LETTER

On the Proper-Path-Decomposition of Trees

Atsushi TAKAHASHI†, Shuichi UENO† *and* Yoji KAJITANI††, *Members*

SUMMARY We introduce the interval set of a graph G which is a representation of the proper-path-decomposition of G, and show a linear time algorithm to construct an optimal interval set for any tree T. It is shown that a proper-path-decomposition of T with optimal width can be obtained from an optimal interval set of T in $O(n \log n)$ time.
key words: *proper-path-width, proper-path-decomposition, path-width, path-decomposition, polynomial time algorithm*

1. Introduction

Graphs we consider are connected, have at least two vertices, and may have loops and multiple edges. Let G be a graph, and $V(G)$ and $E(G)$ denote the vertex set and edge set of G, respectively. Let $\mathcal{X}=(X_1, X_2, \cdots, X_r)$ be a sequence of subsets of $V(G)$. The *width* of $\mathcal{X}$ is $\max_{1\leq i\leq r}|X_i|-1$. $\mathcal{X}$ is called a *proper-path-decomposition* of G if the following conditions are satisfied: (i) For any distinct i and j, $X_i \subsetneqq X_j$; (ii) $\bigcup_{i=1}^{r} X_i = V(G)$; (iii) For any edge $(u, v) \in E(G)$, there exists an i such that $u, v \in X_i$; (iv) For all a, b, and c with $1\leq a\leq b\leq c\leq r$, $X_a \cap X_c \subseteq X_b$; (v) For all a, b, and c with $1\leq a<b<c\leq r$, $|X_a \cap X_c|\leq |X_b|-2$. The *proper-path-width* of G, denoted by $ppw(G)$, is the minimum width over all proper-path-decompositions of G. If $\mathcal{X}$ satisfies (i), (ii), (iii), and (iv), $\mathcal{X}$ is called a *path-decomposition* of G. The *path-width* of G, denoted by $pw(G)$, is the minimum width over all path-decompositions of G. Notice that $\mathcal{X}$ satisfies condition (iv) if and only if each vertex of G appears in consecutive X_i's [11]. It is not difficult to see that a path-decomposition $\mathcal{X}$ satisfies condition (v) if and only if $|X_{i-1} \cap X_{i+1}|\leq |X_i|-2$ holds for any i with $2\leq i\leq r-1$ [12]. A (proper-)path-decomposition with width k is called a k-(proper-)path-decomposition. Many graph parameters which are equivalent to the path-width or proper-path-width can be found in the literature [1], [3], [5], [6], [8], [10]–[12].

It is known that the problems of computing $pw(G)$ and $ppw(G)$ are NP-hard for general graphs but can be solved in linear time for trees [4], [8], [10], [12]. It is also known that for any fixed integer k, a k-path-decomposition of G with path-width at most k can be obtained, if exists, in $O(n \operatorname{los} n)$ time for general graphs by combining the results in [1] and [9], and in $O(n+e)$ time for cographs [2], where $n=|V(G)|$ and $e=|E(G)|$.

In this paper, we give an $O(n \log n)$ time algorithm to obtain a $ppw(T)$-proper-path-decomposition of a tree T with n vertices. It should be noted that our algorithm works for any tree with unbounded proper-path-width, and it is a linear time algorithm for trees with a bounded proper-path-width. We introduce the interval set of a graph G which is a representation of the proper-path-decomposition of G, and show a linear time algorithm to construct an optimal interval set for any tree T. We show that a $pw(T)$-proper-path-decomposition of T can be obtained from an optimal interval set of T in $O(n \log n)$ time. By a similar argument, a $pw(T)$-path-decomposition can be found in $O(n \log n)$ time for any tree T with n vertices.

2. Interval Set and Proper-Path-Decomposition

Let Z be the set of integers. We denote an *interval* on integers by I. Two intervals I_1 and I_2 on integers are said to be *adjacent* if there exist integers $i \in I_1$ and $j \in I_2$ such that $|i-j|\leq 1$, and said to be *independent* if there exists no integer $i \in I_1 \cap I_2$ such that $\{i-1, i+1\} \subsetneqq I_1$ and $\{i-1, i+1\} \subsetneqq I_2$. A set $\mathcal{J}$ of distinct non-singleton intervals on integers such that any two distinct intervals are independent is called an *interval set* of a graph G if there exists a one-to-one correspondence J: $V(G) \rightarrow \mathcal{J}$ such that $J(u)$ and $J(v)$ are adjacent if $(u, v) \in E(G)$. For any $i \in Z$ and a set $\mathcal{J}$ of intervals on integers, define $\mathcal{J}(i)=\{I | i \in I, I \in \mathcal{J}\}$. The *density* of $\mathcal{J}$ is $\max_{i\in Z}|\mathcal{J}(i)|$. An interval set $\mathcal{J}$ of G is said to be *optimal* if the density of $\mathcal{J}$ is minimum over all interval sets of G.

