SIAM J. COMPUT.
Vol. 35, No. 5, pp. 1023–1058

FINDING FOUR INDEPENDENT TREES*

SEAN CURRAN†, ORLANDO LEE‡, AND XINGXING YU§

Abstract. Motivated by a multitree approach to the design of reliable communication protocols, Itai and Rodeh gave a linear time algorithm for finding two independent spanning trees in a 2-connected graph. Cheriyan and Maheshwari gave an $O(|V|^2)$ algorithm for finding three independent spanning trees in a 3-connected graph. In this paper we present an $O(|V|^3)$ algorithm for finding four independent spanning trees in a 4-connected graph. We make use of chain decompositions of 4-connected graphs.

Key words. Connectivity, chain decomposition, numbering, independent trees, algorithm

AMS subject classifications. 05C40, 05C85, 05C38, 05C75

DOI. 10.1137/S0097539703436734

1. Introduction.

We consider simple graphs only. For a graph G, we use $V(G)$ and $E(G)$ to denote the vertex set and edge set of G, respectively.

For a tree T and $x, y \in V(T)$, let $T[x, y]$ denote the unique path from x to y in T. A *rooted tree* is a tree with a specified vertex called the *root* of T. Let G be a graph, let $r \in V(G)$, and let T and T' be trees of G rooted at r. We say that T and T' are *independent* if for every $x \in V(T) \cap V(T')$, the paths $T[r, x], T'[r, x]$ have no vertex in common except r and x.

The study of independent spanning trees started with Itai and Rodeh [11], where they proposed a multitree approach to reliability in distributed networks (see also [5]). They developed a linear time algorithm that, given any vertex r in a 2-connected graph G, finds two independent spanning trees of G rooted at r. Later, Cheriyan and Maheshwari [1] proved that for any vertex r in a 3-connected graph G, there exist three independent spanning trees of G rooted at r. Furthermore, they gave an $O(|V(G)|^2)$ algorithm for finding these trees.

Itai and Zehavi [12] also proved that every 3-connected graph contains three independent spanning trees (rooted at any vertex), and they conjectured that for any k-connected graph G and for any $r \in V(G)$, there exist k independent spanning trees of G rooted at r. According to Schrijver [14], the Itai-Zehavi conjecture is part of a more general conjecture by Frank [6]. Huck [9] proved this conjecture for planar 4-connected graphs. Later, Miura et al. [13] gave a linear time algorithm for finding four independent rooted spanning trees in a planar 4-connected graph.

Our main result is the following.

---

*Received by the editors October 28, 2003; accepted for publication (in revised form) August 15, 2005; published electronically March 3, 2006.
http://www.siam.org/journals/sicomp/35-5/43673.html

†School of Mathematics, Georgia Institute of Technology, Atlanta, GA 30332 (curran@math.gatech.edu). The work of this author was partially supported by an NSF VIGRE grant.

‡Instituto de Computação, Universidade Estadual de Campinas, Avenida Albert Einstein, 1251, Caixa Postal 6176, 13083–971 Campinas - SP, Brazil (lee@ic.unicamp.br). The work of this author was supported by CNPq (Proc: 200611/00-3)—Brazil.

§School of Mathematics, Georgia Institute of Technology, Atlanta, GA 30332 and Center of Combinatorics, LPMC, Nankai University, Tianjin 300071, China (yu@math.gatech.edu) The work of the third author was partially supported by NSF grant DMS-0245530 and by NSA grant MDA-904-03-1-0052.

THEOREM 1.1. *Let G be a 4-connected graph, and let $r \in V(G)$. Then there exist four independent spanning trees of G rooted at r. Moreover, such trees can be found in $O(|V(G)|^3)$ time.*

To provide motivation for our method, we first describe Itai and Rodeh's method for constructing two independent spanning trees rooted at a vertex r in a 2-connected graph. Let G be a 2-connected graph, and let r and t be two adjacent vertices of G. An *r-t numbering* is a function $g : V(G) \mapsto \{1, \ldots, n\}$ with $n \geq |V(G)|$ satisfying the following properties:

(i) $g(r) = 1$ and $g(t) = n$.

(ii) Every vertex $v \in V(G) - \{r, t\}$ has a neighbor u with $g(u) < g(v)$ and a neighbor w with $g(w) > g(v)$.

An r-t numbering can be produced from an ear decomposition of G. From an r-t numbering g, Itai and Rodeh define two independent spanning trees T_1 and T_2 of G rooted at r as follows. For each vertex $v \in V(G) - \{r\}$, specify its parent in each tree. In tree T_1, for each $v \in V(G) - \{r\}$, the parent of v is a neighbor u for which $g(u) < g(v)$. In tree T_2, the parent of t is r and, for each $v \in V(G) - \{r, t\}$, the parent of v is a neighbor w for which $g(w) > g(u)$. It is not hard to show that T_1 and T_2 are independent spanning trees in G rooted at r.

The idea for constructing four independent spanning trees in a 4-connected graph is inspired by the 2-connected case. The main difference is that we need to use two numberings instead of one. This idea can be roughly described as follows. Let G be a 4-connected graph, and let $r \in V(G)$. First, we compute a decomposition of G into "planar chains," a generalization of ear decomposition, which we describe in section 2. From this decomposition, we find two numberings g and f. We then construct these trees from these numberings.

The main difficulty with this idea lies in the fact that it is not possible to number all vertices of G, because the "chains" in our decomposition need not be paths. Fortunately, the nonpath part of the chains are planar, and we can compute four independent spanning trees in each one of these planar parts using the algorithm of Miura et al. [13] mentioned above. These trees are then used to number every vertex in the planar parts that has neighbors outside its chain. Once these numberings are computed, we can construct four independent spanning trees.

The rest of this paper is organized as follows. The remainder of this section is devoted to notation and terminology. In section 2 we describe chain decomposition of a graph and state the main decomposition result from [4] (also see [3]). In section 3 we describe known results for the planar case and give some auxiliary lemmas. In section 4 we give algorithms for constructing the required numberings. In section 5 we describe an algorithm for constructing four independent spanning trees in a 4-connected graph, and we prove its correctness in section 6.

Throughout this paper, we use $A := B$ to rename B as A or to define A as B. We use the notation xy (or yx) to represent an edge with ends x and y. Let G be a graph. For any $S \subseteq V(G)$, let $G[S]$ denote the subgraph of G with $V(G[S]) = S$ and $E(G[S])$ consisting of the edges of G with both ends in S; we say that $G[S]$ is the subgraph of G *induced* by S. Let $G - S := G[V(G) - S]$. A subgraph H of G is an *induced subgraph* of G if $G[V(H)] = H$. We also say that H is *induced* in G. For any $H \subseteq G$ and $S \subseteq V(G) \cup E(G)$, $H + S$ denotes the graph with vertex set $V(H) \cup (S \cap V(G))$ and edge set $E(H) \cup \{uv \in S : \{u, v\} \subseteq V(H) \cup (S \cap V(G))\}$.

A graph G is *k-connected*, where k is a positive integer, if $|V(G)| \geq k + 1$ and, for any $S \subseteq V(G)$ with $|S| \leq k - 1$, $G - S$ is connected. A subgraph H of G is *nonseparating* in G if $G - V(H)$ is connected.

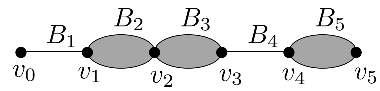

FIG. 1. *Example of a chain.*

.

Let G be a graph. For $S \subseteq V(G)$, let $N_G(S) := \{x \in V(G) - S : xy \in E(G)$ for some $y \in S\}$. For a subgraph H of G, let $N_G(H) := N_G(V(H))$. When $S = \{x\}$, we let $N_G(x) := N_G(\{x\})$. When there exists no ambiguity, we may simply use $N(S), N(H)$, and $N(x)$, instead of $N_G(S), N_G(H)$, and $N_G(x)$, respectively.

We describe a *path* in G as a sequence $P = (v_1, v_2, \ldots, v_k)$ of distinct vertices of G such that $v_i v_{i+1} \in E(G)$, $1 \le i \le k-1$. The vertices v_1 and v_k are called the *ends* of the path P, and the vertices in $V(P) - \{v_1, v_k\}$ are called the *internal vertices* of P. For $1 \le i \le j \le k$, let $P[v_i, v_j] := (v_i, \ldots, v_j)$, and for $1 \le i < j \le k$, let $P(v_i, v_j) := P[v_{i+1}, v_{j-1}]$. For $A, B \subseteq V(G)$, we say that a path P is an A-B path if one end of P is in A, the other end is in B, and no internal vertex of P is in $A \cup B$. If P is a path with ends a and b, we say that P is a *path from a to b*, or P is an *a-b* path. Two paths P and Q are *disjoint* if $V(P) \cap V(Q) = \emptyset$. Two paths are *internally disjoint* if no internal vertex of one path is contained in the other. Given a path P in G and a set $S \subseteq V(G)$ (respectively, a subgraph S of G), we say that P is *internally disjoint from S* if no internal vertex of P is contained in S (respectively, $V(S)$). We also describe a *cycle* in G as a sequence $C = (v_1, v_2, \ldots, v_k, v_1)$ such that the vertices $v_1, \ldots, v_k$ are distinct, $v_i v_{i+1} \in E(G)$, for $1 \le i \le k-1$, and $v_k v_1 \in E(G)$.

2. Chain decomposition. In order to prove Theorem 1.1, we rely on the existence of a *nonseparating chain decomposition* of a 4-connected graph, proved in [4] (also see [3]). Such a decomposition is similar to an ear decomposition. An *ear decomposition* of a graph G is a sequence $(P_0, P_1, \ldots, P_t)$ such that (i) P_0 is a cycle in G, (ii) $P_1, \ldots, P_t$ are paths in G, (iii) $\bigcup_{i=0}^{t} P_i = G$, and (iv) for each $0 \le i \le t-1$, $G_i := \bigcup_{j=0}^{i} P_j$ is 2-connected and $P_{i+1} \cap G_i$ consists of the ends of P_{i+1}. In a nonseparating chain decomposition, the P_i's will be chains and cycle chains, which may be thought of as a generalization of paths and cycles.

DEFINITION 2.1. *A connected graph H is a* chain *if its blocks can be labeled as $B_1, \ldots, B_k$, where $k \ge 1$ is an integer, and its cut vertices can be labeled as $v_1, \ldots, v_{k-1}$ such that*

(i) $V(B_i) \cap V(B_{i+1}) = \{v_i\}$ *for* $1 \le i \le k-1$, *and*
(ii) $V(B_i) \cap V(B_j) = \emptyset$ *if* $|i-j| \ge 2$ *and* $1 \le i, j \le k$.

We let $H := B_1 v_1 B_2 v_2 \ldots v_{k-1} B_k$ denote this situation. If $k \ge 2$, let $v_0 \in V(B_1) - \{v_1\}$ and $v_k \in V(B_k) - \{v_{k-1}\}$, or, if $k = 1$, let $v_0, v_k \in V(B_1)$ with $v_0 \ne v_k$; then we say that H is a v_0-v_k chain, *and we denote this by $H := v_0 B_1 v_1 \ldots v_{k-1} B_k v_k$. We usually fix v_0 and v_k, and we refer to them as the* ends *of H_i. See Figure* 1 *for an example with $k = 5$.*

DEFINITION 2.2. *A connected graph H is a* cyclic chain *if for some integer $k \ge 2$, there exist subgraphs $B_1, \ldots, B_k$ of H and vertices $v_1, \ldots, v_k$ of H such that*

(i) *for $1 \le i \le k$, B_i is 2-connected or B_i is induced by an edge of H,*
(ii) $V(H) = \bigcup_{i=1}^{k} V(B_i)$ *and* $E(H) = \bigcup_{i=1}^{k} E(B_i)$,
(iii) *if $k = 2$, then $V(B_1) \cap V(B_2) = \{v_1, v_2\}$ and $E(B_1) \cap E(B_2) = \emptyset$, and*
(iv) *if $k \ge 3$, then $V(B_i) \cap V(B_{i+1}) = \{v_i\}$ for $1 \le i \le k$, where $B_{k+1} := B_1$, and $V(B_i) \cap V(B_j) = \emptyset$ for $1 \le i < i+2 \le j \le k$ and $(i, j) \ne (1, k)$.*

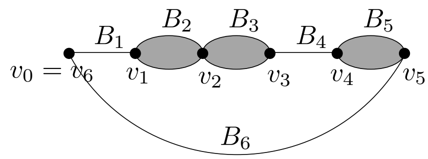

FIG. 2. *Example of a cyclic chain.*

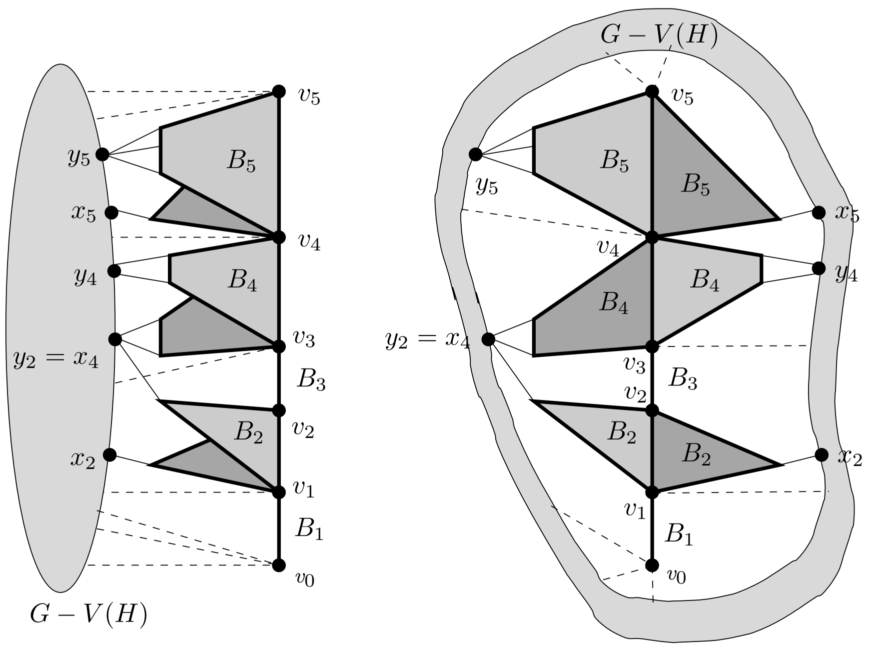

FIG. 3. *A planar chain $H := v_0B_1v_1B_2v_2B_3v_3B_4v_4B_5v_5$ in a graph G.*

For notational convenience, we usually fix one of the vertices $v_1, \ldots, v_k$ as the root *of H, say v_k, and write $H := v_0B_1v_1 \ldots v_{k-1}B_kv_k$ to indicate that H is a* cyclic chain rooted at v_0 *($= v_k$). Each subgraph B_i is called a piece of H. We sometimes write $I(H) := V(H)$. See Figure* 2 *for an example with $k = 6$.*

In the chain decompositions we will work with, the blocks and pieces have a planar structure. Let G be a graph with distinct vertices a, b, c, and d. We say that the quintuple (G, a, b, c, d) is *planar* if G can be drawn in a closed disc in the plane with no pair of edges crossing such that a, b, c, d occur in cyclic order on the boundary of the disc. For a graph G and $x, y \in V(G)$, we use $G - xy$ to denote the graph with vertex set $V(G)$ and edge set $E(G) - \{xy\}$ (note that xy need not be an edge of G).

DEFINITION 2.3. *Let G be a graph, and let $H := v_0B_1v_1 \ldots v_{k-1}B_kv_k$ be a chain (respectively, cyclic chain). If H is an induced subgraph of G, then we say that H is a* chain in G *(respectively,* cyclic chain in G*). We say that H is* planar in G *if, for each $1 \le i \le k$ with $|V(B_i)| \ge 3$ (or equivalently, B_i is 2-connected), there exist distinct vertices $x_i, y_i \in V(G) - V(H)$ such that $(G[V(B_i) \cup \{x_i, y_i\}] - x_iy_i, x_i, v_{i-1}, y_i, v_i)$ is planar, and $B_i - \{v_{i-1}, v_i\}$ is a component of $G - \{x_i, y_i, v_{i-1}, v_i\}$. We also say that H is a* planar v_0-v_k chain (respectively, planar cyclic chain). *See Figure* 3 *for two drawings of an example with $k = 5$. The dashed edges may or may not exist, but they are not part of H.*

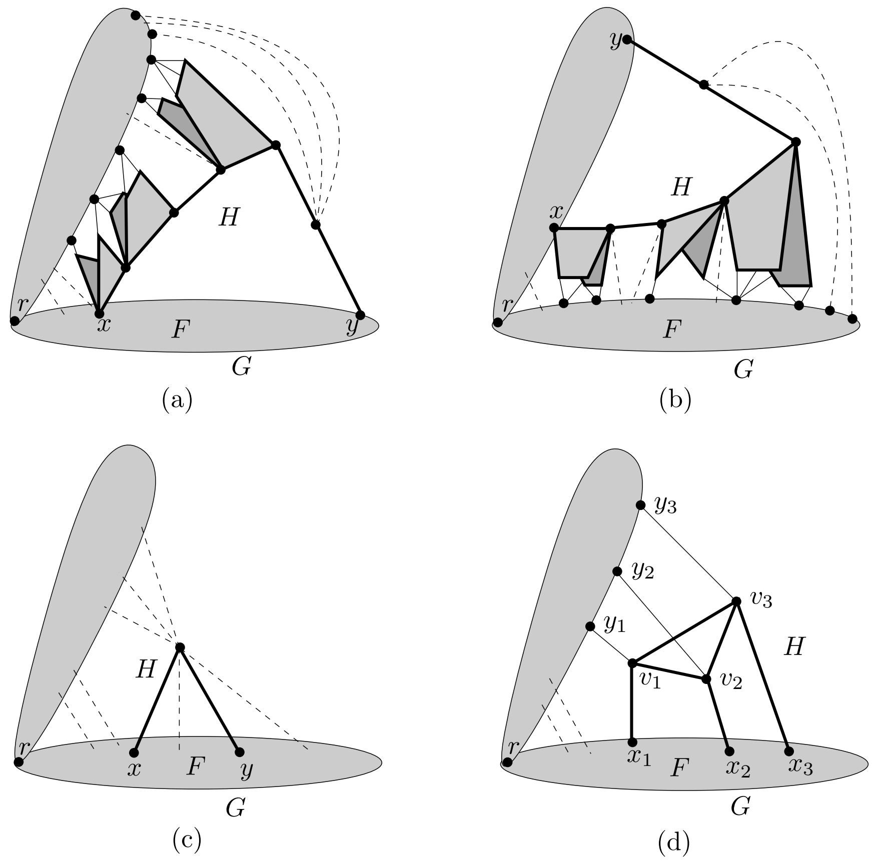

FIG. 4. (a) *An up F-chain,* (b) *a down F-chain,* (c) *an elementary F-chain, and* (d) *a triangle F-chain. The dashed edges need not exist.*

We can now describe the chains in nonseparating chain decompositions. See Figure 4 for illustrations.

