4\ )^K.\end{aligned}$$

(4) $W = \prod_{i=1}^{\mu} b_i$, where $\mu = 41n + m + 1$ and b_i is the same as that in the proof of Theorem 1.

We then define

$$x = \mathit{TEMPLATE} \cdot \mathit{VERTEX} \cdot \mathit{EDGE}$$

where

$$\begin{aligned}\mathit{TEMPLATE} &= W^R\ \textstyle\prod_{i=1}^{n} X_i\ ((a_1a_2a_3a_3a_2a_1)^3g^4)^n\ WW^R, \\ \mathit{VERTEX} &= \textstyle\prod_{i=1}^{n}(\langle v_i \rangle WW^R), \\ \mathit{EDGE} &= \textstyle\prod_{i=1}^{m}(\langle e_i \rangle WW^R).\end{aligned}$$

Finally, let $K' = 41n + m + \mu$.

This transformation can be done in polynomial time. We claim that G has a vertex cover of size at most K if and only if there is a (1,1)-caterpillar with at most K' edges that realizes a walk for x. The proof is not hard and left to the reader. □

We can show the following theorem by a similar reduction:

Theorem 5 *For any integers $k, l \geq 1$, the graph inference from a walk for (k, l)-caterpillars is NP-complete.*

5 Approximability

In this section, we describe some results of approximability of the problems we have dealt with so far.

An optimization problem consists of a set I of possible inputs, a map S which maps each $x \in I$ to a set of feasible solutions, and a measure $m : S(I) \to Q^+$. For a solution s, we call $m(s)$ the cost of s. An algorithm $\mathcal{A}$ for an optimization problem is called an ϵ-approximation algorithm if $cost(\mathcal{A}(x))/opt(x) \leq \epsilon$, where $cost(\mathcal{A}(x))$ is the cost of the solution of x produced by $\mathcal{A}$ and $opt(x)$ is the cost of an optimal solution of x.

Theorem 6 *Let $\epsilon < 2$. If there is a polynomial-time ϵ-approximation algorithm for the graph inference from a walk for trees of bounded degree 3, then $P = NP$.*

Proof Let $c > 0$ be any fixed constant. We modify the reduction in the proof of Theorem 1 by replacing μ with $c \cdot \mu$. This just means that the string W is stretched c times in the length. We denote this modified reduction by R_c. We can easily see that the reduction R_c also gives the NP-hardness of GIW(3-Deg-Tree).

Let x be the string produced by R_c and T be a tree of bounded degree 3 which realizes a walk for x. The following fact holds:

1. Assume that G has a vertex cover U with $|U| \leq K$. If T is smallest, T has $(2c+1)\mu - 1$ edges. Otherwise, T has at least $4c \cdot \mu$ edges.

2. If G does not have any vertex cover U with $|U| \leq K$, then T has at least $4c \cdot \mu$ edges.

Since $\frac{4c}{2c+1} \cdot ((2c+1)\mu - 1) < 4c\mu$, it holds that a $4c/(2c+1)$-approximation algorithm returns a tree of bounded degree 3 with $(2c+1)\mu - 1$ edges which realizes a walk for x if and only if G has a vertex cover U with $|U| \leq K$. For any $\epsilon < 2$, there is a constant c with $\epsilon \leq 4c/(2c+1)$. □

We next show that GIW(3-Deg-Tree) is MAXSNP-hard. By the result due to Arora et al. [2], this implies that there is no polynomial time approximation scheme for GIW(3-Deg-Tree) unless P = NP.

Let Π_1 and Π_2 be two optimization (maximization or minimization) problems. We say that Π_1 *L-reduces* to Π_2 if there are polynomial time algorithms f and g and constants α and $\beta > 0$ such that:

1. Given an instance I_1 of Π_1, algorithm f produces an instance I_2 of Π_2 such that the cost of an optimal solution of I_2, $opt(I_2)$, is at most $\alpha \cdot opt(I_1)$, and

2. Given any solution s_2 of I_2, algorithm g produces in polynomial time a solution s_1 of I_1 satisfying

$$|cost(s_1) - opt(I_1)| \leq \beta \cdot |cost(s_2) - opt(I_2)|.$$

Some basic facts about L-reductions are: First, the composition of two L-reductions is also an L-reduction. Second, if problem Π_1 L-reduces to problem Π_2 and Π_2 can be approximated in polynomial time with relative error δ, i.e., there is an algorithm $\mathcal{A}$ for Π_2 with

$$\delta \geq \frac{|opt(x) - cost(\mathcal{A}(x))|}{opt(x)},$$

then Π_1 can be approximated with relative error $\alpha\beta\delta$. In particular, if Π_2 has a polynomial time approximation scheme, then so does Π_1. The class MAXSNP$_0$ is a class of maximization problems defined syntactically in Papadimitriou and Yannakakis [9, 10]. It is known that every problem in this class can be approximated within *some* constant factor. MAXSNP is defined as the class of all optimization problems that are L-reducible to a problem in MAXSNP$_0$. A problem is MAXSNP-hard if every problem in MAXSNP can be L-reduced to it.