In the following, we denote $a \in A$ if a is a member of a sequence A. The sequence obtained by concatenating sequences A_i $(1\leq i\leq r)$ is denoted by $(A_1, A_2, \cdots, A_r)$.

Suppose that $\mathcal{J}$ is an interval set of G with a one-to-one correspondence J: $V(G) \rightarrow \mathcal{J}$. For any vertex $v \in V(G)$, define that $l(v)$ (respectively, $r(v)$)

Manuscript received April 8, 1994.
Manuscript revised July 30, 1994.
† The authors are with the Department of Electrical and Electronic Engineering, Tokyo Institute of Technology, Tokyo, 152 Japan.
†† The author is with the School of Information Science, Japan Advanced Institute of Science and Technology, Ishikawa-ken, 923-12 Japan.

is the integer i such that $i \in J(v)$ and $i-1 \notin J(v)$ (respectively, $i+1 \notin J(v)$). A sequence $(v_1, v_2, \cdots, v_{|\mathcal{J}|})$ of $V(G)$ is called the left (respectively, right) terminal sequence of $\mathcal{J}$ if $l(v_1)<l(v_2)<\cdots<l(v_{|\mathcal{J}|})$ (respectively, $r(v_1)<r(v_2)<\cdots<r(v_{|\mathcal{J}|})$). A sequence $(L_1, R_1, L_2, R_2, \cdots, L_r, R_r)$ is called the *terminal sequence* of $\mathcal{J}$ if the following conditions are satisfied: $(L_1, L_2, \cdots, L_r)$ and $(R_1, R_2, \cdots, R_r)$ are the left and right terminal sequences of $\mathcal{J}$, respectively; both L_i and R_i are nonempty $(1 \leqq i \leqq r)$; for any vertices $u \in L_i$ and $v \in R_i$ $(1 \leqq i \leqq r)$, $l(u)<r(v)$; for any vertices $v \in R_i$ and $u \in L_{i+1}$ $(1 \leqq i \leqq r-1)$, $r(v)<l(u)$. Notice that $l(u) \neq l(v)$, $r(u) \neq r(v)$, and $l(u) \neq r(v)$ for any distinct vertices $u, v \in V(G)$.

Before proving Theorem 1 below, we need the following lemmas.

Lemma 1: For any graph G, there exists an optimal interval set of G with the terminal sequence $(L_1, R_1, \cdots, L_r, R_r)$ such that $|L_r|=1$ and $r \geqq 2$.

Proof: Suppose that $\mathcal{J}$ is an optimal interval set of G with a one-to-one correspondence J: $V(G) \rightarrow \mathcal{J}$ and the terminal sequence $(L_1, R_1, \cdots, L_r, R_r)$. Since $|V(G)| \geqq 2$, if $|L_r|=1$ then $r \geqq 2$. Thus we assume that $|L_r| \geqq 2$. Let v be the vertex in $V(G)$ such that $l(v)=\max_{w \in L_r} l(w)$, and u be the vertex in $V(G)$ such that $r(u)=\min_{w \in R_r-\{v\}} r(w)$. Define that $J'(v)=\{i \mid l(v)+1 \leqq i \leqq \max_{w \in R_r} r(w)+1, i \in Z\}$, $J'(u)=\{i \mid l(u) \leqq i \leqq l(v), i \in Z\}$, and $J'(w)=J(w)$ for any $w \in V(G)-\{u, v\}$. Let L'_r be the sequence obtained from L_r by deleting v, and R'_r be the sequence obtained from R_r by deleting u and moving v into the last. Then it is not difficult to see that $\{J'(w) \mid w \in V(G)\}$ is an optimal interval set of G with the terminal sequence $(L_1, R_1, \cdots, R_{r-1}, L'_r, u, v, R'_r)$. Thus we have this lemma. □

Lemma 2: For any (proper-)path-decomposition $(X_1, X_2, \cdots, X_r)$ of G, $|X_i| \geqq 2$ $(1 \leqq i \leqq r)$.

Proof: Suppose that $X_l=\{v\}$ for some l $(1 \leqq l \leqq r)$. Since G is connected and contains at least two vertices, there exists $u \in V(G)-\{v\}$ such that $(v, u) \in E(G)$. Thus $\{u, v\} \in X_i$ for some i $(1 \leqq i \leqq r)$ by condition (iii) in the definition of proper-path-decomposition. But this is contradicting to condition (i) in the definition of proper-path-decomposition since $X_l \subset X_i$. □

Theorem 1: For any graph G and an integer k $(k \geqq 1)$, there exists a proper-path-decomposition of G with width k if and only if there exists an interval set of G with density k.