DEFINITION 2.4. *Let G be a graph, let F be a subgraph of G, and let $r \in V(F)$. Let H be a planar x-y chain in G such that $V(H) - \{x, y\} \subseteq V(G) - V(F)$. We say that*

(i) *H is an up F-chain if $\{x, y\} \subseteq V(F)$ and $N_G(H - \{x, y\}) \subseteq (V(G) - V(F - r)) \cup \{x, y\}$,*

(ii) *H is a down F-chain if $\{x, y\} \subseteq V(G) - V(F - r)$ and $N_G(H - \{x, y\}) \subseteq V(F - r) \cup \{x, y\}$, and*

(iii) *H is an elementary F-chain if $\{x, y\} \subseteq V(F)$ and H is an x-y path of length two.*

In any of the three cases we say that H is a planar x-y F-chain in G (or simply a planar F-chain). Let $I(H) := V(H) - \{x, y\}$.

DEFINITION 2.5. *Let G be a graph, let F be a subgraph of G, and let $r \in V(F)$. Suppose that $\{v_1, v_2, v_3\} \subseteq V(G) - V(F)$ induces a triangle T in G and, for each $1 \leq i \leq 3$, v_i has exactly one neighbor x_i in $V(F - r)$ and exactly one neighbor y_i in $V(G) - (V(F) \cup V(T))$, and each v_i has degree four in G. Moreover, assume that x_1, x_2, x_3 are distinct and y_1, y_2, y_3 are distinct. Then we say that $H := T + \{x_1, x_2, x_3, v_1x_1, v_2x_2, v_3x_3\}$ is a triangle F-chain in G. We let $I(H) := \{v_1, v_2, v_3\}$.*

Note that Definitions 2.4 and 2.5 depend on the choice of r and F, but in spite of this, whenever we use these concepts in this paper, it should be clear which pair r, F we refer to.

DEFINITION 2.6. *Let G be a graph, let F be a subgraph of G, and let $r \in V(F)$. By a good F-chain in G, we mean an up F-chain or a down F-chain, or an elementary F-chain or a triangle F-chain.*

We are now ready to describe a chain decomposition, which is similar to an ear decomposition.

DEFINITION 2.7. *Let G be a graph, let $r \in V(G)$, and let $H_1, \ldots, H_t$ be chains in G, where $t \geq 2$. We say that $(H_1, \ldots, H_t)$ is a nonseparating chain decomposition of G rooted at r if the following conditions hold:*

(i) *H_1 is a planar cyclic chain in G rooted at r.*

(ii) *For each $i = 2, \ldots, t-1$, H_i is a good $G[\bigcup_{j=1}^{i-1} I(H_j)]$-chain in G.*

(iii) *$H_t := G - (\bigcup_{j=1}^{t-1} I(H_j) - \{r\})$ is a planar cyclic chain in G rooted at r.*

(iv) *For each $i = 1, \ldots, t-1$, both $G[\bigcup_{j=1}^{i} I(H_j)]$ and $G - ((\bigcup_{j=1}^{i} I(H_j)) - \{r\})$ are 2-connected.*

The chains $H_2, \ldots, H_{t-1}$ are called internal chains of the nonseparating chain decomposition. If ra is a piece of H_1, then we say that $H_1, \ldots, H_t$ is a nonseparating chain decomposition of G starting at ra.

The following result is proved in [4] (also see [3]).

THEOREM 2.8. *Let G be a 4-connected graph, let $r \in V(G)$, and let $ra \in E(G)$. Then G has a nonseparating chain decomposition rooted at r and starting at ra, and such a decomposition can be found in $O(|V(G)|^2|E(G)|)$ time.*

The basic idea for constructing four independent spanning trees (rooted at r) can be described as follows. By Theorem 2.8, G has a nonseparating chain decomposition $(H_1, \ldots, H_t)$ rooted at r. For $1 \leq i \leq t$, let $G_i := G\big[\bigcup_{j=1}^{i} I(H_j)\big]$. We compute two numberings g, f defined on $V(G)$ which resemble r-t numberings. From g we compute two independent spanning trees T_1, T_2 such that for each $i = 1, \ldots, t$, the restriction of T_1 and T_2 to G_i are independent spanning trees in G_i rooted at r. Similarly, from f we compute two spanning trees T_3, T_4 such that for each $i = 1, \ldots, t$, the restriction of T_3 and T_4 to $G - (V(G_i - r))$ are independent spanning trees rooted at r.

3. Planar graphs. Let G be a 4-connected graph, and let $r \in V(G)$. To use a nonseparating chain decomposition of G for constructing four independent spanning trees rooted at r, we must be able to find four independent spanning trees in the planar blocks and pieces. Unlike the original problem, these trees are not rooted at the same vertex, but they are rooted at four distinct vertices. Before we describe this result, we introduce some definitions.

DEFINITION 3.1. *Let T and T' be two trees in a graph G with roots r and r', respectively. We say that T and T' are independent if, for each $x \in V(T) \cap V(T')$, the paths $T[r,x]$ and $T'[r',x]$ have no vertex in common except x (and r if $r = r'$).*

Let G be a graph, and let $S := \{t_1, \ldots, t_4\}$ be a set of vertices of G. A 4-tuple $\mathcal{T} := \{T_1, \ldots, T_4\}$ is an S-system of G if, for $1 \leq i \leq 4$, T_i is a tree of G rooted at t_i, $V(T_i) \subseteq V(G) - (S - \{t_i\})$, and $t_i \in V(T_i)$. An S-system $\mathcal{T} := \{T_1, \ldots, T_4\}$ is independent if the trees in the system are pairwise independent, and an S-system $\mathcal{T}$ is spanning if $V(T_i) = V(G) - (S - \{t_i\})$ for $1 \leq i \leq 4$. See Figure 5 for an example, where the darkened edges are in the trees.

Let G be a graph, let $S \subseteq V(G)$, and let k be a positive integer. We say that G is *(k, S)-connected* if $|V(G)| \geq |S| + 1$, G is connected, and for any $T \subseteq V(G)$ with $|T| \leq k - 1$, every component of $G - T$ contains an element of S.

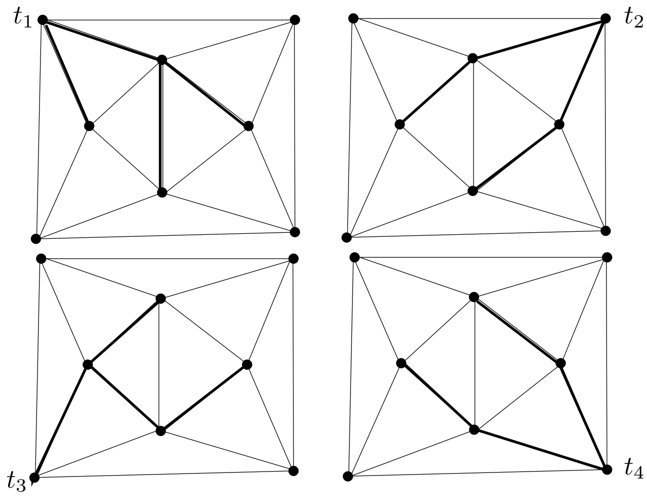

FIG. 5. *Four independent trees in a plane graph forming an independent spanning system.*

THEOREM 3.2. *Let (G, a, b, c, d) be a planar graph, and suppose that G is $(4, \{a, b, c, d\})$-connected. Then there exists an independent spanning $\{a, b, c, d\}$-system of G. Moreover, one can find such a system in linear time.*

The existence of an independent system in Theorem 3.2 was proved by Huck [9]. Huck's proof is not based on a decomposition of a planar graph, but through a careful analysis of his proof, one can extract an $O(|V(G)|^3)$ algorithm. Miura et al. [13] gave a linear algorithm for finding such a system based on a decomposition of 4-connected planar graphs. In fact, the decomposition they obtained can be viewed as a special case of a nonseparating chain decomposition.

Before we proceed, let us recall that the problem of finding an embedding of a planar graph can be solved in linear time [7, 8]. Moreover, the following problem can be solved in linear time: find a drawing of a planar quintuple (G, a, b, c, d) in a closed disc in the plane with no pair of edges crossing such that a, b, c, d occur in cyclic order on the boundary of the disc. We make no further mention of this fact, but it is implicitly used throughout this section.

In what follows we will use Theorem 3.2 to prove some results concerning "orderings" of certain vertices of a planar graph (G, a, b, c, d). These results correspond to Lemmas 3.4, 3.5, 3.6, and 3.7. They will be used in the next section to compute two numberings of subsets of $V(G)$.

DEFINITION 3.3. *Let (G, a, b, c, d) be a planar graph, and let $\{T_a, T_b, T_c, T_d\}$ be an independent spanning $\{a, b, c, d\}$-system of G, where T_v is rooted at v for each $v \in \{a, b, c, d\}$. Let $U \subseteq (N_G(b) \cup N_G(d)) - \{a, c\}$. We say that a permutation $u_1, \ldots, u_p$ of the elements of U is a (T_a, T_c)-ordering of U if, for $i, j \in \{1, \ldots, p\}$ with $i < j$, $T_a[a, u_i]$ and $T_c[c, u_j]$ are (vertex) disjoint. We also say that $u_1, \ldots, u_p$ is (T_a, T_c)-ordered.*

Our first lemma concerns (T_a, T_c)-orderings restricted to elements in $N_G(b) - \{a, c\}$. In this case, this ordering corresponds to a total order.

LEMMA 3.4. *Let (G, a, b, c, d) be a planar graph, and let $\{T_a, T_b, T_c, T_d\}$ be an independent spanning $\{a, b, c, d\}$-system of G, where T_v is rooted at v for each $v \in \{a, b, c, d\}$. Then there exists a unique (T_a, T_c)-ordering of $N_G(b) - \{a, c\}$. Moreover, such an ordering can be found in $O(|V(G)|)$ time.*

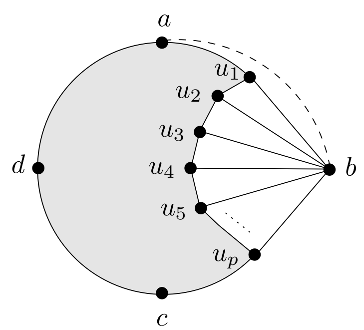

FIG. 6. $u_1, \ldots, u_p$ *is the unique* (T_a, T_c)*-ordering of* $N_G(b) - \{a, c\}$.

Proof. Let $G' := G - \{ab, bc\}$. If b has at most one neighbor in G', then the result follows immediately. So assume b has at least two neighbors in G'. Take an embedding of G' in a closed disc such that a, b, c, d occur in clockwise order on the boundary of the disc (such an embedding for G can be computed in linear time). Let $u_1, \ldots, u_p$ ($p \geq 2$) denote the neighbors of b in G' in that cyclic order around b such that a, u_1, b, u_p, c, d occur in clockwise order on the boundary of the disc (see Figure 6). Since T_a, T_c are independent, we have that for each $i \in \{1, \ldots, p\}$, $T_a[a, u_i]$ and $T_c[c, u_i]$ are internally disjoint. Then by planarity one can see that, for every $i, j \in \{1, \ldots, p\}$ with $i \neq j$, $T_a[a, u_i]$ and $T_c[c, u_j]$ are disjoint if and only if $i < j$. Thus, $u_1, \ldots, u_p$ is the unique (T_a, T_c)-ordering of $N_G(b) - \{a, c\}$. Clearly, such an ordering can be computed in $O(|V(G)|)$ time. ∎

In the next lemma we show that it is possible to extend a (T_a, T_c)-ordering of $N_G(b) - \{a, c\}$ and a (T_a, T_c)-ordering of $N_G(d) - \{a, c\}$ to a (T_a, T_c)-ordering of $(N_G(b) \cup N_G(d)) - \{a, c\}$.

LEMMA 3.5. *Let* (G, a, b, c, d) *be a planar graph, and let* $\{T_a, T_b, T_c, T_d\}$ *be an independent spanning* $\{a, b, c, d\}$*-system of* G*, where* T_v *is rooted at* v *for each* $v \in \{a, b, c, d\}$*. Then there exists a* (T_a, T_c)*-ordering of* $(N_G(b) \cup N_G(d)) - \{a, c\}$*. Moreover, such an ordering can be found in* $O(|V(G)|^2)$ *time.*

Proof. Take an embedding of G in a closed disc such that a, b, c, d occur in clockwise order on the boundary of the disc. Consider the following relation. For $u, v \in (N_G(b) \cup N_G(d)) - \{a, c\}$, we say that $u \prec v$ if either one of the following holds:

(i) $u \in N_G(b)$ and $T_a[a, u] \cap T_c[c, v] = \emptyset$.

(ii) $u \in N_G(d)$, $T_a[a, u] \cap T_c[c, v] = \emptyset$, and $T_a[a, v] \cap T_c[c, u] \neq \emptyset$.

See Figure 7 for an illustration of conditions (i) and (ii). The bold lines denote the paths in T_a and the dashed lines denote the paths in T_c. Next, we show that $\prec$ defines a total order on $(N_G(b) \cup N_G(d)) - \{a, c\}$.

First, we show that for any distinct $x, y \in (N_G(b) \cup N_G(d)) - \{a, c\}$, either $x \prec y$, or $y \prec x$, but not both. If $x, y \in N_G(b)$ or $x, y \in N_G(d)$, then by planarity, either $T_a[a, x] \cap T_c[c, y] = \emptyset$ and $T_a[a, y] \cap T_c[c, x] \neq \emptyset$, or $T_a[a, x] \cap T_c[c, y] \neq \emptyset$ and $T_a[a, y] \cap T_c[c, x] = \emptyset$. So by (i) or (ii), either $x \prec y$, or $y \prec x$, but not both. Thus, we may assume that $x \in N_G(b)$ and $y \in N_G(d)$. If $T_a[a, x] \cap T_c[c, y] = \emptyset$, then x, y satisfy (i) (as u, v) but not (ii) (as v, u), and we have $x \prec y$ and $y \not\prec x$. So assume $T_a[a, x] \cap T_c[c, y] \neq \emptyset$. Then x, y does not satisfy (i) (as u, v), and hence, $x \not\prec y$. Since T_a and T_c are independent, $T_a[a, y]$ and $T_c[c, y]$ are internally disjoint, and $T_a[a, x]$ and $T_c[c, x]$ are internally disjoint. By planarity, $T_a[a, y] \cap T_c[c, x] = \emptyset$. Therefore, $y \prec x$.

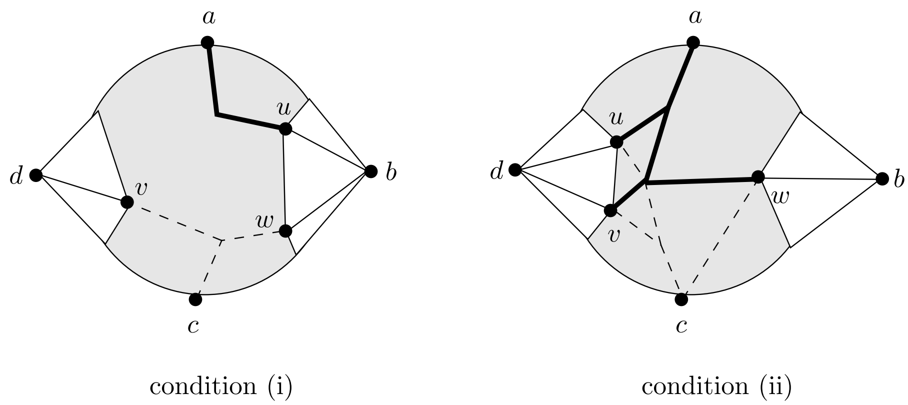

FIG. 7. $u \prec v$ and $u \prec w$.