Theorem 7 *The graph inference from a walk for trees of bounded degree 3 is MAXSNP-hard.*

Proof We give an L-reduction from 4-DEGREE VERTEX COVER, which is shown in [9, 10] that the problem is MAXSNP-complete. The L-reduction is obtained in the following way: At first, in a similar way to construct the reduction in the proof of Theorem 1, which is from 3-DEGREE VERTEX COVER, we construct a reduction from 4-DEGREE VERTEX COVER which also shows the NP-hardness of GIW(3-Deg-Tree). Next, the reduction from 4-DEGREE VERTEX COVER is modified into the L-reduction.

We briefly describe a key of the L-reduction. A string x will be defined using the strings

$$\begin{array}{ll} X_{i,j} & \text{for } i \in \{1,2,\ldots,n\} \text{ and } j \in \{1,2,3,4\}, \\ \langle v,j\rangle & \text{for } v \in V \text{ and } j \in \{1,2,3,4\}, \\ \langle e\rangle & \text{for } e \in E, \\ W' \text{ and } W, & \end{array}$$

which are similar to $X_{i,j}, \langle v,j\rangle, \langle e\rangle, W'$ and W in the proof of Theorem 1, respectively. Suppose that there is a tree T of bounded degree 3 with at most $3\mu - 1$ edges which realizes a walk w for x, where $\mu = 70n+m+2$. It should be easily seen that the induced subgraph, denoted by T_i, of the subwalks of w for $X_{i,1}, X_{i,2}, X_{i,3}$ and $X_{i,4}$ is isomorphic to the tree in Fig. 20. We denote by $\overline{d}_i$ the new free node of the tree, which is one endpoint of the edge labeled r_2 and also one endpoint of the edge labeled r_3.

Furthermore, it is also easy to see that if the subwalk for $\langle e\rangle$ selects T_i, i.e., the end node of the subwalk for the prefix $(a_1a_1)^n$ of $\langle e\rangle$ coincides with the node of T_i corresponding to the node $\odot$ of the tree in Fig. 20, then the node of T_i corresponding to the node $\overline{d}_i$ of the tree in Fig. 20 must have an adjacent edge labeled γ. Then, the number of edges labeled γ in a tree realizing a walk for x is the size of a vertex cover.

We define the string x as follows:

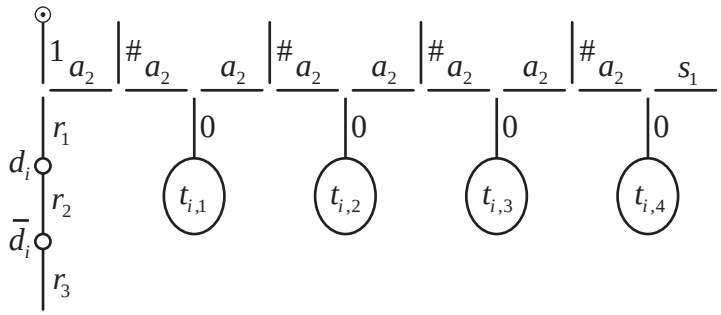


Figure 20: This tree has one internal free node $\overline{d}_i$ in addition to the nodes d_i and $d_{i,j}, \overline{d}_{i,j}$ in $t_{i,j}$ for $j = 1,2,3,4$.

(1) For $i \in \{1,2,\ldots,n\}$ and $j \in \{1,2,3,4\}$,

$$\begin{array}{rcl} x_{i,j} & = & 0g_1h_1h_2h_2h_1g_2g_2h_1\ \overline{\alpha}_j\overline{\alpha}_j\ h_2g_3g_3h_2\ \alpha_j\alpha_j\ h_1g_1\ \beta_{i,j}\beta_{i,j}\ 0. \\ X_{i,j} & = & (a_1\#\#a_1)^i\ 1\ (r_1r_2r_3r_3r_2r_1)^{\lfloor(4-j)/3\rfloor}\ (a_2\#\#a_2)^j \\ & & x_{i,j}\ (s_1s_1)^{\lfloor j/4\rfloor}\ (a_2a_2)^j\ 1\ (s_2s_2)^{\lfloor i/n\rfloor}\ (a_1a_1)^i. \end{array}$$

(2) For $v \in V$ and $j \in \{1,2,3,4\}$,

$$\langle v,j\rangle \ = \ (a_1a_1)^n\ 1r_1\ vv\ r_1\ (a_2a_2)^j\ 0g_1\ vv\ h_1\ \alpha_j\alpha_j\ h_1g_10\ (a_2a_2)^j\ 1\ (a_1a_1)^n.$$

(3) For $e = \{u,v\} \in E$,

$$\begin{array}{rcl} \langle e\rangle & = & (a_1a_1)^n\ 1\ (r_1r_2\gamma\gamma r_2r_1)\ (a_2a_2)^4\ 0 \\ & & g_1h_1h_2h_2h_1\ uu\ h_1h_2h_2h_1\ vv\ h_1h_2h_2h_1g_10\ (a_2a_2)^4\ 1\ (a_1a_1)^n. \end{array}$$

(4) $W = \prod_{i=1}^{\mu} b_i$ and $W' = \prod_{i=1}^{\mu}(\#\#b_i)$, where $\mu = 70n + m + 2$ and b_i is defined in the proof of Theorem 1.