Proof: Suppose that $\mathcal{X}=(X_1, X_2, \cdots, X_r)$ is a k-proper-path-decomposition of G. Let $V_1=X_1$, $V_i=X_i-X_{i-1}$ $(2 \leqq i \leqq r)$, $U_i=X_i-X_{i+1}$ $(1 \leqq i \leqq r-1)$, and $U_r=X_r$. Let $v_i \in V_i$ and $u_i \in U_i$ such that $v_i \neq u_i$ $(1 \leqq i \leqq r)$. Notice that $V_i \neq \emptyset$, $U_i \neq \emptyset$, and $|V_i \cup U_i| \geqq 2$ by Lemma 2 and conditions (i) and (v) in the definition of the proper-path-decomposition. Let $\mathcal{J}$ be the set of intervals defined as follows:

1. Let $i=1$ and $j=1$;
2. For each vertex $w \in V_i-\{v_i\}$, define $l(w)=j$ and let $j=j+1$;
3. Define $r(u_i)=j$ and $l(v_i)=j+1$, and let $j=j+2$;
4. For each vertex $w \in U_i-\{u_i\}$, define $r(w)=j$ and let $j=j+1$;
5. If $i<r$ then let $i=i+1$ and return to 2;
6. Define $J(w)=\{i \mid l(w)<i<r(w), i \in Z\}$ for any $w \in V(G)$, and let $\mathcal{J}=\{J(w) \mid w \in V(G)\}$.

First, we show that the intervals in $\mathcal{J}$ are well-defined. Since both of $(V_1, V_2, \cdots, V_r)$ and $(U_1, U_2, \cdots, U_r)$ are partitions of $V(G)$, both $l(w)$ and $r(w)$ are defined for any vertex $w \in V(G)$. Assume that $w \in V_i$ $(1 \leqq i \leqq r)$ and $w \in U_j$ $(1 \leqq j \leqq r)$. If $j<i$ then $w \in X_j \cap X_i$ and $w \notin X_{j+1}$. But this is contradicting to condition (iv) in the definition of the proper-path-decomposition since $X_j \cap X_i \nsubseteq X_{j+1}$. Thus $i \leqq j$. If $i<j$ then trivially $l(w)<r(w)$ by the definition of $l(w)$ and $r(w)$. If $i=j$ then also $l(w)<r(w)$ since $v_i \neq u_i$. Thus $J(w)$ is a non-singleton interval on integers for any vertex $w \in V(G)$. Hence $\mathcal{J}$ is a set of distinct non-singleton intervals on integers such that any two distinct intervals in $\mathcal{J}$ are independent, and J: $V(G) \rightarrow \mathcal{J}$ is a one-to-one correspondence. Next, we show that $\mathcal{J}$ is an interval set of G. For some edge $(u, v) \in E(G)$, assume that $\{u, v\} \subseteq X_i$ by condition (iii) in the definition of the proper-path-decomposition. If $\{u, v\} \subseteq X_i-\{v_i\}$ then intervals $J(u)$ and $J(v)$ are adjacent to each other since $\{J(u), J(v)\} \subseteq \mathcal{J}(r(u_i))$. Similarly, if $\{u, v\} \subseteq X_i-\{u_i\}$ then intervals $J(u)$ and $J(v)$ are adjacent to each other since $\{J(u), J(v)\} \subseteq \mathcal{J}(l(v_i))$. Otherwise ($\{u, v\}=\{u_i, v_i\}$) intervals $J(u)$ and $J(v)$ are adjacent to each other since $l(v_i)-r(u_i)=1$. Thus for any edge $(u, v) \in E(G)$, intervals $J(u)$ and $J(v)$ are adjacent to each other. That is, $\mathcal{J}$ is an interval set of G. Finally, we show that the density of $\mathcal{J}$ is k. It is easy to see that $\max_{w \in V_i}|\mathcal{J}(l(w))|=|\mathcal{J}(r(u_i))|=|\mathcal{J}(l(v_i))|=\max_{w \in U_i}|\mathcal{J}(r(w))|$ for any i $(1 \leqq i \leqq r)$. Since $\max_{1 \leqq i \leqq r}|\mathcal{J}(l(v_i))|=\max_{1 \leqq i \leqq r}|X_i-\{u_i\}|=k$, the density of $\mathcal{J}$ is k. Thus $\mathcal{J}$ is an interval set of G with density k.