It remains to show that $\prec$ is transitive. Let $x, y, z \in (N_G(b) \cup N_G(d)) - \{a, c\}$, and assume that $x \prec y$ and $y \prec z$. We will show that $x \prec z$. We have eight cases by considering which of x, y, z are in $N_G(b)$.

(1) $x, y, z \in N_G(b)$. Since $x \prec y$ and $y \prec z$, it follows from (i) that $T_a[a, x] \cap T_c[c, y] = \emptyset$ and $T_a[a, y] \cap T_c[c, z] = \emptyset$. So by planarity, $T_a[a, x] \cap T_c[c, z] = \emptyset$, and by (i), $x \prec z$.

(2) $x, y, z \in N_G(d)$. Since $x \prec y$ and $y \prec z$, it follows from (ii) that $T_a[a, x] \cap T_c[c, y] = \emptyset$, $T_a[a, y] \cap T_c[c, x] \neq \emptyset$, $T_a[a, y] \cap T_c[c, z] = \emptyset$, and $T_a[a, z] \cap T_c[c, y] \neq \emptyset$. So by planarity, $T_a[a, x] \cap T_c[c, z] = \emptyset$ and $T_a[a, z] \cap T_c[c, x] \neq \emptyset$. By (ii), $x \prec z$.

(3) $y, z \in N_G(b)$ and $x \in N_G(d)$. Since T_a and T_c are independent, $P := T_a[a, y] \cup T_c[c, y]$ is an a-c path in $G - \{b, d\}$. Note that P divides the disc into two closed regions, say B and D, with b in B and d in D. Since $x \prec y$ and $x \in N_G(d)$, it follows from (ii) that $T_a[a, x] \cap T_c[c, y] = \emptyset$ and $T_a[a, y] \cap T_c[c, x] \neq \emptyset$. Since $y \prec z$ and $y \in N_G(b)$, it follows from (i) that $T_a[a, y] \cap T_c[c, z] = \emptyset$. So $T_c[c, z]$ lies in B and $T_a[a, x]$ lies in D. Therefore, by planarity, $T_a[a, x] \cap T_c[c, z] = \emptyset$. Since $T_a[a, y] \cap T_c[c, x] \neq \emptyset$, it follows by planarity that $T_a[a, z] \cap T_c[c, x] \neq \emptyset$. Therefore, $x \prec z$.

(4) $y, z \in N_G(d)$ and $x \in N_G(b)$. Since T_a and T_c are independent, $P := T_a[a, y] \cup T_c[c, y]$ is an a-c path in $G - \{b, d\}$, and P divides the disc into two closed regions B and D, with b in B and d in D. Since $x \prec y$ and $x \in N_G(b)$, it follows from (i) that $T_a[a, x] \cap T_c[c, y] = \emptyset$, and since $y \prec z$ and $y \in N_G(d)$, it follows from (ii) that $T_a[a, y] \cap T_c[c, z] = \emptyset$. So $T_c[c, z]$ lies in D and $T_a[a, x]$ lies in B. By planarity, $T_a[a, x] \cap T_c[c, z] = \emptyset$, and hence by (i), $x \prec z$.

(5) $x, y \in N_G(d)$ and $z \in N_G(b)$. Since T_a and T_c are independent, $P := T_a[a, y] \cup T_c[c, y]$ is an a-c path in $G - \{b, d\}$, and P divides the disc into two closed regions B and D, with b in B and d in D. Since $x \prec y$ and $x \in N_G(d)$, it follows from (ii) that $T_a[a, x] \cap T_c[c, y] = \emptyset$, and since $y \prec z$ and $y \in N_G(d)$, it follows from (ii) that $T_a[a, y] \cap T_c[c, z] = \emptyset$. Thus, $T_a[a, x]$ lies in D and $T_c[c, z]$ lies in B. By planarity, $T_a[a, x] \cap T_c[c, z] = \emptyset$. Moreover, since $y \prec z$ and $y \in N_G(d)$, it follows from (ii) that $T_a[a, z] \cap T_c[c, y] \neq \emptyset$. By planarity, $T_a[a, z] \cap T_c[c, x] \neq \emptyset$. Therefore by (ii), $x \prec z$.

(6) $x, y \in N_G(b)$ and $z \in N_G(d)$. Since T_a and T_c are independent, $P := T_a[a, y] \cup T_c[c, y]$ is an a-c path in $G - \{b, d\}$, and P divides the disc into two closed

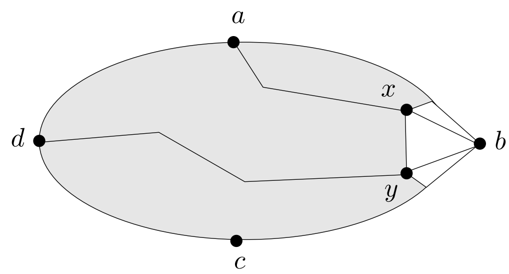

FIG. 8. *Two disjoint paths, one in T_a and the other in T_d.*

regions B and D, with b in B and d in D. Since $x \prec y$ and $x \in N_G(b)$, it follows from (i) that $T_a[a,x] \cap T_c[c,y] = \emptyset$, and since $y \prec z$ and $y \in N_G(b)$, it follows from (i) that $T_a[a,y] \cap T_c[c,z] = \emptyset$. So $T_a[a,x]$ lies in B and $T_c[c,z]$ lies in D. By planarity $T_a[a,x] \cap T_c[c,z] = \emptyset$. Therefore by (i), $x \prec z$.

(7) $x, z \in N_G(b)$ and $y \in N_G(d)$. We have shown that either $x \prec z$ or $z \prec x$. Suppose for a contradiction that $z \prec x$. Then by (i), $T_a[a,z] \cap T_c[c,x] = \emptyset$, and by planarity, $T_a[a,x] \cap T_c[c,z] \neq \emptyset$. Since $y \prec z$ and $y \in N_G(d)$, $T_a[a,z] \cap T_c[c,y] \neq \emptyset$. But then, by planarity, $T_a[a,x] \cap T_c[c,y] \neq \emptyset$, which is a contradiction to (i) since $x \prec y$ and $x \in N_G(b)$. Therefore, $x \prec z$.

(8) $x, z \in N_G(d)$ and $y \in N_G(b)$. We have shown that either $x \prec z$ or $z \prec x$. Suppose for a contradiction that $z \prec x$. Then by (ii), $T_a[a,z] \cap T_c[c,x] = \emptyset$ and $T_a[a,x] \cap T_c[c,z] \neq \emptyset$. Since $x \prec y$ and $x \in N_G(d)$, $T_a[a,y] \cap T_c[c,x] \neq \emptyset$. But then, by planarity, $T_a[a,y] \cap T_c[c,z] \neq \emptyset$, which is a contradiction to (i) since $y \prec z$ and $y \in N_G(b)$. Therefore, $x \prec z$.

Thus, $\prec$ defines a total order on $(N_G(b) \cup N_G(d)) - \{a,c\}$. Hence, the required (T_a, T_c)-ordering exists.

Furthermore, this ordering can be found as follows. Let $b_1, \ldots, b_p$ be the (T_a, T_c)-ordering of $N_G(b) - \{a,c\}$, and let $d_1, \ldots, d_q$ be the (T_a, T_c)-ordering of $N_G(d) - \{a,c\}$. Both exist by Lemma 3.4. Theses sequences are ordered under $\prec$. We can decide in $O(|V(G)|)$ time whether $b_i \prec d_j$ or $d_j \prec b_i$ (by checking which of (i) or (ii) holds) for any pair b_i, d_j, $1 \leq i \leq p$, $1 \leq j \leq q$. Thus, using the so-called merge technique in [2], we can merge the two sequences to obtain a sequence ordered under $\prec$ in $O(|V(G)|^2)$ time. ∎

The last two lemmas of this section will also be needed in section 5. Figure 8 illustrates Lemma 3.6, and Figure 9 illustrates Lemma 3.7.

LEMMA 3.6. *Let (G, a, b, c, d) be a planar graph, and let $\{T_a, T_b, T_c, T_d\}$ be an independent spanning $\{a,b,c,d\}$-system of G, where T_v is rooted at v for each $v \in \{a,b,c,d\}$. Assume that b has at least two neighbors in $V(G) - \{a,c\}$. Then for any (T_a, T_c)-ordered pair $x, y \in N_G(b) - \{a,c\}$, $T_a[a,x] \cap T_d[d,y] = \emptyset$.*

Proof. Take an embedding of G in a disc such that a, b, c, d occur in clockwise order on the boundary of the disc. Let $x, y \in N_G(b) - \{a,c\}$ such that x, y is (T_a, T_c)-ordered (see Figure 8). Hence, $T_a[a,x] \cap T_c[c,y] = \emptyset$. Since T_a and T_d are independent, $P := T_a[a,x] \cup T_d[d,x]$ is an a-d path in $G - \{a,c\}$, and P divides the disc into two closed regions B and C, with b in B. By planarity and since $T_a[a,x] \cap T_c[c,y] = \emptyset$, $T_d[d,y]$ lies in B. Then by planarity, $T_a[a,x] \cap T_d[d,y] = \emptyset$. ∎

LEMMA 3.7. *Let (G, a, b, c, d) be a planar graph, and let $\{T_a, T_b, T_c, T_d\}$ be an independent spanning $\{a,b,c,d\}$-system of G, where T_v is rooted at v for each $v \in$*

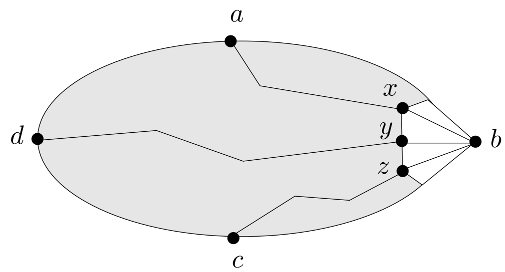

FIG. 9. *Three disjoint paths contained in T_a, T_c, and T_d, respectively.*

$\{a, b, c, d\}$. *Assume that b has at least three neighbors in $V(G) - \{a, c\}$. Then for any (T_a, T_c)-ordered triple $x, y, z \in N_G(b) - \{a, c\}$, $T_a[a, x], T_d[d, y]$, and $T_c[c, z]$ are pairwise disjoint.*

Proof. Take an embedding of G in a disc such that a, b, c, d occur in clockwise order on the boundary of the disc. Let $x, y, z \in N_G(b)$ such that x, y, z is (T_a, T_c)-ordered (see Figure 9). Hence, $T_a[a, x] \cap T_c[c, y] = \emptyset$ and $T_a[a, y] \cap T_c[c, z] = \emptyset$.

By Lemma 3.4, $T_a[a, x] \cap T_c[c, z] = \emptyset$. Hence, the path $P := T_d[d, y] + \{b, yb\}$ divides the disc into closed regions A and C, with $T_a[a, x]$ in A and $T_c[c, z]$ in C. By Lemma 3.6, $T_a[a, x] \cap T_d[d, y] = \emptyset$. By applying a mirror image version of Lemma 3.6, we can show that $T_d[d, y] \cap T_c[c, z] = \emptyset$. ∎

4. Numberings. By Theorem 2.8, G has a nonseparating chain decomposition rooted at r. In this section we will combine this decomposition with Theorem 3.2 to produce a numbering of a subset of $V(G)$. This numbering will be used in the next section to construct four independent spanning trees rooted at r.

In the rest of this section we fix the following notation.

Notation 4.1. Let G be a 4-connected graph, and let $r \in V(G)$. Fix a nonseparating chain decomposition of G rooted at r, say $\mathcal{C} := (H_1, \dots, H_t)$, $t \geq 2$. Define the sequences $G_0, G_1, \dots, G_{t-1}$ and $\bar{G}_1, \dots, \bar{G}_t$ as follows:

(i) $G_0 := \bar{G}_t := (\{r\}, \emptyset)$.

(ii) *For* $i = 1, \dots, t-1$, $G_i := G[\bigcup_{j=1}^{i} I(H_j)]$ *and* $\bar{G}_i := G - (V(G_i) - \{r\})$.

Notation 4.2. Suppose that H_i $(1 \leq i \leq t)$ is an up G_{i-1}-chain in G or a down G_{i-1}-chain in G. Let $H_i := v_0 B_1 v_1 B_2 v_2 \dots v_{k-1} B_k v_k$. For each 2-connected B_j there exist u_j, w_j (both on $V(G_{i-1})$ or both on $V(\bar{G}_i)$) such that $B_j - \{v_{j-1}, v_j\}$ is a component of $G - \{v_{j-1}, v_j, u_j, w_j\}$, and $(B_j^+, v_{j-1}, u_j, v_j, w_j)$ is planar, where $B_j^+ := G[V(B_j) \cup \{u_j, w_j\}] - u_j w_j$. We refer to each such B_j^+ as a *planar section* in $\mathcal{C}$. The vertices v_{j-1}, v_j, u_j, w_j are the *terminals* of B_j^+. See Figure 10 for an illustration. Note that the notation above depends on i. For the sake of clarity we will not make it explicit in the notation, but whenever we use this we will make clear which i we refer to. Furthermore, the algorithms we will describe deal with each H_i separately, and thus no confusion should arise.

DEFINITION 4.3. *Suppose that H_i $(1 \leq i \leq t)$ is a triangle G_{i-1}-chain in G. See Figure 4. Let $I(H_i) := \{v_1, v_2, v_3\}$, let $V(H_i) - I(H_i) := \{x_1, x_2, x_3\}$, and suppose that $x_j v_j \in E(G)$ for $j = 1, 2, 3$. We say that $v_j x_j$ $(j = 1, 2, 3)$ are the legs of H_i.*

DEFINITION 4.4. *Let $D \subseteq V(G)$. A numbering of D is a function from D to $\{1, \dots, |D|\}$. Let g be a numbering of D, let $v_1, \dots, v_k$ be a sequence of distinct vertices in $V(G) - D$, and let $v_0 \in D$. The extension g' of g to $v_1, \dots, v_k$ from v_0 is defined as follows:*

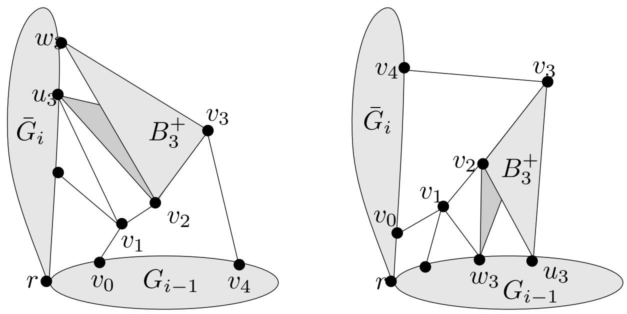

FIG. 10. *v_2, v_3, u_3, w_3 are the terminals of B_3^+.*

(i) *for $1 \le i \le k$, let $g'(v_i) := g(v_0) + i$;*
(ii) *for each $v \in D$ with $g(v) \le g(v_0)$ let $g'(v) = g(v)$; and*
(iii) *for each $v \in D$ with $g(v) > g(v_0)$ let $g'(v) := g(v) + k$.*

Note that g' is a numbering of $D \cup \{v_1, \dots, v_k\}$. For convenience, if $D \subseteq V(G)$ and σ denotes a sequence $v_1, \dots, v_k$ of vertices in $V(G) - D$, we let $D \cup \{\sigma\} := D \cup \{v_1, \dots, v_k\}$.

In order to compute the desired numberings g and f from a nonseparating chain decomposition, we need to find independent spanning systems in the planar sections in $\mathcal{C}$.

Assumption 4.5. For each planar section B_j^+ in $\mathcal{C}$, with terminals v_{j-1}, v_j, u_j, w_j, we compute an independent spanning $\{v_{j-1}, v_j, u_j, w_j\}$-system of B_j^+. By Theorem 3.2, such a system can be computed in $O(|V(B_j^+)| + |E(B_j^+)|)$ time. Since two distinct planar sections are edge-disjoint, the overall time consumed in this phase (for all planar sections in $\mathcal{C}$) is $O(|V(G)| + |E(G)|)$.

Next, we describe the algorithm for computing a numbering g of a subset of $V(G)$. It also computes a sequence $\{r\} = D_0 \subset D_1 \subset \cdots \subset D_{t-1}$ of subsets of $V(G)$ such that for $i = 1, \dots, t$, $N_G(H_i) \cap V(G_{i-1}) \subseteq D_{i-1}$. When the algorithm stops, g is a numbering of D_{t-1}. We note that keeping track of this sequence is not necessary for computing g, but its inclusion will make proofs easier in section 6.

ALGORITHM NUMBERING g.

Description. The algorithm executes $t-1$ iterations, where t is the number of chains in $\mathcal{C}$. At the beginning of the first iteration, we have $i = 1$, $D_0 := \{r\}$, and $g(r) := 1$. At the beginning of each iteration, we have an integer i with $1 \le i \le t-1$, a subset $D_{i-1} \subseteq V(G_{i-1})$ such that $N_G(H_i) \cap V(G_{i-1}) \subseteq D_{i-1}$, and a numbering g of D_{i-1}.

Each iteration consists of the following: update g and define D_i according to the following cases (depending on the type of H_i), and, if $i < t-1$, then set $i \leftarrow i+1$ and start a new iteration.

Case 1. H_i is an elementary G_{i-1}-chain in G.

Let $H_i := v_0 B_1 v_1 B_2 v_2$, and assume that v_0, v_2 are labeled so that $g(v_0) < g(v_2)$. Extend g to v_1 from v_0, and let $D_i := D_{i-1} \cup \{v_1\}$.