(5) $x = TEMPLATE \cdot VERTEX \cdot EDGE$

where

$$\begin{array}{rcl} TEMPLATE & = & W^RW'W^R\ \prod_{i=1}^{n}(\prod_{j=1}^{4}(X_{i,j}WW^R)), \\ VERTEX & = & \prod_{j=1}^{4}(\prod_{i=1}^{n}(\langle v_i,j\rangle WW^R)), \\ EDGE & = & \prod_{i=1}^{m}(\langle e_i\rangle WW^R). \end{array}$$

The string x can be produced in polynomial time. The first condition of L-reduction is satisfied with $\alpha = 2120$ since $\lceil \frac{m}{4} \rceil \leq opt(G)$ and $\lceil \frac{n}{5} \rceil \leq opt(G)$, where $opt(G)$ is the size of minimum covers of G.

We next define an algorithm g as follows: Let s_2 be a solution of GIW(3-Deg-Tree), i.e., a tree T of bounded degree 3 which realizes a walk for x. If s_2 has at most $3\mu - 1$ edges, the algorithm g returns the subset U of V such that for each $v \in V$, v is in U if and only if there is $i \in \{1,\ldots,n\}$ satisfying that the node of T corresponding to d_i of the tree in Fig. 20 has an adjacent edge labeled v and the

node $\tilde{d}_i$ of T corresponding to $\overline{d}_i$ is not free, i.e., $\tilde{d}_i$ has an adjacent edge labeled γ. Otherwise, g returns V. Then it is trivial that the second condition holds with $\beta = 1$.

□

It is easy to see that for GIW(k-Deg-Tree) with $k > 3$, the same results of approximability as GIW(3-Deg-Tree) hold even if the alphabet size is restricted to $k + 1$.

We omit the proofs of the following theorems:

Theorem 8 *Let $\epsilon < 2$. If there is a polynomial-time ϵ-approximation algorithm for the graph inference from a walk for (k, l)-caterpillars for some integers $k, l \geq 1$, then $P = NP$.*

Theorem 9 *For any integers $k, l \geq 1$, the graph inference from a walk for (k, l)-caterpillars is MAXSNP-hard.*

Concluding Remarks

We have shown that the graph inference from a walk for trees of bounded degree k is NP-complete for any $k \geq 3$ even if the alphabet size is restricted to $k + 1$. We also have shown that the graph inference from a walk for (k, l)-caterpillars is NP-complete for $k, l \geq 1$. In addition we have seen that these problems are MAXSNP-hard and the approximation rate cannot be less than 2 unless P=NP. The following problems remain open:

(1) Is there any polynomial-time ϵ-approximation algorithm for GIW(3-Deg-Tree), for some $\epsilon \geq 2$?

(2) Let $k, l \geq 1$. Does GIW((k, l)-Caterpillar) remain NP-complete even if the alphabet size is restricted to a constant?

(3) Let $\epsilon \geq 2$. Is there any polynomial-time ϵ-approximation algorithm for GIW((k, l)-Caterpillar) for $k, l \geq 1$.

(4) Is GIW((k, l)-Caterpillar) solvable in polynomial time if either k or l is unbounded, or both are unbounded?

Acknowledgment

The authors would like to thank Shinichi Shimozono for fruitful discussions.

References

[1] D. Angluin. On the complexity of minimum inference of regular sets. *Inform. Control*, 39:337–350, 1978.

[2] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and hardness of approximation problems. In *Proc. 33rd IEEE Symp. Foundations of Computer Science*, pages 14–23, 1992.

[3] J. A. Aslam and R. L. Rivest. Inferring graphs from walks. In *Proc. 3rd Workshop on Computational Learning Theory*, pages 359–370, 1990.

[4] M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman and Company, 1979.

[5] E. M. Gold. Complexity of automaton identification from given data. *Inform. Control*, 37:302–320, 1978.

[6] J. Haralambides, F. Makedon, and B. Monien. Bandwidth minimization: An approximation algorithm for caterpillars. *Math. Systems Theory*, pages 169–177, 1991.

[7] G. Huet. Confluent reductions: abstract properties and applications to term rewriting systems. *J. Assoc. Comput. Mach.*, 27:797–821, 1980.

[8] O. Maruyama and S. Miyano. Inferring a tree from walks. In *Proc. 17th Mathematical Foundations of Computer Science, Lecture Notes in Computer Science*, volume **629**, pages 383–391, 1992.

[9] C. Papadimitriou and M. Yannakakis. Optimization, approximation and complexity classes. *J. Comput. System Sci.*, 43(3):425–440, 1991.

[10] C. H. Papadimitriou. *Computational Complexity*. Addison-Wesley Publishing Company, 1994.

[11] V. Raghavan. Bounded degree graph inference from walks. *J. Comput. System Sci.*, 49:108–132, 1994.

[12] S. Rudich. Inferring the structure of a markov chain from its output. In *Proc. 26th IEEE Symp. Foundations of Computer Science*, pages 321–326, 1985.