Conversely, suppose that $\mathcal{J}$ is an interval set of G with the terminal sequence $(L_1, R_1, \cdots, L_r, R_r)$ and density k. By Lemma 1, without loss of generality, we assume that $r \geqq 2$ and $|L_r|=1$. Let v_i be the vertex such that $l(v_i)=\min_{w \in L_i} l(w)$ for any i $(1 \leqq i \leqq r)$. We define a sequence $\mathcal{X}=(X_1, X_2, \cdots, X_{r-1})$ as follows:

(i) Define $X_1=L_1 \cup \{v_2\}$;

(ii) Given X_i $(1 \leqq i \leqq r-2)$, define $X_{i+1}=(X_i \cup L_{i+1} \cup \{v_{i+2}\})-R_i$;

Since $R_i \cap L_{i+1}=\emptyset$ $(1 \leqq i \leqq r-2)$ and $L_r=\{v_r\}$, $\mathcal{X}$ satisfies conditions (ii) and (iv) in the definition of the proper-path-decomposition. Since $v_{i+2} \in X_{i+1}-X_i$ and $X_i-X_{i+1}=R_i \neq \emptyset$ $(1 \leqq i \leqq r-2)$, $X_i \nsubseteq X_{i+1}$ and $X_{i+1} \nsubseteq X_i$. Thus $X_i \nsubseteq X_j$ for any distinct i and j, for otherwise $X_i=X_i \cap X_j \subseteq X_{i+1}$ $(i<j)$ or $X_i=X_i \cap X_j \subseteq X_{i-1}$ $(i>j)$. Hence $\mathcal{X}$ satisfies condition (i) in the definition of the proper-path-decomposition. Let v'_i be the vertex such that $l(v_i') = \max_{w \in L_i} l(w)$, and u_i' be the vertex such that $r(u_i') = \max_{w \in R_i} r(w)$ $(1 \leqq i \leqq r)$. Let J: $V(G) \rightarrow \mathcal{J}$ be a one-to-one correspondence. Since $\cup_{w \in L_i} \mathcal{J}(l(w)) = \cup_{w \in R_i} \mathcal{J}(r(w)) = \mathcal{J}(l(v_i'))$ for any i $(1 \leqq i \leqq r)$, if two intervals $I_1, I_2 \in \mathcal{J}$ are adjacent then $I_1, I_2 \in \mathcal{J}(l(v_i'))$ or $\{I_1, I_2\} = \{J(u_i'), J(v_{i+1})\}$ $(1 \leqq i \leqq r-1)$. Notice that $\mathcal{J}(l(v_r')) \subseteq \{J(v) | v \in X_{r-1}\}$ since $v_r = v_r' \in X_{r-1}$. Since $\{J(v) | v \in X_i\} = \mathcal{J}(l(v_i')) \cup \{J(v_{i+1})\}$ $(1 \leqq i \leqq r-1)$, if two intervals $J(u), J(v) \in \mathcal{J}$ are adjacent then $u, v \in X_i$. Notice that $u_i' \in X_i$ $(1 \leqq i \leqq r-1)$. Thus by definition of an interval set, $\mathcal{X}$ satisfies condition (iii) in the definition of the proper-path-decomposition. Since $v_{i+1} \notin X_{i-1} \cup R_i$ and $\emptyset \neq R_i \nsubseteq X_{i+1}$, we have $|X_{i-1} \cap X_{i+1}| \leqq |X_i| - 2$ $(2 \leqq i \leqq r-2)$. Thus $\mathcal{X}$ satisfies condition (v) in the definition of the proper-path-decomposition. Since $\max_{1 \leqq i \leqq r-1} |X_i| = \max_{1 \leqq i \leqq r-1} |\mathcal{J}(l(v_i')) \cup \{J(v_{i+1})\}| = k+1$, the width of $\mathcal{X}$ is k. Therefore $\mathcal{X}$ is a k-proper-path-decomposition of G. □

Corollary 1: For any graph G on n vertices, a k-proper-path-decomposition of G can be obtained in $O(kn)$ time if the terminal sequence of an interval set of G with density k is given.

Notice that $r \leqq n-k$ for any k-proper-path-decomposition $(X_1, X_2, \cdots, X_r)$ of G on n vertices.

3. The Algorithm

We define the path-vector $\overline{pv}(v, T) = (p, c, n)$ for any tree T with a vertex $v \in V(T)$ as the root to compute $ppw(T)$. p describes the proper-path-width of T. c and n describe the condition of T as follows: If there exists $u \in V(T) - \{v\}$ such that $T \backslash \{u\}$, the graph obtained from T by deleting u, has two connected components with proper-path-width $ppw(T)$ and without v, then $c=3$ and n is the path-vector of the connected component of $T \backslash \{u\}$ containing v; otherwise, c is the number of the connected components of $T \backslash \{v\}$ with proper-path-width $ppw(T)$ and $n = nul$. It should be noted that for any vertex $u \in V(T)$ the number of connected components of $T \backslash \{u\}$ with proper-path-width $ppw(T)$ is at most two [11]. Notice also that if there exists u such that $T \backslash \{u\}$ has two connected components with proper-path-width $ppw(T)$ and without v then u is uniquely determined. If there is no such u then the number of connected components of $T \backslash \{w\}$ with proper-path-width $ppw(T)$ and without v is not more than the number of connected components of $T \backslash \{v\}$ with proper-path-width $ppw(T)$. In the following, we denote an element x in $\overline{pv}(v, T)$ by $\overline{pv}(v, T)|x$.