Case 2. $i = 1$, or H_i is an up G_{i-1}-chain in G but not an elementary G_{i-1}-chain.

Let $H_i := v_0 B_1 v_1 \dots v_{k-1} B_k v_k$, and suppose that $v_0, \dots, v_k$ and $B_1, \dots, B_k$ are labeled so that $v_0 = v_k = r$ when $i = 1$ and $g(v_0) < g(v_k)$ when $i \ne 1$. For each 2-connected B_j, let u_j, w_j denote the terminals of B_j^+ other than v_{j-1}, v_j. Let $T^j_{v_{j-1}}, T^j_{v_j}$ denote the trees rooted, respectively, at v_{j-1}, v_j in the independent spanning $\{v_{j-1}, v_j, u_j, w_j\}$-system of B_j^+ computed in Assumption 4.5.

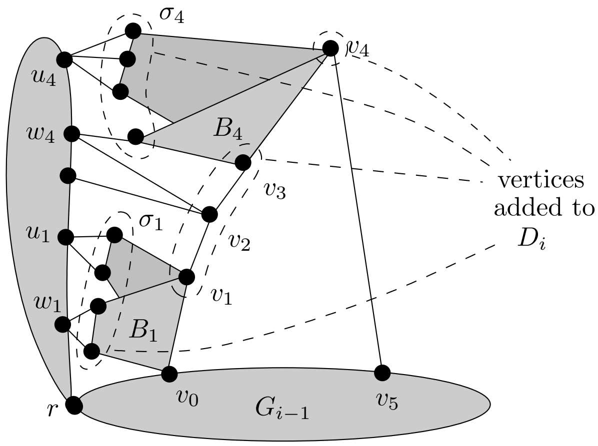

FIG. 11. *Extending the numbering g to an up G_{i-1}-chain.*

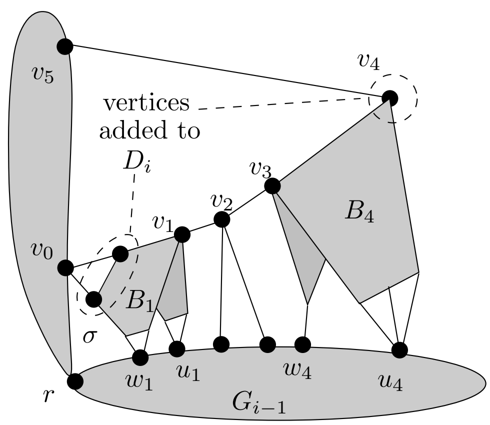

FIG. 12. *Extending the numbering g to a down G_{i-1}-chain.*

For each $j = 1, \ldots, k$, compute a sequence σ_j as follows. If B_j is 2-connected, then let σ_j be a $(T^j_{v_{j-1}}, T^j_{v_j})$-ordering of $N_{B_j^+}(\{u_j, w_j\}) - \{v_{j-1}, v_j\}$ (the existence of this ordering is guaranteed by Lemma 3.5). If B_j is trivial, then let σ_j denote the empty sequence.

Extend g to $\sigma := \sigma_1, v_1, \sigma_2, v_2, \ldots, v_{k-1}, \sigma_k$ from v_0, and let $D_i := D_{i-1} \cup \{\sigma\}$. See Figure 11 for an illustration.

Case 3. H_i is a down G_{i-1}-chain in G but not an elementary G_{i-1}-chain.

Let $H_i := v_0 B_1 v_1 \ldots v_{k-1} B_k v_k$. For each 2-connected block B_j let u_j, w_j denote the terminals of B_j^+ other than v_{j-1}, v_j with $g(u_j) < g(w_j)$. Let $T^j_{u_j}, T^j_{w_j}$ denote trees rooted, respectively, at u_j, w_j in the independent spanning $\{v_{j-1}, v_j, u_j, w_j\}$-system of B_j^+ computed in Assumption 4.5.

Let $D_i := D_{i-1} \cup N_{B_1}(v_0) \cup N_{B_k}(v_k)$. See Figure 12 for an illustration. Extend g according to the following three subcases.

Subcase 3.1. $k = 1$ (thus, B_1 is 2-connected).

Let σ denote a $(T^1_{u_1}, T^1_{w_1})$-ordering of $N_{B_1^+}(\{v_0, v_1\}) - \{u_1, w_1\} = N_{B_1}(v_0) \cup N_{B_k}(v_k)$ (the existence of this ordering is guaranteed by Lemma 3.5). Extend g to σ from u_1.

Subcase 3.2. $k = 2$, and B_1 or B_2 is trivial.

Note that since H_i is not an elementary G_{i-1}-chain, B_1 or B_2 is nontrivial.

Assume then (renaming B_1 and B_2 if necessary) that B_1 is 2-connected and B_2 is trivial. Extend g according to the following cases.

(i) v_1 has no neighbor in $V(G_{i-1})$. Let q_1, q_2, q_3 be neighbors of v_1 in B_1 (they exist since G is 4-connected), and assume that q_1, q_2, q_3 is $(T^1_{u_1}, T^1_{w_1})$-ordered (this is possible by Lemma 3.4). By Lemma 3.7, $T^1_{u_1}[u_1, q_1], T^1_{v_0}[v_0, q_2]$, and $T^1_{w_1}[w_1, q_3]$ are disjoint. Let $H := B_1^+ \cup B_2$ (note that $H - \{v_0, v_2, u_1, w_1\}$ is a component of $G - \{v_0, v_2, u_1, w_1\}$). Then (H, v_0, u_1, v_2, w_1) is planar, and $\{T^1_{v_0} + \{v_1, v_1q_2\}, T^1_{v_1} + \{v_2, v_1v_2\}, T^1_{u_1} + \{v_1, v_1q_1\}, T^1_{w_1} + \{v_1, v_1q_3\}\}$ forms an independent spanning $\{v_0, u_1, v_2, w_1\}$-system of H.
Let σ denote a $(T^1_{u_1} + \{v_1, v_1q_1\}, T^1_{w_1} + \{v_1, v_1q_3\})$-ordering of $N_H(\{v_0, v_2\}) - \{u_1, w_1\}$ (the existence of this ordering is guaranteed by Lemma 3.5). Extend g to σ from u_1.
Comment: we also keep track of q_1, q_2, q_3 for the construction of the independent spanning trees.

- v_1 has a neighbor in $V(G_{i-1})$. Let $x \in N_G(v_1) \cap V(G_{i-1})$ with $g(x)$ minimum, and let σ denote a $(T^1_{u_1}, T^1_{w_1})$-ordering of $N_{B_1^+}(v_0) - \{u_1, w_1\}$ (the existence of this ordering is guaranteed by Lemma 3.4). If $g(x) > g(u_1)$, then extend g to σ, v_1 from u_1, where σ, v_1 is the sequence obtained from σ by adding v_1 at the end. If $g(x) \le g(u_1)$, then extend g to v_1, σ from x, where v_1, σ is the sequence obtained from σ by adding v_1 in the front.

Subcase 3.3. $k \ge 3$, or $k = 2$ and both B_1, B_2 are 2-connected.

Extend g to $N_{B_1}(v_0)$ according to the following cases.

- B_1 is 2-connected. Let σ denote a $(T^1_{u_1}, T^1_{w_1})$-ordering of $N_{B_1^+}(v_0) - \{u_1, w_1\} = N_{B_1}(v_0)$ (the existence of this ordering is guaranteed by Lemma 3.4). Extend g to σ from u_1.
- Both B_1 and B_2 are trivial. Let $x \in N_G(v_1) \cap V(G_{i-1})$ with $g(x)$ minimum. Extend g to v_1 from x.
- B_1 is trivial and B_2 is 2-connected.
  - If v_1 has no neighbor in $V(G_{i-1})$, extend g to v_1 from u_2.
  - If v_1 has a neighbor in $V(G_{i-1})$, let $x \in N_G(v_1) \cap V(G_{i-1})$ with $g(x)$ minimum. If $g(x) > g(u_2)$, then extend g to v_1 from u_2. If $g(x) \le g(u_2)$, then extend g to v_1 from x.

Extend (the resulting) g to $N_{B_k}(v_k)$ according to the following cases.

- B_k is 2-connected. Let σ be a $(T^k_{u_k}, T^k_{w_k})$-ordering of $N_{B_k^+}(v_k) - \{u_k, w_k\} = N_{B_k}(v_k)$ (the existence of this ordering is guaranteed by Lemma 3.4). Extend g to σ from u_k.
- Both B_k and B_{k-1} are trivial. Let $x \in N_G(v_{k-1}) \cap V(G_{i-1})$ with $g(x)$ minimum. Extend g to v_{k-1} from x.
- B_k is trivial and B_{k-1} is 2-connected.
  - If v_{k-1} has no neighbor in $V(G_{i-1})$, extend g to v_{k-1} from u_{k-1}.
  - If v_{k-1} has a neighbor in $V(G_{i-1})$, let $x \in N_G(v_{k-1}) \cap V(G_{i-1})$ with $g(x)$ minimum. If $g(x) > g(u_{k-1})$, then extend g to v_{k-1} from u_{k-1}. If $g(x) \le g(u_{k-1})$, then extend g to v_{k-1} from x.

Case 4. H_i is a triangle G_{i-1}-chain in G.

Let $I(H_i) := \{v_1, v_2, v_3\}$, and let $v_j x_j$ $(j = 1, 2, 3)$ be the legs of H_i. Suppose that v_1, v_2, v_3 are labeled so that $g(x_1) < g(x_2) < g(x_3)$. Let $D_i := D_{i-1} \cup \{v_1, v_2, v_3\}$. Extend g to v_1, v_2, v_3 from x_2.

This concludes the description of the algorithm for computing g.

LEMMA 4.6. *Algorithm Numbering g runs in $O(|V(G)|^3)$ time.*

Proof. Observe that at the ith iteration, Algorithm Numbering g extends the current numbering g^i to a sequence σ from a previously numbered vertex $v \in D_{i-1}$. Clearly, given g^i, σ, and v, this extension can be computed in $O(|V(G)|)$ time. We now analyze the time spent at each iteration of Algorithm Numbering g according to Cases 1–4. We use the same notation as in the algorithm.

If Case 1 occurs (H_i is an elementary G_{i-1}-chain in G), then Algorithm Numbering g extends g to v_1. This can be done in $O(|V(G)|)$ time.

If Case 2 occurs (H_i is an up G_{i-1}-chain but not an elementary G_{i-1}-chain), then Algorithm Numbering g computes sequences $\sigma_1, \ldots, \sigma_k$, where σ_j denotes the empty sequence when B_j is trivial, and σ_j is a $(T^j_{v_{j-1}}, T^j_{v_j})$-ordering of $N_{B_j^+}(\{u_j, w_j\}) - \{v_{j-1}, v_j\}$ when B_j is 2-connected. In the latter case, by Lemma 3.5 the sequence σ_j can be computed in $O(|V(B_j^+)|^2)$ time. Thus, the algorithm spends $O(|V(G)|^2)$ time to compute $\sigma_1, \ldots, \sigma_k$. After that, the algorithm extends g to $v_0, \sigma_1, v_1, \ldots, v_{k-1}, \sigma_k, v_k$, which can be done in $O(|V(G)|)$ time. Therefore, the algorithm spends $O(|V(G)|^2)$ time if Case 2 occurs.

If Case 3 occurs (H_i is a down G_{i-1}-chain but not an elementary G_{i-1}-chain), then Algorithm Numbering g considers three cases.

- If Subcase 3.1 occurs ($k = 1$), then the algorithm computes a $(T^1_{u_1}, T^1_{w_1})$-ordering σ of $N_{B_1^+}(\{v_0, v_1\}) - \{u_1, w_1\}$ and extends g to σ from u_1. The sequence σ can be computed in $O(|V(G)|^2)$ time by Lemma 3.5, and the extension of g can be computed in $O(|V(G)|)$ time, resulting in $O(|V(G)|^2)$ time for this iteration.
- If Subcase 3.2 occurs ($k = 2$, and B_1 or B_2 is trivial), then the algorithm considers two subcases, according to whether or not v_1 has a neighbor in $V(G_{i-1})$.
  - If v_1 has no neighbor in $V(G_{i-1})$, the algorithm chooses neighbors q_1, q_2, q_3 of v_1 in B_1 and computes a $(T^1_{u_1} + \{v_1, v_1q_1\}, T^1_{w_1} + \{v_1, v_1q_3\})$-ordering σ of $N_{B_1}(v_0) \cup N_{B_k}(v_k) = N_H(\{v_0, v_2\}) - \{u_1, w_1\}$ as in Subcase 3.1 and extends g to σ from u_1. Thus, the algorithm spends $O(|V(G)|^2)$ time in this case.
  - If v_1 has a neighbor in $V(G_{i-1})$, then the algorithm computes a $(T^1_{u_1}, T^1_{w_1})$-ordering σ of $N_{B_1^+}(v_0) - \{u_1, w_1\}$, and it performs an extension on g. The sequence σ can be computed in $O(|V(G)|)$ time by Lemma 3.4, and the extension can be computed in $O(|V(G)|)$ time. Thus, the algorithm spends $O(|V(G)|)$ time in this case.
- If Subcase 3.3 occurs ($k \geq 3$, or $k = 2$ and both B_1, B_2 are 2-connected), then the algorithm extends g to $N_{B_1}(v_0)$ and extends g to $N_{B_k}(v_k)$. The algorithm may need to compute a $(T^1_{u_1}, T^1_{w_1})$-ordering of $N_{B_1^+}(v_0) - \{u_1, w_1\}$ and a $(T^k_{u_k}, T^k_{w_k})$-ordering of $N_{B_k^+}(v_k) - \{u_k, w_k\}$, but both can be done in $O(|V(G)|)$ time by Lemma 3.4. It is not hard to check that the algorithm spends $O(|V(G)|)$ time in this case.

If Case 4 occurs (H_i is a triangle chain), then Algorithm Numbering g extends g to v_1, v_2, v_3. This can be done in $O(|V(G)|)$ time.

From the analysis above, it follows that Algorithm Numbering g spends $O(|V(G)|^2)$ time in each iteration. Since the number of iterations is $t < n \to t < |V(G)|$, the numbering g can be computed in $O(|V(G)|^3)$ time. □

Note that the extension operation does not affect the order of the vertices previously numbered, although their actual g values may have changed. Thus, at each iteration the algorithm orders the vertices in $D_i - D_{i-1}$ without affecting the order of the vertices in D_{i-1}. In fact, it does not affect the order of the vertices in D_j for every $1 \leq j \leq i-1$.

The numbering g will be used to construct *two* independent spanning trees rooted at r from $\mathcal{C} = (H_1, \ldots, H_t)$ in order from H_1 to H_t. For constructing the other two spanning trees we compute a numbering f by examining the chains of $\mathcal{C}$ in reverse order.

The algorithm for computing f is analogous to Algorithm Numbering g when it deals with an up G_{i-1}-chain or a down G_{i-1}-chain or elementary G_{i-1}-chain. The differences appear when it deals with a triangle G_{i-1}-chain. The algorithm also computes a sequence $\{r\} = D'_{t+1} \subset D'_t \subset \cdots \subset D'_2$ of subsets of $V(G)$ such that for $t \geq i \geq 1$, $N_G(H_i) \cap V(\bar{G}_i) \subseteq D'_{i+1}$.

Algorithm numbering f.

Description. The algorithm executes $t-1$ iterations, where t is the number of chains in $\mathcal{C}' := (H_1, \ldots, H_t)$. At the beginning of the first iteration, we have $i = t$, $D'_{t+1} := \{r\}$, and $f(r) := 1$. At the beginning of each iteration, we have an integer i with $t \geq i \geq 2$, a subset $D'_{i+1} \subseteq V(\bar{G}_i)$ such that $N_G(H_i) \cap V(\bar{G}_i) \subseteq D'_{i+1}$, and a numbering f of D'_{i+1}.

Each iteration consists of the following: update f and define D'_i according to the following cases (depending on the chain type of H_i), and, if $i > 2$, then set $i \leftarrow i-1$ and start a new iteration.

Case 1. H_i is an elementary G_{i-1}-chain in G.

Let $H_i := v_0B_1v_1B_2v_2$, and let v'_0, v'_2 be neighbors of v_1 in $V(\bar{G}_i)$ with $f(v'_0) < f(v'_2)$. Extend f to v_1 from v'_0, and let $D'_i := D'_{i+1} \cup \{v_1\}$.

Case 2. $i = t$, or H_i is a down G_{i-1}-chain in G but not an elementary G_{i-1}-chain.

Let $H_i := v_0B_1v_1 \ldots v_{k-1}B_kv_k$, and suppose that $v_0, \ldots, v_k$ and $B_1, \ldots, B_k$ are labeled so that $v_0 = v_k = r$ when $i = t$ and $f(v_0) < f(v_k)$ when $i \neq t$. For each 2-connected B_j, let u_j, w_j be the terminals of B_j^+ other than v_{j-1}, v_j. Let $T^j_{v_{j-1}}, T^j_{v_j}$ denote the trees rooted, respectively, at v_{j-1}, v_j in the independent spanning $\{v_{j-1}, v_j, u_j, w_j\}$-system of B_j^+ computed in Assumption 4.5.

For each $j = 1, \ldots, k$ compute a sequence σ_j as follows. If B_j is 2-connected, then let σ_j be a $(T^j_{v_{j-1}}, T^j_{v_j})$-ordering of $N_{B_j^+}(\{u_j, w_j\}) - \{v_{j-1}, v_j\}$ (the existence of this ordering is guaranteed by Lemma 3.5). If B_j is trivial, then let σ_j denote the empty sequence.