Let T_0 be a tree with root $v \in V(T_0)$ and P_0 be the path-vector of T_0. We recursively define T_i and P_i $(1 \leqq i \leqq l)$ while $P_{i-1}|c=3$ as follows: Let $u_{i-1} \in V(T_{i-1}) - \{v\}$ be the vertex such that $T_{i-1} \backslash \{u_{i-1}\}$ has two connected components with proper-path-width $ppw(T_{i-1})$ and without v, T_i be the connected component of $T_{i-1} \backslash \{u_{i-1}\}$ containing v as the root, and P_i be the path-vector of T_i. Assume that $P_l|c \neq 3$. We call such path-vectors $P_0, P_1, \cdots, P_l$ the chain of the path-vector P_0. We define b, n^*, b^*, and btm in the chain of P_0 as follows: Define that $P_i|b = P_{i-1}$ $(1 \leqq i \leqq l)$; define that $P_i|n^* = P_j$ if $i=0$ or $P_i|p < P_{i-1}|p - 1$ $(1 \leqq i \leqq l)$ where j is the maximum integer such that $j - i = P_i|p - P_j|p$; define that $P_i|b^* = P_j$ if $P_j|n^*$ is defined and $P_j|n^* = P_i$; define that $P_0|btm = P_l$. Thus we extend a path-vector as $\overline{pv}(v, T) = (p, c, n, b, n^*, b^*, btm)$ to reduce the time to traverse the chain as used in [7]. It was shown that we can compute $ppw(T)$ in linear time for any tree T by computing path-vectors of subtrees of T [12].

As shown in Figs. 1 and 2, we can modify the algorithm in [12] to construct the terminal sequence of an optimal interval set of a tree.

Let T_0 be a tree with root $v_0 \in V(T_0)$ and proper-path-width k. Suppose that $\overline{pv}(v_0, T_0)|c=2$. Let T_1 be a connected component of $T_0 \backslash \{v_0\}$ with proper-path-width k, and $v_1 \in V(T_1)$ be the vertex adjacent to v_0 in T_0. We recursively define T_i and $v_i \in V(T_i)$ $(2 \leqq i \leqq a)$ while $T_{i-1} \backslash \{v_{i-1}\}$ has a component with proper-path-width k as follows: Let T_i be a connected component of $T_{i-1} \backslash \{v_{i-1}\}$ with proper-path-width k and $v_i \in V(T_i)$ be the vertex adjacent to v_{i-1} in T_{i-1}. $T_a \backslash \{v_a\}$ has no connected component with proper-path-width k. Let T_{a+1} be the other connected component of $T_0 \backslash \{v_0\}$ with proper-path-width k, and $v_{a+1} \in V(T_{a+1})$ be the vertex adjacent to v_0 in T_0. Define recursively T_i and $v_i \in V(T_i)$ $(a+2 \leqq i \leqq b)$ as above. Notice that $T_i \backslash \{v_i\}$ $(1 \leqq i \leqq b)$ has at most one connected component with proper-path-width k, for otherwise $T_0 \backslash \{v_i\}$ has three or more connected components with proper-path-width k. Let H_i' $(0 \leqq i \leqq b)$ be the union of components of $T_i \backslash \{v_i\}$ with proper-path-width $\leqq k-1$, and H_i $(0 \leqq i \leqq b)$ be the induced subgraph of T_0 on $V(H_i') \cup \{v_i\}$. Let W_i' be the terminal sequence of an optimal interval set of H_i'. Since $ppw(H_i') \leqq k-1$ $(0 \leqq i \leqq b)$, $W_i = (v_i, W_i', v_i)$ is the terminal sequence of an interval set of H_i with density at most k by Theorem 1. It is easy to see that there exists an interval set $\mathcal{J}$ of T_0 with density k such that the terminal sequence of $\mathcal{J}$ is $(W_a, W_{a-1}, \cdots, W_1, W_0, W_{a+1}, W_{a+2}, \cdots, W_b)$.

Thus, if $\overline{pv}(v_0, T_0)|c=2$, we assume that the terminal sequence of an interval set of T_0 with density k is $(W_L, v_0, W_0', v_0, W_R)$ where $W_L = (W_a, W_{a-1}, \cdots, W_1)$ and $W_R = (W_{a+1}, W_{a+2}, \cdots, W_b)$. If $\overline{pv}(v_0, T_0)|c=1$ then $T_0 \backslash \{v_0\}$ has just one connected component with proper-path-width k, the sequence W_R above is empty, and we assume that the terminal sequence of an interval set of T_0 with density k is (W_L, v_0, W_0', v_0). Similarly, if $\overline{pv}(v_0, T_0)|c=0$ then $T_0 \backslash \{v_0\}$ has no connected component with proper-path-width k, and we assume that the terminal sequence of an interval set of T_0 with density k is (v_0, W_0', v_0).

If $\overline{pv}(v_0, T_0)|c \leqq 2$ then we denote a terminal sequence, W_L, W_0', W_R, v, and (W_L, W_0', W_R) by W, L,

Procedure MERGE(P_s, P_t)

Input: P_s (path-vector of tree T_s rooted at s)
P_t (path-vector of tree T_t rooted at t)
Output: the path-vector of tree rooted at s
obtained from T_s and T_t by adding an edge (s,t).

1. if $P_s|p > P_t|p$ then
1.1 if $P_s|c \le 2$ then $P_s := (p, c, -, \{L, r, -, (P_t|W, C), r, R\}, (L, P_t|W, C, R))$;
1.2 else if $P_s|n^*|p < P_t|p$ then $P_s := (p+1, 0, -, \{-, r, -, (P_t|W, D), r, -\}, (P_t|W, D))$;
1.3 else if $P_s|n^*|p = P_t|p$ then
1.3.1 if $P_s|n^*|c \ge 2$ or $P_t|c \ge 2$ then $P_s := (p+1, 0, -, \{-, r, -, (P_t|W, D), r, -\}, (P_t|W, D))$;
1.3.2 else if $P_s|n^*|c = 0$ then $P_s|n^* := (p, 1, -, \{P_t|W, r, -, C, r, -\}, (P_t|W, D))$;
1.3.3 else if $P_s|n^*|c = 1$ then $P_s|n^* := (p, 2, -, \{L, r, -, C, r, P_t|\overline{W}\}, (D, P_t|\overline{W}))$;
endif
1.4 else if $P_s|n^*|c \le 2$ then $P_s|n^* := (p, c, -, \{L, r, -, (P_t|W, C), r, R\}, (L, P_t|W, C, R))$;
1.5 else if $P_s|n^*|c = 3$ then
1.5.1 $P_s|n^*|n :=$ MERGE($P_s|n^*|n$, P_t);
1.5.2 if $P_s|n^*|n|p = P_s|n^*|p$ then $P_s := (p+1, 0, -, \{-, r, -, D, r, -\}, D)$;
endif
1.6 return(P_s);
2. else if $P_s|p = P_t|p$ then
2.1 if $P_s|c \ge 2$ or $P_t|c \ge 2$ then $P_s := (p+1, 0, -, \{-, r, -, (P_t|W, D), r, -\}, (P_t|W, D))$;
2.2 else if $P_s|c = 0$ then $P_s := (p, 1, -, \{P_t|W, r, -, C, r, -\}, (P_t|W, D))$;
2.3 else if $P_s|c = 1$ then $P_s := (p, 2, -, \{L, r, -, C, r, P_t|\overline{W}\}, (D, P_t|\overline{W}))$;
endif
2.4 return(P_s);
3. else if $P_s|p < P_t|p$ then
3.1 if $P_t|c \le 1$ then $P_t := (p, 1, -, \{W, P_s|r, -, P_s|D, P_s|r, -\}, (W, P_s|D))$;
3.2 else if $P_t|c = 2$ then $P_t := (p, 3, P_s, \{L, r, P_s|W, C, r, R\}, (L, r, P_s|D, C, r, R))$;
3.3 else if $P_s|p > P_t|n^*|p$ then $P_t := (p+1, 0, -, \{-, P_s|r, -, (W, P_s|D), P_s|r, -\}, (W, P_s|D))$;
3.4 else if $P_s|p = P_t|n^*|p$ then
3.4.1 if $P_s|c \ge 2$ or $P_t|n^*|c \ge 2$ then
$P_t := (p+1, 0, -, \{-, P_s|r, -, (W, P_s|D), P_s|r, -\}, (W, P_s|D))$;
3.4.2 else if $P_s|c = 0$ then $P_t|n^* := (p, 1, -, \{W, P_s|r, -, P_s|C, P_s|r, -\}, (W, P_s|D))$;
3.4.3 else if $P_s|c = 1$ then $P_t|n^* := (p, 2, -, \{P_s|L, P_s|r, -, P_s|C, P_s|r, \overline{W}\}, (P_s|D, \overline{W}))$;
endif
3.5 else if $P_t|n^*|c \le 1$ then $P_t|n^* := (p, 1, -, \{W, P_s|r, -, P_s|C, P_s|r, -\}, (W, P_s|D))$;
3.6 else if $P_t|n^*|c = 2$ then $P_t|n^* := (p, 3, P_s, \{L, r, P_s|W, C, r, R\}, (L, r, P_s|D, C, r, R))$;
3.7 else if $P_t|n^*|c = 3$ then
3.7.1 $P_t|n^*|n :=$ MERGE(P_s, $P_t|n^*|n$);
3.7.2 if $P_t|n^*|n|p = P_t|n^*|p$ then $P_t := (p+1, 0, -, \{-, P_s|r, -, D, P_s|r, -\}, D)$;
endif
3.8 return(P_t);
endif
END

Fig. 1 Procedure MERGE.