Extend f to $\sigma := \sigma_1, v_1, \sigma_2, v_2, \ldots, v_{k-1}, \sigma_k$ from v_0, and let $D'_i := D'_{i+1} \cup \{\sigma\}$. See Figure 13 for an illustration.

Case 3. H_i is an up G_{i-1}-chain in G but not an elementary G_{i-1}-chain in G.

Let $H_i := v_0B_1v_1 \ldots v_{k-1}B_kv_k$. For each 2-connected B_j, let u_j, w_j denote the terminals of B_j^+ other than v_{j-1}, v_j, with $f(u_j) < f(w_j)$. Let $T^j_{u_j}, T^j_{w_j}$ denote the trees rooted, respectively, at u_j, w_j in the independent spanning $\{v_{j-1}, v_j, u_j, w_j\}$-system of B_j^+ computed in Assumption 4.5.

Let $D'_i := D'_{i+1} \cup N_{B_1}(v_0) \cup N_{B_k}(v_k)$. See Figure 14 for an illustration. Extend f according to the following three subcases.

Subcase 3.1. $k = 1$ (thus, B_1 is 2-connected).

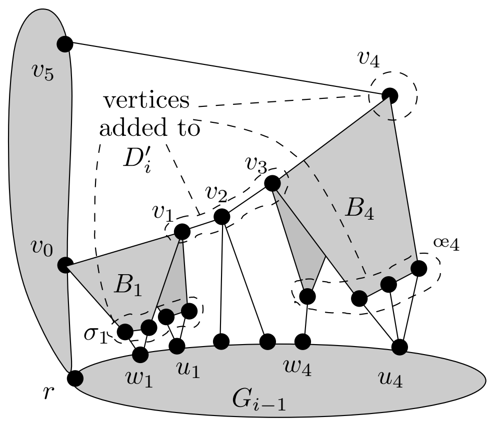

FIG. 13. *Extending the numbering f to a down G_{i-1}-chain.*

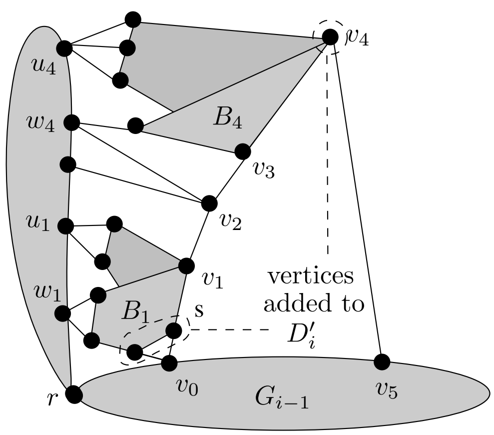

FIG. 14. *Extending the numbering f to an up G_{i-1}-chain.*

Let σ denote a $(T^1_{u_1}, T^1_{w_1})$-ordering of $N_{B_1^+}(\{v_0, v_1\}) - \{u_1, w_1\} = N_{B_1}(v_0) \cup N_{B_k}(v_k)$ (the existence of this ordering is guaranteed by Lemma 3.5). Extend f to σ from u_1.

Subcase 3.2. $k = 2$, and B_1 or B_2 is trivial.

Note that since H_i is not an elementary chain, B_1 or B_2 is nontrivial.

Assume then (renaming B_1 and B_2 if necessary) that B_1 is 2-connected and B_2 is trivial. Extend f according to the following cases.

- v_1 has no neighbor in $V(\bar{G}_i)$. Let q_1, q_2, q_3 be distinct neighbors of v_1 in B_1 (they exist since G is 4-connected), and assume that q_1, q_2, q_3 is $(T^1_{u_1}, T^1_{w_1})$-ordered (this is possible by Lemma 3.4). By Lemma 3.7, $T^1_{u_1}[u_1, q_1], T^1_{v_0}[v_0, q_2]$, and $T^1_{w_1}[w_1, q_3]$ are disjoint. Let $H := B_1^+ \cup B_2$. Note that $H - \{v_0, v_2, u_1, w_1\}$ is a component of $G - \{v_0, v_2, u_1, w_1\}$, (H, v_0, u_1, v_2, w_1) is planar, and $\{T^1_{v_0} + \{v_1, v_1q_2\}, T^1_{v_1} + \{v_2, v_1v_2\}, T^1_{u_1} + \{v_1, v_1q_1\}, T^1_{w_1} + \{v_1, v_1q_3\}\}$ is an independent spanning $\{v_0, v_2, u_1, w_1\}$-system of H.

  Let σ denote a $(T^1_{u_1} + \{v_1, v_1q_1\}, T^1_{w_1} + \{v_1, v_1q_3\})$-ordering of $N_H(\{v_0, v_2\}) - \{u_1, w_1\} = N_{B_1}(v_0) \cup N_{B_k}(v_k)$ (the existence of this ordering is guaranteed by Lemma 3.5). Extend f to σ from u_1.

Comment: we also keep track of q_1, q_2, q_3 for the construction of the independent spanning trees.

- v_1 has a neighbor in $V(\bar{G}_i)$. Let $x \in N_G(v_1) \cap V(\bar{G}_i)$ with $f(x)$ minimum, and let σ denote a $(T^1_{u_1}, T^1_{w_1})$-ordering of $N_{B_1}(v_0) = N_{B_1^+}(v_0) - \{u_1, w_1\}$ (the existence of this ordering is guaranteed by Lemma 3.4). If $f(x) > f(u_1)$, then extend f to σ, v_1 from u_1. If $f(x) \le f(u_1)$, then extend f to v_1, σ from x.

Subcase 3.3. $k \ge 3$, or $k = 2$ and both B_1, B_2 are 2-connected.

Extend f to $N_{B_1}(v_0)$ according to the following cases.

- B_1 is 2-connected. Let σ denote a $(T^1_{u_1}, T^1_{w_1})$-ordering of $N_{B_1}(v_0) = N_{B_1^+}(v_0) - \{u_1, w_1\}$ (the existence of this ordering is guaranteed by Lemma 3.4). Extend f to σ from u_1.
- Both B_1 and B_2 are trivial. Let $x \in N_G(v_1) \cap V(\bar{G}_i)$ with $f(x)$ minimum. Extend f to v_1 from x.
- B_1 is trivial and B_2 is 2-connected.
  - If v_1 has no neighbor in $V(\bar{G}_i)$, extend f to v_1 from u_2.
  - If v_1 has a neighbor in $V(\bar{G}_i)$, let $x \in N_G(v_1) \cap V(\bar{G}_i)$ with $f(x)$ minimum. If $f(x) > f(u_2)$, then extend f to v_1 from u_2. If $f(x) \le f(u_2)$, then extend f to v_1 from x.

Extend (the resulting) f to $N_{B_k}(v_k)$ according to the following cases.

- B_k is 2-connected. Let σ denote a $(T^k_{u_k}, T^k_{w_k})$-ordering of $N_{B_k}(v_k) = N_{B_k^+}(v_k) - \{u_k, w_k\}$ (the existence of this ordering is guaranteed by Lemma 3.4). Extend f to σ from u_k.
- Both B_k and B_{k-1} are trivial. Let $x \in N_G(v_{k-1}) \cap V(\bar{G}_i)$ with $f(x)$ minimum. Extend f to v_{k-1} from x.
- B_k is trivial and B_{k-1} is 2-connected.
  - If v_{k-1} has no neighbor in $V(\bar{G}_i)$, extend f to v_{k-1} from u_{k-1}.
  - If v_{k-1} has a neighbor in $V(\bar{G}_i)$, let $x \in N_G(v_{k-1}) \cap V(\bar{G}_i)$ with $f(x)$ minimum. If $f(x) > f(u_{k-1})$, then extend f to v_{k-1} from u_{k-1}. If $f(x) \le f(u_{k-1})$, then extend f to v_{k-1} from x.

Case 4. H_i is a triangle G_{i-1}-chain in G.

Let $I(H_i) := \{v_1, v_2, v_3\}$, let $v_j x_j$ $(j = 1, 2, 3)$ be the legs of H_i, and let $y_1, y_2, y_3 \in V(\bar{G}_i)$ such that $y_1v_1, y_2v_2, y_3v_3 \in E(G)$. Assume that v_1, v_2, v_3 are labeled so that $g(x_1) < g(x_2) < g(x_3)$. Let $D'_i := D'_{i+1} \cup \{v_1, v_2, v_3\}$.

- If $f(y_1) < f(y_2)$ and $f(y_1) < f(y_3)$, then extend f to v_1, v_2, v_3 from y_1.
- If $f(y_2) < f(y_1)$ and $f(y_2) < f(y_3)$, then extend f to v_2, v_1, v_3 from y_2.
- If $f(y_3) < f(y_1) < f(y_2)$, then extend f to v_3 from y_3 and extend (the resulting) f to v_1, v_2 from y_1.
- If $f(y_3) < f(y_2) < f(y_1)$, then extend f to v_3 from y_3 and extend (the resulting) f to v_2, v_1 from y_2.

This concludes the description of the algorithm for computing f. The proof of the next lemma is similar to the proof of Lemma 4.6, and we omit it.

LEMMA 4.7. *Algorithm Numbering f runs in $O(|V(G)|^3)$ time.*

5. Construction of spanning trees. We now describe how to use Theorem 3.2 and the two numberings of the last section to produce four independent spanning trees. This will follow from Algorithm Trees. The proof of its correction and analysis of its complexity will be given in the next section.

ALGORITHM TREES.

Description. Let G be a 4-connected graph, let $r \in V(G)$, and let $\mathcal{C} = (H_1, \ldots, H_t)$ be a nonseparating chain decomposition of G rooted at r. Let $G_0 = \bar{G}_t = (\{r\}, \emptyset)$, and for $1 \le i \le t-1$, let $G_i := G[\bigcup_{j=1}^{i} I(H_j)]$ and $\bar{G}_i = G - (V(G_i) - \{r\})$. The algorithm executes t iterations, where t is the number of chains in $\mathcal{C}$. At the first iteration, we have $i = 1$ and $T_1 = T_2 = T_3 = T_4 = G_0$. At the beginning of each iteration, we have an integer i with $1 \le i \le t$, spanning trees T_1, T_2 in G_{i-1} and spanning forests T_3, T_4 in $G_{i-1} - r$.

Each iteration consists of the following: update T_1, T_2, T_3, T_4 by adding certain vertices and edges of H_i to T_1, T_2, T_3, T_4 according to the following four cases (depending on the type of H_i), and, if $i < t$, then set $i \leftarrow i + 1$ and start a new iteration. After t iterations, T_1, T_2, T_3, T_4 will be independent spanning trees in G rooted at r.

Case 1. H_i is an elementary G_{i-1}-chain in G.

Let $H_i := v_0 B_1 v_1 B_2 v_2$ with $g(v_0) < g(v_2)$. Let v_0', v_2' be neighbors of v_1 in $V(\bar{G}_i)$ with $f(v_0') < f(v_2')$.

Set $T_1 \leftarrow T_1 + \{v_1, v_0 v_1\}, T_2 \leftarrow T_2 + \{v_1, v_1 v_2\}, T_3 \leftarrow T_3 + \{v_0', v_1, v_0' v_1\}$, and $T_4 \leftarrow T_4 + \{v_2', v_1, v_2' v_1\}$.

Case 2. $i = 1$, or H_i is an up G_{i-1}-chain in G but not an elementary G_{i-1}-chain in G.

Let $H_i := v_0 B_1 v_1 \ldots v_{k-1} B_k v_k$, with $v_0 = v_k = r$ when $i = 1$, and $g(v_0) < g(v_k)$ when $i \ne 1$.

For each 2-connected block B_j, let u_j, w_j denote the terminals of B_j^+ other than v_{j-1}, v_j with $f(u_j) < f(w_j)$, and let $T^j_{v_{j-1}}, T^j_{v_j}, T^j_{u_j}, T^j_{w_j}$ denote the trees rooted, respectively, at v_{j-1}, v_j, u_j, w_j in the independent spanning $\{v_{j-1}, v_j, u_j, w_j\}$-system of B_j^+ computed in Assumption 4.5.

Let $J := \{j : 1 \le j \le k, B_j \text{ is 2-connected}\}$, and let $\bar{J} := \{1, \ldots, k\} - J$.

First, set

$$T_1 \leftarrow T_1 \cup \left(\bigcup_{j \in \bar{J} - \{k\}} B_j \right) \cup \left(\bigcup_{j \in J} T^j_{v_{j-1}} \right),$$

$$T_2 \leftarrow T_2 \cup \left(\bigcup_{j \in \bar{J} - \{1\}} B_j \right) \cup \left(\bigcup_{j \in J} T^j_{v_j} \right),$$

$$T_3 \leftarrow T_3 \cup \left(\bigcup_{j \in J} T^j_{u_j} \right), \text{ and}$$

$$T_4 \leftarrow T_4 \cup \left(\bigcup_{j \in J} T^j_{w_j} \right).$$

Now for each $j = 1, \ldots, k-1$ add v_j and edges incident to v_j to T_1, T_2, T_3, T_4 according to the following cases (at this stage, $v_0, v_k \notin V(T_3 \cup T_4)$).

Subcase 2.1. B_j and B_{j+1} are trivial.

Let p_3, p_4 be neighbors of v_j in $V(\bar{G}_i)$ with $f(p_3)$ minimum (hence $f(p_3) < f(p_4)$). Set $T_3 \leftarrow T_3 + \{v_j, p_3, v_j p_3\}$ and $T_4 \leftarrow T_4 + \{v_j, p_4, v_j p_4\}$.

Subcase 2.2. B_j is 2-connected and B_{j+1} is trivial.

- If v_j has no neighbor in $V(\bar{G}_i)$, then let p_1, p_3, p_4 be neighbors of v_j in B_j (they exist since G is 4-connected), and assume that p_3, p_1, p_4 is $(T^j_{u_j}, T^j_{w_j})$-ordered (this is possible by Lemma 3.4). By Lemma 3.7, $T^j_{u_j}[u_j, p_3], T^j_{v_{j-1}}[v_{j-1}, p_1]$,

and $T^j_{w_j}[w_j,p_4]$ are disjoint. If $k=2$, then we also require that p_3,p_1,p_4 be the vertices q_1,q_2,q_3, respectively, chosen in Subcase 3.2 of Algorithm Numbering f.
Set $T_1 \leftarrow T_1+\{v_j,v_jp_1\}, T_3 \leftarrow T_3+\{v_j,v_jp_3\}$, and $T_4 \leftarrow T_4+\{v_j,v_jp_4\}$.

- If v_j has a neighbor in $V(\bar{G}_i)$, then let $x \in N_G(v_j)\cap V(\bar{G}_i)$ with $f(x)$ minimum.
  - If $f(x) > f(u_j)$, then let p_1,p_3 be neighbors of v_j in B_j such that the paths $T^j_{v_{j-1}}[v_{j-1},p_1]$ and $T^j_{u_j}[u_j,p_3]$ are disjoint (they exist by Lemma 3.6).
    Set $T_1 \leftarrow T_1+\{v_j,v_jp_1\}, T_3 \leftarrow T_3+\{v_j,v_jp_3\}$, and $T_4 \leftarrow T_4+\{v_j,x,v_jx\}$.
  - If $f(x) \le f(u_j)$, then let p_1,p_4 be neighbors of v_j in B_j such that the paths $T^j_{v_{j-1}}[v_{j-1},p_1]$ and $T^j_{w_j}[w_j,p_4]$ are disjoint (they exist by Lemma 3.6).
    Set $T_1 \leftarrow T_1+\{v_j,v_jp_1\}, T_3 \leftarrow T_3+\{v_j,x,v_jx\}$, and $T_4 \leftarrow T_4+\{v_j,v_jp_4\}$.

Subcase 2.3. B_j is trivial and B_{j+1} is 2-connected.

- If v_j has no neighbor in $V(\bar{G}_i)$, then let p_2,p_3,p_4 be neighbors of v_j in B_{j+1} (they exist since G is 4-connected), and assume that p_3,p_2,p_4 is $(T^{j+1}_{u_{j+1}},T^{j+1}_{w_{j+1}})$-ordered (this is possible by Lemma 3.4). By Lemma 3.7, $T^{j+1}_{u_{j+1}}[u_{j+1},p_3]$, $T^{j+1}_{v_{j+1}}[v_{j+1},p_2]$, and $T^{j+1}_{w_{j+1}}[w_{j+1},p_4]$ are disjoint. If $k=2$, then we also require that p_3,p_2,p_4 be the vertices q_1,q_2,q_3, respectively, chosen in Subcase 3.2 of Algorithm Numbering f.
  Set $T_2 \leftarrow T_2+\{v_j,v_jp_2\}, T_3 \leftarrow T_3+\{v_j,v_jp_3\}$, and $T_4 \leftarrow T_4+\{v_j,v_jp_4\}$.
- If v_j has a neighbor in $V(\bar{G}_i)$, then let $x \in N_G(v_j)\cap V(\bar{G}_i)$ with $f(x)$ minimum.
  - If $f(x) > f(u_{j+1})$, then let p_2,p_3 be neighbors of v_j in B_{j+1} such that the paths $T^{j+1}_{v_{j+1}}[v_{j+1},p_2]$ and $T^{j+1}_{u_{j+1}}[u_{j+1},p_3]$ are disjoint (they exist by Lemma 3.6).
    Set $T_2 \leftarrow T_2+\{v_j,v_jp_2\}, T_3 \leftarrow T_3+\{v_j,v_jp_3\}$, and $T_4 \leftarrow T_4+\{v_j,x,v_jx\}$.
  - If $f(x) \le f(u_{j+1})$, then let p_2,p_4 be neighbors of v_j in B_{j+1} such that the paths $T^{j+1}_{v_{j+1}}[v_{j+1},p_2]$ and $T^{j+1}_{w_{j+1}}[w_{j+1},p_4]$ are disjoint (they exist by Lemma 3.6).
    Set $T_2 \leftarrow T_2+\{v_j,v_jp_2\}, T_3 \leftarrow T_3+\{v_j,x,v_jx\}$, and $T_4 \leftarrow T_4+\{v_j,v_jp_4\}$.

Subcase 2.4. B_j and B_{j+1} are 2-connected.
Note that $f(u_j) < f(w_{j+1})$ or $f(u_{j+1}) < f(w_j)$.

- If $f(u_j) < f(w_{j+1})$, then let p_1,p_3 be neighbors of v_j in B_j such that the paths $T^j_{v_{j-1}}[v_{j-1},p_1]$ and $T^j_{u_j}[u_j,p_3]$ are disjoint (they exist by Lemma 3.6), and let p_2,p_4 be neighbors of v_j in B_{j+1} such that the paths $T^{j+1}_{v_{j+1}}[v_{j+1},p_2]$ and $T^{j+1}_{w_{j+1}}[w_{j+1},p_4]$ are disjoint (they exist by Lemma 3.6).
  Set $T_1 \leftarrow T_1+\{v_j,v_jp_1\}, T_2 \leftarrow T_2+\{v_j,v_jp_2\}, T_3 \leftarrow T_3+\{v_j,v_jp_3\}$, and $T_4 \leftarrow T_4+\{v_j,v_jp_4\}$.
- If $f(u_j) \ge f(w_{j+1})$, then $f(u_{j+1}) < f(w_j)$. Let p_1,p_4 be neighbors of v_j in B_j such that the paths $T^j_{v_{j-1}}[v_{j-1},p_1]$ and $T^j_{w_j}[w_j,p_4]$ are disjoint (they exist by Lemma 3.6), and let p_2,p_3 be neighbors of v_j in B_{j+1} such that the paths $T^{j+1}_{v_{j+1}}[v_{j+1},p_2]$ and $T^{j+1}_{u_{j+1}}[u_{j+1},p_3]$ are disjoint (they exist by Lemma 3.6).
  Set $T_1 \leftarrow T_1+\{v_j,v_jp_1\}, T_2 \leftarrow T_2+\{v_j,v_jp_2\}, T_3 \leftarrow T_3+\{v_j,v_jp_3\}$, and $T_4 \leftarrow T_4+\{v_j,v_jp_4\}$.

Case 3. $i=t$, or H_i is a down G_{i-1}-chain in G.

Let $H_i := v_0B_1v_1\dots v_{k-1}B_kv_k$, with $v_0=v_k=r$ when $i=t$, and $f(v_0)<f(v_k)$ when $i \ne t$.

For each 2-connected block B_j, let u_j, w_j denote the terminals of B_j^+ other than v_{j-1}, v_j, with $g(u_j) < g(w_j)$, and let $T^j_{v_{j-1}}, T^j_{v_j}, T^j_{u_j}, T^j_{w_j}$ denote the trees rooted, respectively, at v_{j-1}, v_j, u_j, w_j in the independent spanning $\{v_{j-1}, v_j, u_j, w_j\}$-system of B_j^+ computed in Assumption 4.5.

Let $J := \{j : 1 \le j \le k, B_j \text{ is 2-connected}\}$, and let $\bar{J} := \{1, \dots, k\} - J$.

First, set

$$T_1 \leftarrow T_1 \cup \left(\bigcup_{j \in J} T^j_{u_j} \right),$$

$$T_2 \leftarrow T_2 \cup \left(\bigcup_{j \in J} T^j_{w_j} \right),$$

$$T_3 \leftarrow T_3 \cup \left(\bigcup_{j \in \bar{J} - \{k\}} B_j \right) \cup \left(\bigcup_{j \in J} T^j_{v_{j-1}} \right), \text{ and}$$

$$T_4 \leftarrow T_4 \cup \left(\bigcup_{j \in \bar{J} - \{1\}} B_j \right) \cup \left(\bigcup_{j \in J} T^j_{v_j} \right).$$

Now for each $j = 1, \dots, k-1$ add v_j and edges incident to v_j to T_1, T_2, T_3, T_4 according to the following cases (at this stage $v_0, v_k \notin V(T_1 \cup T_2)$).

Subcase 3.1. B_j and B_{j+1} are trivial blocks.

Let p_1, p_2 be neighbors of v_j in $V(G_{i-1})$ with $g(p_1)$ minimum (hence $g(p_1) < g(p_2)$).

Set $T_1 \leftarrow T_1 + \{v_j, v_j p_1\}$ and $T_2 \leftarrow T_2 + \{v_j, v_j p_2\}$.

Subcase 3.2. B_j is 2-connected and B_{j+1} is trivial.

- If v_j has no neighbor in $V(G_{i-1})$, then let p_1, p_2, p_3 be neighbors of v_j in B_j (they exist since G is 4-connected), and assume that p_1, p_3, p_2 is $(T^j_{u_j}, T^j_{w_j})$-ordered (this is possible by Lemma 3.4). By Lemma 3.7, $T^j_{u_j}[u_j, p_1]$, $T^j_{v_{j-1}}[v_{j-1}, p_3]$, and $T^j_{w_j}[w_j, p_2]$ are disjoint. If $k = 2$, then we also require that p_1, p_3, p_2 be the vertices q_1, q_2, q_3, respectively, chosen in Subcase 3.2 of Algorithm Numbering g.

  Set $T_1 \leftarrow T_1 + \{v_j, v_j p_1\}, T_2 \leftarrow T_2 + \{v_j, v_j p_2\}$, and $T_3 \leftarrow T_3 + \{v_j, v_j p_3\}$.
- If v_j has a neighbor in $V(G_{i-1})$, then let $x \in N_G(v_j) \cap V(G_{i-1})$ with $g(x)$ minimum.
  - If $g(x) > g(u_j)$, then let p_1, p_3 be neighbors of v_j in B_j such that the paths $T^j_{u_j}[u_j, p_1]$ and $T^j_{v_{j-1}}[v_{j-1}, p_3]$ are disjoint (they exist by Lemma 3.6).

    Set $T_1 \leftarrow T_1 + \{v_j, v_j p_1\}, T_2 \leftarrow T_2 + \{v_j, v_j x\}$, and $T_3 \leftarrow T_3 + \{v_j, v_j p_3\}$.
  - If $g(x) \le g(u_j)$, then let p_2, p_3 be neighbors of v_j in B_j such that the paths $T^j_{w_j}[w_j, p_2]$ and $T^j_{v_{j-1}}[v_{j-1}, p_3]$ are disjoint (they exist by Lemma 3.6).

    Set $T_1 \leftarrow T_1 + \{v_j, v_j x\}, T_2 \leftarrow T_2 + \{v_j, v_j p_2\}$, and $T_3 \leftarrow T_3 + \{v_j, v_j p_3\}$.

Subcase 3.3. B_j is trivial and B_{j+1} is 2-connected.

- If v_j has no neighbor in $V(G_{i-1})$, then let p_1, p_2, p_4 be neighbors of v_j in B_{j+1} (they exist since G is 4-connected), and assume that p_1, p_4, p_2 is $(T^{j+1}_{u_{j+1}}, T^{j+1}_{w_{j+1}})$-ordered (this is possible by Lemma 3.4). By Lemma 3.7, $T^{j+1}_{u_{j+1}}[u_{j+1}, p_1], T^{j+1}_{v_{j+1}}[v_{j+1}, p_4]$, and $T^{j+1}_{w_{j+1}}[w_{j+1}, p_2]$ are disjoint. If $k = 2$, then

we also require that p_1, p_4, p_2 be the vertices q_1, q_2, q_3, respectively, chosen in Subcase 3.2 of Algorithm Numbering g.
Set $T_1 \leftarrow T_1 + \{v_j, v_jp_1\}, T_2 \leftarrow T_2 + \{v_j, v_jp_2\}$, and $T_4 \leftarrow T_4 + \{v_j, v_jp_4\}$.

- If v_j has a neighbor in $V(G_{i-1})$, then let $x \in N_G(v_j) \cap V(G_{i-1})$ with $g(x)$ minimum.
  - If $g(x) > g(u_{j+1})$, then let p_1, p_4 be neighbors of v_j in B_{j+1} such that the paths $T^{j+1}_{u_{j+1}}[u_{j+1}, p_1]$ and $T^{j+1}_{v_{j+1}}[v_{j+1}, p_4]$ are disjoint (they exist by Lemma 3.6).
    Set $T_1 \leftarrow T_1 + \{v_j, v_jp_1\}, T_2 \leftarrow T_2 + \{v_j, v_jx\}$, and $T_4 \leftarrow T_4 + \{v_j, v_jp_4\}$.
  - If $g(x) \le g(u_{j+1})$, then let p_2, p_4 be neighbors of v_j in B_{j+1} such that the paths $T^{j+1}_{w_{j+1}}[w_{j+1}, p_2]$ and $T^{j+1}_{v_{j+1}}[v_{j+1}, p_4]$ are disjoint (they exist by Lemma 3.6).
    Set $T_1 \leftarrow T_1 + \{v_j, v_jx\}, T_2 \leftarrow T_2 + \{v_j, v_jp_2\}$, and $T_4 \leftarrow T_4 + \{v_j, v_jp_4\}$.

Subcase 3.4. B_j and B_{j+1} are 2-connected.

Note that $g(u_j) < g(w_{j+1})$ or $g(u_{j+1}) < g(w_j)$.

- If $g(u_j) < g(w_{j+1})$, then let p_1, p_3 be neighbors of v_j in B_j such that the paths $T^j_{u_j}[u_j, p_1]$ and $T^j_{v_{j-1}}[v_{j-1}, p_3]$ are disjoint (they exist by Lemma 3.6), and let p_2, p_4 be neighbors of v_j in B_{j+1} such that the paths $T^{j+1}_{w_{j+1}}[w_{j+1}, p_2]$ and $T^{j+1}_{v_{j+1}}[v_{j+1}, p_4]$ are disjoint (they exist by Lemma 3.6).
  Set $T_1 \leftarrow T_1 + \{v_j, v_jp_1\}, T_2 \leftarrow T_2 + \{v_j, v_jp_2\}, T_3 \leftarrow T_3 + \{v_j, v_jp_3\}$, and $T_4 \leftarrow T_4 + \{v_j, v_jp_4\}$.
- If $g(u_j) \ge g(w_{j+1})$, then $g(u_{j+1}) < g(w_j)$. Let p_2, p_3 be neighbors of v_j in B_j such that the paths $T^j_{w_j}[w_j, p_2]$ and $T^j_{v_{j-1}}[v_{j-1}, p_3]$ are disjoint (they exist by Lemma 3.6), and let p_1, p_4 be neighbors of v_j in B_{j+1} such that the paths $T^{j+1}_{u_{j+1}}[u_{j+1}, p_1]$ and $T^{j+1}_{v_{j+1}}[v_{j+1}, p_4]$ are disjoint (they exist by Lemma 3.6).
  Set $T_1 \leftarrow T_1 + \{v_j, v_jp_1\}, T_2 \leftarrow T_2 + \{v_j, v_jp_2\}, T_3 \leftarrow T_3 + \{v_j, v_jp_3\}$, and $T_4 \leftarrow T_4 + \{v_j, v_jp_4\}$.

Case 4. H_i is a triangle G_{i-1}-chain in G.

Let $I(H_i) := \{v_1, v_2, v_3\}$, let v_jx_j $(j = 1, 2, 3)$ be the legs of H_i, and let $y_1, y_2, y_3 \in V(\bar{G}_i)$ such that $y_1v_1, y_2v_2, y_3v_3 \in E(G)$. Assume that v_1, v_2, v_3 are labeled so that $g(x_1) < g(x_2) < g(x_3)$.

Update T_1, T_2, T_3, T_4 according to the following four possibilities.

- If $f(y_1) < f(y_2)$ and $f(y_1) < f(y_3)$ then set
  $T_1 \leftarrow T_1 + \{v_1, v_2, v_3, x_1v_1, x_2v_2, v_2v_3\}, T_2 \leftarrow T_2 + \{v_1, v_2, v_3, x_3v_3, v_3v_1, v_3v_2\}$,
  $T_3 \leftarrow T_3 + \{v_1, v_2, v_3, y_1v_1, v_1v_2, v_1v_3\}$, $T_4 \leftarrow T_4 + \{v_1, v_2, v_3, y_2v_2, v_2v_1, y_3v_3\}$.
- If $f(y_2) < f(y_1)$ and $f(y_2) < f(y_3)$ then set
  $T_1 \leftarrow T_1 + \{v_1, v_2, v_3, x_1v_1, x_2v_2, v_1v_3\}, T_2 \leftarrow T_2 + \{v_1, v_2, v_3, x_3v_3, v_3v_1, v_3v_2\}$,
  $T_3 \leftarrow T_3 + \{v_1, v_2, v_3, y_2v_2, v_2v_1, v_2v_3\}$, $T_4 \leftarrow T_4 + \{v_1, v_2, v_3, y_1v_1, v_1v_2, y_3v_3\}$.
- If $f(y_3) < f(y_1) < f(y_2)$ then set
  $T_1 \leftarrow T_1 + \{v_1, v_2, v_3, x_1v_1, x_2v_2, v_1v_3\}, T_2 \leftarrow T_2 + \{v_1, v_2, v_3, x_3v_3, v_3v_1, v_3v_2\}$,
  $T_3 \leftarrow T_3 + \{v_1, v_2, v_3, y_1v_1, v_1v_2, y_3v_3\}$, $T_4 \leftarrow T_4 + \{v_1, v_2, v_3, y_2v_2, v_2v_1, v_2v_3\}$.
- If $f(y_3) < f(y_2) < f(y_1)$ then set
  $T_1 \leftarrow T_1 + \{v_1, v_2, v_3, x_1v_1, x_2v_2, v_2v_3\}, T_2 \leftarrow T_2 + \{v_1, v_2, v_3, x_3v_3, v_3v_1, v_3v_2\}$,
  $T_3 \leftarrow T_3 + \{v_1, v_2, v_3, y_2v_2, v_2v_1, y_3v_3\}$, $T_4 \leftarrow T_4 + \{v_1, v_2, v_3, y_1v_1, v_1v_2, v_1v_3\}$.

6. Correctness of Algorithm Trees. In this section we will prove Theorem 1.1. More precisely, we will show that the subgraphs T_1, T_2, T_3, T_4 returned by Algorithm Trees are independent spanning trees of G rooted at r, and they can be computed in $O(|V(G)|^3)$ time.

Notation 6.1. Let G be a 4-connected graph, let $r \in V(G)$, and let $\mathcal{C} = (H_1, \ldots, H_t)$ be a nonseparating chain decomposition of G rooted at r. Let $G_0 = \bar{G}_t = (\{r\}, \emptyset)$, and for $1 \le i \le t-1$, let $G_i := G[\bigcup_{j=1}^{i} I(H_j)]$ and $\bar{G}_i = G - (V(G_i) - \{r\})$. Let T_1, T_2, T_3, T_4 denote the subgraphs returned by Algorithm Trees. Let D, D' denote the sets of vertices returned by Algorithm Numbering g and Algorithm Numbering f, respectively.

We start with a series of seven simple lemmas which follow from the cases of Algorithm Trees. The first lemma follows immediately by inspecting Case 1 of Algorithm Trees.

LEMMA 6.2. *Let $H_i := v_0B_1v_1B_2v_2$ be an elementary G_{i-1}-chain in G, with $g(v_0) < g(v_2)$. Then v_1 has neighbors v_0', v_2' in $V(\bar{G}_i)$, with $f(v_0') < f(v_2')$, such that*

(1) *$E(T_1 \cap H_i) = \{v_0v_1\}$ and $E(T_2 \cap H_i) = \{v_1v_2\}$, and*

(2) *$E(T_3 \cap H_i') = \{v_0'v_1\}$ and $E(T_4 \cap H_i') = \{v_1v_2'\}$, where $H_i' = v_0'B_1'v_1B_2'v_2'$ is an elementary $\bar{G}_i$-chain in G.*

The next lemma follows by inspecting Case 2 (for $i = 1$) of Algorithm Trees.

LEMMA 6.3. *Let $H_1 := v_0B_1v_1 \ldots v_{k-1}B_kv_k$, with $v_0 = v_k = r$, and for each 2-connected B_j, let u_j, w_j denote the terminals of B_j^+ other than v_{j-1}, v_j, with $f(u_j) < f(w_j)$. Let H_1^+ be the graph obtained from H_1 by adding $N_G(H_1 - r) - \{r\}$ and the edges of G from $V(H_1)$ to $N_G(H_1 - r) - \{r\}$. Then*

(1) *$T_1 \cap H_1$ is a spanning tree of H_1 rooted at r and contains no edge from r to $N_{B_k}(r)$,*

(2) *$T_2 \cap H_1$ is a spanning tree of H_1 rooted at r and contains no edge from r to $N_{B_1}(r)$,*

(3) *$T_3 \cap (H_1^+ - r)$ is a spanning forest of $H_1^+ - r$, and each component of $T_3 \cap (H_1^+ - r)$ either is a tree in $B_j^+ - w_j$ rooted at u_j for some $j \in \{1, \ldots, k\}$ or is induced by a single edge with one end in $V(\bar{G}_1)$ and the other in $\{v_1, \ldots, v_{k-1}\}$, and*

(4) *$T_4 \cap (H_1^+ - r)$ is a spanning forest of $H_1^+ - r$, and each component of $T_4 \cap (H_1^+ - r)$ either is a tree in $B_j^+ - u_j$ rooted at w_j for some $j \in \{1, \ldots, k\}$ or is induced by a single edge with one end in $V(\bar{G}_1)$ and the other in $\{v_1, \ldots, v_{k-1}\}$.*

By inspecting Case 2 (for $i \neq 1$) of Algorithms Trees, we have the following lemma.