C, R, r, and D, respectively.

Suppose that $\overline{pv}(v_0, T_0)|c=3$. Let $u \in V(T_0) - \{v_0\}$ be the vertex such that $T_0 \backslash \{u\}$ has two connected components with proper-path-width k. Let T_L and T_R be two connected components of $T_0 \backslash \{u\}$ with proper-path-width k, T^* be the connected component of $T_0 \backslash \{u\}$ containing v_0, and T' be the union of the other connected components of $T_0 \backslash \{u\}$. Let $u_l \in T_L$ and $u_r \in T_R$ be the vertices adjacent to u in T_0. Since $T_L \backslash \{u_l\}$ has at most one connected component with proper-path-width k, $\overline{pv}(u_l, T_L)|c \leqq 1$ and $\overline{pv}(u_r, T_R)|c \leqq 1$. Thus we assume that the terminal sequences of optimal interval sets of T_L and T_R are $W_L = (W_L', u_l)$ and $W_R = (u_r, W_R')$, respectively. Then it is easy to see that there exists an interval set $\mathcal{I}$ of T_0 with density k such that the terminal sequence of $\mathcal{I}$ is $(W_L, u, W^*, W', u, W_R)$ where W^* and W' are the terminal sequences of optimal interval sets of T^*, and T', respectively.

If $\overline{pv}(v_0, T_0)|c=3$ then we denote a terminal sequence, W_L, W^*, W', W_R, and u by W, L, N, C, R, and r, respectively. Moreover, the sequence obtained from the terminal sequence by deleting v is denoted by D.

We extend a path-vector as $\overline{pv}(v, T) = (p, c, n, b, n^*, b^*, btm, \{L, r, N, C, r, R\}, D)$. Notice that $W = (L, r, N, C, r, R)$.

In the procedure, we omit the description of substitutions for b, n^*, b^*, and btm in the path-vector because no confusion is caused. Moreover, after substitutions, we can update n^*, b^*, and btm in the path-vectors in the chain in constant time. So we also omit the description of these operations. Thus we denote the path-vector $\overline{pv}(v, T) = (p, c, n, \{L, r, N, C, r, R\}, D)$. The reverse of a terminal sequence is denoted by $\overline{W}$, and maintained in the procedure together with the reverses of L, N, C, R, and D. But we also omit the

Procedure LMERGE(P_s, P_t)
[Input: P_s (path-vector of tree T_s rooted at s)
P_t (path-vector of tree T_t rooted at t)
Output: the path-vector of tree rooted at s
obtained from T_s and T_t by adding an edge (s,t).]
1. **if** $P_s|p > P_t|p$ **and** $P_s|c = 3$ **then**
1.1 **if** $P_s|btm|b^*|p \geq P_t|p$ **then** let P' be $P_s|btm|b^*$;
1.2 **else**
let P' be the path-vector P in the chain of P_s such that $P|n^*$ is defined and $P|p \geq P_t|p > P|n^*|n|p$;
1.3 $P' :=$ MERGE(P', P_t);
1.4 **return**(P_s);
endif
2. **if** $P_s|p < P_t|p$ **and** $P_t|c = 3$ **then**
2.1 **if** $P_t|btm|b^*|p \geq P_s|p$ **then** let P' be $P_t|btm|b^*$;
2.2 **else**
let P' be the path-vector P in the chain of P_t such that $P|n^*$ is defined and $P|p \geq P_s|p > P|n^*|n|p$;
2.3 $P' :=$ MERGE(P_s, P');
2.4 **return**(P_t);
endif
3. **return**(MERGE(P_s, P_t));
END

Procedure DFS(s)
[Input: a vertex s
Output: the path-vector of the maximal subtree rooted at s]
1. $P_s := (1, 0, -, \{-, s, -, -, s, -\}, -)$; /* path-vector of a tree with one vertex s */
2. **for all** children t of s in T **do**
2.1 $P_t :=$ DFS(t);
2.2 $P_s :=$ LMERGE(P_s, P_t);
endfor
3. **return**(P_s);
END

Procedure MAIN(T)
[Input: a tree T
Output: the proper-path-width of T]
1. Let r be a vertex in $V(T)$;
2. $\overline{pv}(r,T)$:= DFS(r);
3. **return**($\overline{pv}(r,T)|W$);
END

Fig. 2 The algorithm to construct the terminal sequence of an interval set of a tree.

description of these operations. For the simplicity, if the substitution for P uses $P|x$, we abbreviate $P|x$ to x.