LEMMA 6.4. *Let $H_i := v_0B_1v_1 \ldots v_{k-1}B_kv_k$ be an up G_{i-1}-chain in G $(2 \le i \le t-1)$, with $g(v_0) < g(v_k)$, and for each 2-connected block B_j, let u_j, w_j denote the terminals of B_j^+ other than v_{j-1}, v_j, with $f(u_j) < f(w_j)$. Let H_i^+ be the graph obtained from H_i by adding $N_G(H_i - \{v_0, v_k\}) - \{v_0, v_k\}$ and the edges of G from $V(H_i)$ to $N_G(H_i - \{v_0, v_k\}) - \{v_0, v_k\}$. Then*

(1) *$T_1 \cap (H_i - v_k)$ is a spanning tree of $H_i - v_k$ rooted at v_0, and T_1 contains no edge from v_k to $N_{B_k}(v_k)$,*

(2) *$T_2 \cap (H_i - v_0)$ is a spanning tree of $H_i - v_0$ rooted at v_k, and T_2 contains no edge from v_0 to $N_{B_1}(v_0)$,*

(3) *$T_3 \cap (H_i^+ - \{v_0, v_k\})$ is a spanning forest of $H_i^+ - \{v_0, v_k\}$, and each component of $T_3 \cap (H_i^+ - \{v_0, v_k\})$ either is a tree in $B_j^+ - w_j$ rooted at u_j for some $j \in \{1, \ldots, k\}$ or is induced by a single edge with one end in $V(\bar{G}_i)$ and the other in $\{v_1, \ldots, v_{k-1}\}$, and*

(4) *$T_4 \cap (H_i^+ - \{v_0, v_k\})$ is a spanning forest of $H_i^+ - \{v_0, v_k\}$, and each component of $T_4 \cap (H_i^+ - \{v_0, v_k\})$ either is a tree in $B_j^+ - u_j$ rooted at w_j for some $j \in \{1, \ldots, k\}$ or is induced by a single edge with one end in $V(\bar{G}_i)$ and the other in $\{v_1, \ldots, v_{k-1}\}$.*

By a simple inspection of Case 3 (for $i = t$) of Algorithm Trees, we have the following lemma.

LEMMA 6.5. *Let* $H_t := v_0B_1v_1 \dots v_{k-1}B_kv_k$, *with* $v_0 = v_k = r$, *and for each* 2-*connected* B_j, *let* u_j, w_j *denote the terminals of* B_j^+ *other than* v_{j-1}, v_j, *with* $g(u_j) < g(w_j)$. *Let* H_t^+ *be the graph obtained from* H_t *by adding* $N_G(H_t - r) - \{r\}$ *and the edges of* G *from* $V(H_t)$ *to* $N_G(H_t - r) - \{r\}$. *Then*

(1) $T_1 \cap (H_t^+ - r)$ *is a spanning forest of* $H_t^+ - r$, *and each component of* $T_1 \cap (H_t^+ - r)$ *either is a tree in* $B_j^+ - w_j$ *rooted at* u_j *for some* $j \in \{1, \dots, k\}$ *or is induced by a single edge with one end in* $V(G_{t-1})$ *and the other in* $\{v_1, \dots, v_{k-1}\}$,
(2) $T_2 \cap (H_t^+ - r)$ *is a spanning forest of* $H_t^+ - r$, *and each component of* $T_2 \cap (H_t^+ - r)$ *either is a tree in* $B_j^+ - u_j$ *rooted at* w_j *for some* $j \in \{1, \dots, k\}$ *or is induced by a single edge with one end in* $V(G_{t-1})$ *and the other in* $\{v_1, \dots, v_{k-1}\}$,
(3) $T_3 \cap H_t$ *is a spanning tree of* H_t *rooted at* r *and contains no edge from* r *to* $N_{B_k}(r)$, *and*
(4) $T_4 \cap H_t$ *is a spanning tree of* H_t *rooted at* r *and contains no edge from* r *to* $N_{B_1}(r)$.

The next lemma follows from a simple inspection of Case 3 (for $i \neq t$) of Algorithm Trees.

LEMMA 6.6. *Let* $H_i := v_0B_1v_1 \dots v_{k-1}B_kv_k$ *be a down* G_{i-1}-*chain in* $G\,(2 \leq i \leq t-1)$, *with* $f(v_0) < f(v_k)$, *and for each* 2-*connected block* B_j, *let* u_j, w_j *denote the terminals of* B_j^+ *other than* v_{j-1}, v_j, *with* $g(u_j) < g(w_j)$. *Let* H_i^+ *be the graph obtained from* H_i *by adding* $N_G(H_i - \{v_0, v_k\}) - \{v_0, v_k\}$ *and the edges of* G *from* $V(H_i)$ *to* $N_G(H_i - \{v_0, v_k\}) - \{v_0, v_k\}$. *Then*

(1) $T_1 \cap (H_i^+ - \{v_0, v_k\})$ *is a spanning forest of* $H_i^+ - \{v_0, v_k\}$, *and each component of* $T_1 \cap (H_i^+ - \{v_0, v_k\})$ *either is a tree in* $B_j^+ - w_j$ *rooted at* u_j *for some* $j \in \{1, \dots, k\}$ *or is induced by a single edge with one end in* $V(G_{i-1})$ *and the other in* $\{v_1, \dots, v_{k-1}\}$,
(2) $T_2 \cap (H_i^+ - \{v_0, v_k\})$ *is a spanning forest of* $H_i^+ - \{v_0, v_k\}$, *and each component of* $T_2 \cap (H_i^+ - \{v_0, v_k\})$ *either is a tree in* $B_j^+ - u_j$ *rooted at* w_j *for some* $j \in \{1, \dots, k\}$ *or is induced by a single edge with one end in* $V(G_{i-1})$ *and the other in* $\{v_1, \dots, v_{k-1}\}$,
(3) $T_3 \cap (H_i - v_k)$ *is a spanning tree of* $H_i - v_k$ *rooted at* v_0, *and* T_3 *contains no edge from* v_k *to* $N_{B_k}(v_k)$, *and*
(4) $T_4 \cap (H_i - v_0)$ *is a spanning tree of* $H_i - v_0$ *rooted at* v_k, *and* T_4 *contains no edge from* v_0 *to* $N_{B_1}(v_0)$.

Finally, by a simple inspection of Case 4 of Algorithm Trees, we have the following lemma.

LEMMA 6.7. *Let* H_i *be a triangle* G_{i-1}-*chain in* G $(2 \leq i \leq t-1)$. *Let* $I(H_i) := \{v_1, v_2, v_3\}$, *let* $y_1, y_2, y_3 \in V(\bar{G}_i)$ *such that* $y_1v_1, y_2v_2, y_3v_3 \in E(G)$, *and let* v_jx_j $(j = 1, 2, 3)$ *be the legs of* H_i, *with* $g(x_1) < g(x_2) < g(x_3)$. *Let* $H_i^+ := H_i + \{y_1, y_2, y_3, y_1v_1, y_2v_2, y_3v_3\}$.

- *If* $f(y_1) < f(y_2)$ *and* $f(y_1) < f(y_3)$, *then*
$E(T_1 \cap H_i^+) = \{x_1v_1, x_2v_2, v_2v_3\}$, $E(T_2 \cap H_i^+) = \{x_3v_3, v_3v_1, v_3v_2\}$,
$E(T_3 \cap H_i^+) = \{y_1v_1, v_1v_2, v_1v_3\}$, *and* $E(T_4 \cap H_i^+) = \{y_2v_2, v_2v_1, y_3v_3\}$.
- *If* $f(y_2) < f(y_1)$ *and* $f(y_2) < f(y_3)$, *then*
$E(T_1 \cap H_i^+) = \{x_1v_1, x_2v_2, v_1v_3\}$, $E(T_2 \cap H_i^+) = \{x_3v_3, v_3v_1, v_3v_2\}$,
$E(T_3 \cap H_i^+) = \{y_2v_2, v_2v_1, v_2v_3\}$, *and* $E(T_4 \cap H_i^+) = \{y_1v_1, v_1v_2, y_3v_3\}$.

- *If* $f(y_3) < f(y_1) < f(y_2)$, *then* $E(T_1 \cap H_i^+) = \{x_1v_1, x_2v_2, v_1v_3\}$, $E(T_2 \cap H_i^+) = \{x_3v_3, v_3v_1, v_3v_2\}$, $E(T_3 \cap H_i^+) = \{y_1v_1, v_1v_2, y_3v_3\}$, *and* $E(T_4 \cap H_i^+) = \{y_2v_2, v_2v_1, v_2v_3\}$.
- *If* $f(y_3) < f(y_2) < f(y_1)$, *then* $E(T_1 \cap H_i^+) = \{x_1v_1, x_2v_2, v_2v_3\}$, $E(T_2 \cap H_i^+) = \{x_3v_3, v_3v_1, v_3v_2\}$, $E(T_3 \cap H_i^+) = \{y_2v_2, v_2v_1, y_3v_3\}$, *and* $E(T_4 \cap H_i^+) = \{y_1v_1, v_1v_2, v_1v_3\}$.

We can now show that T_1, T_2, T_3, and T_4 are spanning trees of G.

LEMMA 6.8. *For every* $i = 1, \ldots, t$, $T_1 \cap G_i$ *and* $T_2 \cap G_i$ *are spanning trees of* G_i.

Proof. Note that every $v \in V(G) - \{r\}$ is an internal vertex of some chain H_i in $\mathcal{C}$. The result follows by induction on i with the help of (1) of Lemma 6.2, (1) and (2) of Lemma 6.3, (1) and (2) of Lemma 6.4, (1) and (2) of Lemma 6.5, (1) and (2) of Lemma 6.6, and Lemma 6.7. □

LEMMA 6.9. *For every* $i = t, \ldots, 1$, $T_3 \cap \bar{G}_i$ *and* $T_4 \cap \bar{G}_i$ *are spanning trees of* $\bar{G}_i$.

Proof. The result follows by induction on $t - i$ with the help of (2) of Lemma 6.2, (3) and (4) of Lemma 6.3, (3) and (4) of Lemma 6.4, (3) and (4) of Lemma 6.5, (3) and (4) of Lemma 6.6, and Lemma 6.7. □

Lemmas 6.8 and 6.9 imply the following.

COROLLARY 6.10. T_1, T_2, T_3, T_4 *are spanning trees of* G.

Now we proceed to show that T_1, T_2, T_3, T_4 are independent spanning trees of G rooted at r. The proof consists of several lemmas.

LEMMA 6.11. *For any* $1 \le i \le t$ *and for any* $v \in I(H_i) - \{r\}$, *there exist vertices* z_1, z_2, z_3, z_4 *such that*

(1) $z_1, z_2 \in V(G_{i-1})$, *and either* $z_1 = z_2 = r$ *or* $g(z_1) < g(z_2)$ *(and* $g(z_1) < g(v) < g(z_2)$ *if* $v \in D$*)*,

(2) $z_3, z_4 \in V(\bar{G}_i)$, *and either* $z_3 = z_4 = r$ *or* $f(z_3) < f(z_4)$ *(and* $f(z_3) < f(v) < f(z_4)$ *if* $v \in D'$*), and*

(3) $T_i[z_i, v]$, $i = 1, 2, 3, 4$, *are internally disjoint paths in* G, *and* $V(T_i[z_i, v]) - z_i \subseteq I(H_i)$.

Proof. Let $1 \le i \le t$ and $v \in I(H_i) - \{r\}$. We consider the four cases of Algorithm Trees.

Case 1. H_i is an elementary G_{i-1}-chain in G.

In this case, $2 \le i \le t-1$. Let $H_i := v_0B_1v_1B_2v_2$, with $g(v_0) < g(v_2)$. This is the same as in Case 1 of Algorithm Trees. Then $v_0, v_2 \in V(G_{i-1})$, $v = v_1$, and by Case 1 of Algorithm Numbering g, we have $g(v_0) < g(v_1) < g(v_2)$. By Lemma 6.2, there exist $v_0', v_2' \in V(\bar{G}_i)$, with $f(v_0') < f(v_2')$, such that $v_0v \in E(T_1), v_2v \in E(T_2), v_0'v \in E(T_3)$, and $v_2'v \in E(T_4)$. By Case 1 of Algorithm Numbering f, $f(v_0') < f(v_1) < f(v_2')$. Thus, the result follows by taking $z_1 := v_0$, $z_2 := v_2$, $z_3 := v_0'$, and $z_4 := v_2'$.

Case 2. $i = 1$, or H_i is an up G_{i-1}-chain in G but not an elementary G_{i-1}-chain.

Let $H_i := v_0B_1v_1 \ldots v_{k-1}B_kv_k$, with $v_0 = v_k = r$ when $i = 1$, and $g(v_0) < g(v_k)$ when $i \ne 1$. For each 2-connected B_j, let u_j, w_j denote the terminals of B_j^+ other than v_{j-1}, v_j, with $f(u_j) < f(w_j)$, and let $T^j_{v_{j-1}}, T^j_{v_j}, T^j_{u_j}, T^j_{w_j}$ denote the trees rooted, respectively, at v_{j-1}, v_j, u_j, w_j in the independent spanning $\{v_{j-1}, v_j, u_j, w_j\}$-system of B_j^+ computed in Assumption 4.5. This is the same as in Case 2 of Algorithm Trees.

Let $j \in \{1, \ldots, k-1\}$. If $i = 1$, then by (1) and (2) of Lemma 6.3, $T_1[r, v_j] \subseteq \bigcup_{l=1}^{j} B_l$ and $T_2[r, v_j] \subseteq \bigcup_{l=j+1}^{k} B_l$. If $i \ne 1$, then v_j is a cut vertex of H_i, and hence, by (1) and (2) of Lemma 6.4, $T_1[v_0, v_j] \subseteq \bigcup_{l=1}^{j} B_l$ and $T_2[v_k, v_j] \subseteq \bigcup_{l=j+1}^{k} B_l$.

First, let us consider the case when $v \ne v_j$ for $j = 1, \ldots, k-1$. Thus, there exists some j, $1 \le j \le k$, such that B_j is 2-connected and $v \in V(B_j) - \{v_{j-1}, v_j\}$. By Case 2 of Algorithm Numbering g, we know that $g(v_0) \le g(v_{j-1}) < g(v_j) \le$

$g(v_k)$, and if $v \in D$, then $g(v_0) \le g(v_{j-1}) < g(v) < g(v_j) \le g(v_k)$. Furthermore, $T^j_{v_{j-1}}[v_{j-1}, v], T^j_{v_j}[v_j, v], T^j_{u_j}[u_j, v]$, and $T^j_{w_j}[w_j, v]$ are internally disjoint, because $\{T^j_{v_{j-1}}, T^j_{v_j}, T^j_{u_j}, T^j_{w_j}\}$ is an independent spanning $\{v_{j-1}, v_j, u_j, w_j\}$-system of B^+_j. By the construction in Case 2 of Algorithm Trees, $T_1[v_{j-1}, v] = T^j_{v_{j-1}}[v_{j-1}, v]$, $T_2[v_j, v] = T^j_{v_j}[v_j, v]$, $T_3[u_j, v] = T^j_{u_j}[u_j, v]$, and $T_4[w_j, v] = T^j_{w_j}[w_j, v]$. By Case 3 of Algorithm Numbering f, if $v \in D'$, then $f(u_j) < f(v) < f(w_j)$. Moreover, $T_1[v_0, v_{j-1}] \subseteq \bigcup_{l=1}^{j-1} B_l$ and $T_2[v_k, v_j] \subseteq \bigcup_{l=j+1}^{k} B_l$. Let $z_1 := v_0, z_2 := v_k, z_3 := u_j$, and $z_4 := w_j$. Clearly, (1)–(3) hold.

So assume that $v = v_j$ for some j, $1 \le j \le k-1$. Let $z_1 := v_0$ and $z_2 := v_k$. Then by Case 2 of Algorithm Numbering g, $g(z_1) < g(v) < g(z_2)$. We will define z_3 and z_4 and prove that (1)–(3) hold. We do this by analyzing how Algorithm Trees chooses the neighbors p_3, p_4 of v_j in the trees T_3, T_4, respectively.

Subcase 2.1. B_j and B_{j+1} are trivial (Subcase 2.1 in Algorithm Trees).

Then Algorithm Trees chooses neighbors p_3, p_4 of v_j in $V(\bar{G}_i)$ with $f(p_3)$ minimum (and hence $f(p_3) < f(p_4)$). If $v_j \in D'$, then by Case 3 of Algorithm Numbering f, we have $f(p_3) < f(v_j) < f(p_4)$. Let $z_3 := p_3$ and $z_4 := p_4$. Clearly, (1)–(3) hold.