Procedure MERGE shown in Fig. 1 recursively calculates the path-vector of T_0 from the path-vector P_s of T_s and the path-vector P_t of T_t in $O(\max(ppw(T_s), ppw(T_t)))$ time. Note that the time complexity of Procedure MERGE is $O(1)$ except for recursive calls. In Procedure LMERGE shown in Fig. 2, we can determine P' in $O(\min(ppw(T_s), ppw(T_t)))$ time by using btm and b^* in the chain of the path-vector. If P' is determined at 1.2 or 2.2 in Procedure LMERGE then the number of recursive calls of Procedure MERGE is at most $P'|n^*|n|p < \min(ppw(T_s), ppw(T_t))$. Otherwise Procedure MERGE returns the path-vector in $O(1)$ time. Thus Procedure LMERGE calculates the path-vector of the join of two subtrees in $O(\min(ppw(T_s), ppw(T_t)))$ time. Procedure DFS shown in Fig. 2 computes the path-vector of a maximal subtree rooted at s in T from the path-vectors of maximal subtrees rooted at children of s in T by using Procedure LMERGE. Procedure MAIN shown in Fig. 2 obtains the proper-path-width of T from the path-vector of T obtained by Procedure DFS. The algorithm starts with the isolated vertices obtained from T by deleting all edges in T and reconstruct T by adding edge by edge while computing path-vectors of connected components. Thus we can obtained the terminal sequence of an interval set of T with width $ppw(T)$ in linear time.

Theorem 2: For any tree T with proper-path-width k, the terminal sequence of an interval set of T with density k can be obtained in linear time.

By Corollary 1 and Theorem 2, we obtain the following theorem.

Theorem 3: For any tree T with proper-path-width k, a k-proper-path-decomposition of T can be obtained in $O(n \log n)$ time.

Notice that $ppw(T) = O(\log n)$ for any tree T on n vertices. It should be noted that a k-proper-path-decomposition of T, if exists, can be obtained in linear time if k is fixed. By a similar argument, a $pw(T)$-path-decomposition can be obtained in

$O(n \log n)$ time for any tree T with n vertices.

References

[1] Bodlaender, H. L. and Kloks, T., "Better algorithms for the pathwidth and treewidth of graphs," in J. L. Albert, B. Monien, and M. R. Artalejo, eds., *Proc. the 18th International colloquium on Automata, Languages and Programming*, vol. 510 of LNCS, pp. 544-555, Springer-Verlag, Berlin, New York, 1991.

[2] Bodlaender, H. L. and Möhring, R. H., "The pathwidth and treewidth of cographs," *SIAM J. Disc. Math.*, vol. 6, no. 2, pp. 181-188, 1993.

[3] Kirousis, L. M. and Papadimitriou, C. H., "Interval graphs and searching," *Discrete Mathematics*, vol. 55, pp. 181-184, 1985.

[4] Kirousis, L. M. and Papadimitriou, C. H., "Searching and pebbling," *Theoretical Computer Science*, vol. 47, pp. 205-218, 1986.

[5] Korach, E. and Solel, N., "Tree-width, path-width, and cutwidth," *Discrete Applied Mathematics*, vol. 43, pp. 97-101, 1993.

[6] Kornai, A. and Tuza, Z., "Narrowness, pathwidth, and their application in natural language processing," *Discrete Applied Mathematics*, vol. 36, pp. 87-92, 1992.

[7] Megiddo, N., Hakimi, S. L., Garey, M. R., Johnson, D. S. and Papadimitriou, C. H., "The complexity of searching a graph," *Journal of the Association for Computing Machinery*, vol. 35, no. 1, pp. 18-44, Jan. 1988.

[8] Möhring, R. H., "Graph problems related to gate matrix layout and PLA folding," in G. Tinhofer, E. Mayr, H. Noltemeier, and M. Syslo, eds., *Computational Graph Theory*, pp. 17-51, Springer-Verlag, Wien, New York, computing suppl. 7 edition, 1990.

[9] Reed, B. A., "Finding approximate separators and computing tree width quickly," in *Proc. 21st ACM Symposium on the Theory of Computing*, pp. 221-228, 1992.

[10] Scheffler, P., "A linear algorithm for the pathwidth of trees," in R. Bodendiek and R. Henn, eds., *Topics in Combinatorics and Graph Theory*, pp. 613-620, Physica-Verlag, Heidelberg, 1990.

[11] Takahashi, A., Ueno, S. and Kajitani, Y., "Minimal acyclic forbidden minors for the family of graphs with bounded path-width," *Discrete Mathematics*, vol. 127, pp. 293-304, 1994.

[12] Takahashi, A., Ueno, S. and Kajitani, Y., "Mixed-searching and proper-path-width," in W. L. Hsu and R. C. T. Lee, eds., *ISA '91 Algorithms: Proc. 2nd International Symposium on Algorithms*, pp. 61-71, Springer-Verlag, 1991. (LNCS 557).