Subcase 2.2. B_j is 2-connected and B_{j+1} is trivial (Subcase 2.2 in Algorithm Trees).

- If v_j has no neighbor in $V(\bar{G}_i)$, then Algorithm Trees chooses three neighbors p_1, p_3, p_4 of v_j in B_j such that $T^j_{v_{j-1}}[v_{j-1}, p_1]$, $T^j_{u_j}[u_j, p_3]$, and $T^j_{w_j}[w_j, p_4]$ are disjoint. By construction, $T_1[v_{j-1}, v_j] = T^j_{v_{j-1}}[v_{j-1}, p_1] + \{v_j, v_jp_1\}$, $T_3[u_j, v_j] = T^j_{u_j}[u_j, p_3] + \{v_j, v_jp_3\}$, and $T_4[w_j, v_j] = T^j_{w_j}[w_j, p_4] + \{v_j, v_jp_4\}$. Moreover, $T_1[v_0, v_{j-1}] \subseteq \bigcup_{l=1}^{j-1} B_l$ and $T_2[v_k, v_j] \subseteq \bigcup_{l=j+1}^{k} B_l$. Therefore, $T_1[v_0, v_j]$, $T_2[v_k, v_j]$, $T_3[u_j, v_j]$, and $T_4[w_j, v_j]$ are internally disjoint. If $v_j \in D'$, then by Case 3 of Algorithm Numbering f, we have $j = k-1$ and $f(u_j) < f(v) < f(w_j)$. Let $z_3 := u_j$ and $z_4 := w_j$. Clearly, (1)–(3) hold.
- If v_j has a neighbor in $V(\bar{G}_i)$, then Algorithm Trees chooses a vertex $x \in N_G(v_j) \cap V(\bar{G}_i)$ with $f(x)$ minimum.
  - If $f(x) > f(u_j)$, then the algorithm chooses neighbors p_1, p_3 of v_j in B_j such that $T^j_{v_{j-1}}[v_{j-1}, p_1]$ and $T^j_{u_j}[u_j, p_3]$ are disjoint. By construction, $T_1[v_{j-1}, v_j] = T^j_{v_{j-1}}[v_{j-1}, p_1] + \{v_j, v_jp_1\}$, $T_3[u_j, v_j] = T^j_{u_j}[u_j, p_3] + \{v_j, v_jp_3\}$, and $T_4[x, v_j]$ is induced by the edge xv_j. Moreover, $T_1[v_0, v_{j-1}] \subseteq \bigcup_{l=1}^{j-1} B_l$ and $T_2[v_k, v_j] \subseteq \bigcup_{l=j+1}^{k} B_l$. Therefore, $T_1[v_0, v_j], T_2[v_k, v_j]$, $T_3[u_j, v_j]$, and $T_4[x, v_j]$ are internally disjoint. If $v_j \in D'$, then by Case 3 of Algorithm Numbering f, we have $f(u_j) < f(v) < f(x)$. Let $z_3 := u_j$ and $z_4 := x$. Clearly, (1)–(3) hold.
  - If $f(x) \le f(u_j)$, then Algorithm Trees chooses neighbors p_1, p_4 of v_j in B_j such that $T^j_{v_{j-1}}[v_{j-1}, p_1]$ and $T^j_{w_j}[w_j, p_4]$ are disjoint. By construction, $T_1[v_{j-1}, v_j] = T^j_{v_{j-1}}[v_{j-1}, p_1] + \{v_j, v_jp_1\}$, $T_4[w_j, v_j] = T^j_{w_j}[w_j, p_4] + \{v_j, v_jp_4\}$, and $T_3[x, v_j]$ is induced by the edge xv_j. Moreover, $T_1[v_0, v_{j-1}] \subseteq \bigcup_{l=1}^{j-1} B_l$ and $T_2[v_k, v_j] \subseteq \bigcup_{l=j+1}^{k} B_l$. Therefore, $T_1[v_0, v_j]$, $T_2[v_k, v_j]$, $T_3[x, v_j]$, and $T_4[w_j, v_j]$ are internally disjoint. If $v_j \in D'$, then by Case 3 of Algorithm Numbering f, we have $f(x) < f(v) < f(w_j)$. Let $z_3 := x$ and $z_4 := w_j$. Clearly, (1)–(3) hold.

Subcase 2.3. B_j is trivial and B_{j+1} is 2-connected (Subcase 2.3 in Algorithm Trees).

In this case, if $v_j \in D'$, then $j = 1$ by Case 3 of Algorithm Numbering f. The arguments for the proof are similar to Subcase 2.2, and we indicate only the choice of

z_3 and z_4. In each case below, one can show that (1)–(3) hold for the corresponding choice of z_3, z_4.

- If v_j has no neighbor in $V(\bar{G}_i)$, then let $z_3 := u_{j+1}$ and $z_4 := w_{j+1}$.
- If v_j has a neighbor in $V(\bar{G}_i)$, then Algorithm Trees chooses a vertex $x \in N_G(v_j) \cap V(\bar{G}_i)$ with $f(x)$ minimum.
  - If $f(x) > f(u_{j+1})$, then let $z_3 := u_{j+1}$ and $z_4 := x$.
  - If $f(x) \leq f(u_{j+1})$, then let $z_3 := x$ and $z_4 := w_{j+1}$.

Subcase 2.4. Both B_j and B_{j+1} are 2-connected (Subcase 2.4 in Algorithm Trees).

Since G is 4-connected and $(B_j^+, v_{j-1}, u_j, v_j, w_j)$ and $(B_{j+1}^+, v_j, u_{j+1}, v_{j+1}, w_{j+1})$ are both planar, $v_j \notin N_{B_j}(v_{j-1}) \cup N_{B_{j+1}}(v_{j+1})$. Hence, $v_j \notin D'$ by Case 3 of Algorithm Numbering f. Note that $f(u_j) < f(w_{j+1})$ or $f(u_{j+1}) < f(w_j)$.

- If $f(u_j) < f(w_{j+1})$, then Algorithm Trees chooses neighbors p_1, p_3 of v_j in B_j such that $T^j_{v_{j-1}}[v_{j-1}, p_1]$, $T^j_{u_j}[u_j, p_3]$ are disjoint and neighbors p_2, p_4 of v_j in B_{j+1} such that $T^j_{v_{j+1}}[v_{j+1}, p_2], T^j_{w_{j+1}}[w_{j+1}, p_4]$ are disjoint. By construction, $T_1[v_{j-1}, v_j] = T^j_{v_{j-1}}[v_{j-1}, p_1] + \{v_j, v_jp_1\}$, $T_3[u_j, v_j] = T^j_{u_j}[u_j, p_3] + \{v_j, v_jp_3\}$, $T_2[v_{j+1}, v_j] = T^{j+1}_{v_{j+1}}[v_{j+1}, p_2] + \{v_j, v_jp_2\}$, and $T_4[w_{j+1}, v_j] = T^{j+1}_{w_{j+1}}[w_{j+1}, p_4] + \{v_j, v_jp_4\}$. Moreover, $T_1[v_0, v_{j-1}] \subseteq \bigcup_{l=1}^{j-1} B_l$ and $T_2[v_k, v_{j+1}] \subseteq \bigcup_{l=j+2}^{k} B_l$. Thus, $T_1[v_0, v_j]$, $T_2[v_k, v_j]$, $T_3[u_j, v_j]$, and $T_4[w_{j+1}, v_j]$ are internally disjoint. Let $z_3 := u_j$ and $z_4 := w_{j+1}$. Clearly, (1)–(3) hold.
- If $f(u_j) \geq f(w_{j+1})$, then $f(u_{j+1}) < f(w_j)$, and Algorithm Trees chooses neighbors p_1, p_4 of v_j in B_j such that $T^j_{v_{j-1}}[v_{j-1}, p_1]$ and $T^j_{w_j}[w_j, p_4]$ are disjoint and neighbors p_2, p_3 of v_j in B_{j+1} such that $T^{j+1}_{v_{j+1}}[v_{j+1}, p_2]$ and $T^{j+1}_{u_{j+1}}[u_{j+1}, p_3]$ are disjoint. Let $z_3 := u_{j+1}$ and $z_4 := w_j$. One can show as in the above paragraph that $T_1[v_0, v_j]$, $T_2[v_k, v_j]$, $T_3[u_{j+1}, v_j]$, and $T_4[w_j, v_j]$ are internally disjoint and (1)–(3) hold.

Case 3. $i = t$, or H_i is a down G_{i-1}-chain in G but not an elementary G_{i-1}-chain.

Let $H_i := v_0 B_1 v_1 \dots v_{k-1} B_k v_k$, with $v_0 = v_k = r$ when $i = t$, and $f(v_0) < f(v_k)$ when $i \neq t$. For each 2-connected B_j, let u_j, w_j denote the terminals of B_j^+ other than v_{j-1}, v_j, with $g(u_j) < g(w_j)$, and let $T^j_{v_{j-1}}, T^j_{v_j}, T^j_{u_j}, T^j_{w_j}$ denote the trees rooted, respectively, at v_{j-1}, v_j, u_j, w_j in the independent spanning $\{v_{j-1}, v_j, u_j, w_j\}$-system of B_j^+ computed in Assumption 4.5. This is the same as in Case 3 of Algorithm Trees.

Let $j \in \{1, \dots, k-1\}$. If $i = t$, then by (3) and (4) of Lemma 6.5, $T_3[v_0, v_j] \subseteq \bigcup_{l=1}^{j} B_l$ and $T_4[v_k, v_j] \subseteq \bigcup_{l=j+1}^{k} B_l$. If $i \neq t$, then v_j is a cut vertex of H_i, and hence, by (3) and (4) of Lemma 6.6, $T_3[v_0, v_j] \subseteq \bigcup_{l=1}^{j} B_l$ and $T_4[v_k, v_j] \subseteq \bigcup_{l=j+1}^{k} B_l$.

First, let us consider the case when $v \neq v_j$ for $j = 1, \dots, k-1$. Thus, there exists some j, $1 \leq j \leq k$, such that B_j is 2-connected and $v \in V(B_j) - \{v_{j-1}, v_j\}$. By Case 2 of Algorithm Numbering f, we know that $f(v_0) \leq f(v_{j-1}) < f(v_j) \leq f(v_k)$, and if $v_j \in D'$, then $f(v_0) \leq f(v_{j-1}) < f(v) < f(v_j) \leq f(v_k)$. Furthermore, $T^j_{v_{j-1}}[v_{j-1}, v], T^j_{v_j}[v_j, v], T^j_{u_j}[u_j, v]$, and $T^j_{w_j}[w_j, v]$ are internally disjoint because $\{T^j_{v_{j-1}}, T^j_{v_j}, T^j_{u_j}, T^j_{w_j}\}$ is an independent spanning $\{v_{j-1}, v_j, u_j, w_j\}$-system of B_j^+. By the construction in Case 3 of Algorithm Trees, $T_1[u_j, v] = T^j_{u_j}[u_j, v]$, $T_2[w_j, v] = T^j_{w_j}[w_j, v]$, $T_3[v_{j-1}, v] = T^j_{v_{j-1}}[v_{j-1}, v]$, and $T_4[v_j, v] = T^j_{v_j}[v_j, v]$. Moreover, $T_3[v_0, v_{j-1}] \subseteq \bigcup_{l=1}^{j-1} B_l$ and $T_4[v_k, v_j] \subseteq \bigcup_{l=j+1}^{k} B_l$. Let $z_1 := u_j, z_2 := w_j, z_3 := v_0$, and $z_4 := v_k$. Clearly, (1)–(3) hold.

So assume that $v = v_j$ for some j, $1 \leq j \leq k-1$. Let $z_3 := v_0$ and $z_4 := v_k$. By Case 2 of Algorithm Numbering f, we have $f(z_2) < f(v) < f(z_4)$. We will define z_1 and z_2 and prove that (1)–(3) hold. We do this by analyzing how Algorithm Trees chooses the neighbors p_1, p_2 of v_j in the trees T_1, T_2, respectively.

Subcase 3.1. B_j and B_{j+1} are trivial (Subcase 3.1 in Algorithm Trees).

Then Algorithm Trees chooses neighbors p_1, p_2 of v_j in $V(G_{i-1})$ with $g(p_1)$ minimum (and so $g(p_1) < g(p_2)$). By Subcase 3.3 of Algorithm Numbering g, we have $g(p_1) < g(v) < g(p_2)$. Let $z_1 := p_1$ and $z_2 := p_2$. Clearly, (1)–(3) hold.

Subcase 3.2. B_j is 2-connected and B_{j+1} is trivial (Subcase 3.2 in Algorithm Trees).

- If v_j has no neighbor in $V(G_{i-1})$, then Algorithm Trees chooses three neighbors p_1, p_2, p_3 of v_j in B_j such that $T^j_{u_j}[u_j, p_1]$, $T^j_{w_j}[w_j, p_2]$, and $T^j_{v_{j-1}}[v_{j-1}, p_3]$ are disjoint. By construction, $T_1[u_j, v_j] = T^j_{u_j}[u_j, p_1] + \{v_j, v_jp_1\}$, $T_2[w_j, v_j] = T^j_{w_j}[w_j, p_2] + \{v_j, v_jp_2\}$, and $T_3[v_{j-1}, v_j] = T^j_{v_{j-1}}[v_{j-1}, p_3] + \{v_j, v_jp_3\}$. Moreover, $T_3[v_0, v_{j-1}] \subseteq \bigcup_{l=1}^{j-1} B_l$ and $T_4[v_k, v_j] \subseteq \bigcup_{l=j+1}^{k} B_l$. Therefore, $T_1[u_j, v_j]$, $T_2[w_j, v_j]$, $T_3[v_0, v_j]$, and $T_4[v_k, v_j]$ are internally disjoint. In this case, if $v_j \in D$, then by Case 3 of Algorithm Numbering g, we have $j = k-1$ and $g(u_j) < g(v) < g(w_j)$. Let $z_1 := u_j$ and $z_2 := w_j$. Clearly, (1)–(3) hold.
- If v_j has a neighbor in $V(G_{i-1})$, then Algorithm Trees chooses a vertex $x \in N_G(v_j) \cap V(G_{i-1})$ with $g(x)$ minimum.
  - If $g(x) > g(u_j)$, then the algorithm chooses neighbors p_1, p_3 of v_j in B_j such that $T^j_{u_j}[u_j, p_1]$ and $T^j_{v_{j-1}}[v_{j-1}, p_3]$ are disjoint. By construction, $T_1[u_j, v_j] = T^j_{u_j}[u_j, p_1] + \{v_j, v_jp_1\}$, $T_3[v_{j-1}, v_j] = T^j_{v_{j-1}}[v_{j-1}, p_3] + \{v_j, v_jp_3\}$, and $T_2[x, v_j]$ is induced by the edge xv_j. Moreover, $T_3[v_0, v_{j-1}] \subseteq \bigcup_{l=1}^{j-1} B_l$ and $T_4[v_k, v_j] \subseteq \bigcup_{l=j+1}^{k} B_l$. Therefore, $T_1[u_j, v_j]$, $T_2[x, v_j]$, $T_3[v_0, v_j]$, and $T_4[v_k, v_j]$ are internally disjoint. If $v_j \in D$, then by Case 3 of Algorithm Numbering g, we have $j = k-1$ and $g(u_j) < g(v) < g(x)$. Let $z_1 := u_j$ and $z_4 := x$. Clearly, (1)–(3) hold.
  - If $g(x) \leq g(u_j)$, then Algorithm Trees chooses neighbors p_2, p_3 of v_j in B_j such that $T^j_{w_j}[w_j, p_2]$ and $T^j_{v_{j-1}}[v_{j-1}, p_3]$ are disjoint. By construction, $T_2[w_j, v_j] = T^j_{w_j}[w_j, p_2] + \{v_j, v_jp_2\}$, $T_3[v_{j-1}, v_j] = T^j_{v_{j-1}}[v_{j-1}, p_3] + \{v_j, v_jp_3\}$, and $T_1[x, v_j]$ is induced by the edge xv_j. Moreover, $T_3[v_0, v_{j-1}] \subseteq \bigcup_{l=1}^{j-1} B_l$ and $T_4[v_k, v_j] \subseteq \bigcup_{l=j+1}^{k} B_l$. Therefore, $T_1[x, v_j], T_2[w_j, v_j]$, $T_3[v_0, v_j]$, and $T_4[v_k, v_j]$, are internally disjoint. If $v_j \in D$, then by Case 3 of Algorithm Numbering g, we have $j = k-1$ and $g(x) < g(v) < g(w_j)$. Let $z_1 := x$ and $z_2 := w_j$. Clearly, (1)–(3) hold.

Subcase 3.3. B_j is trivial and B_{j+1} is 2-connected (Subcase 3.3 in Algorithm Trees).

In this case, if $v_j \in D$, then $j = 1$ by Case 3 of Algorithm Numbering g. The arguments for this case are similar to Subcase 3.2, and we indicate only the choice of z_1 and z_2. In each case below, one can show that (1)–(3) hold for the corresponding choice of z_1, z_2.

- If v_j has no neighbor in $V(G_{i-1})$, then let $z_1 := u_{j+1}$ and $z_2 := w_{j+1}$.
- If v_j has a neighbor in $V(G_{i-1})$, then Algorithm Trees chooses a vertex $x \in N_G(v_j) \cap V(G_{i-1})$ with $g(x)$ minimum.
  - If $g(x) > g(u_{j+1})$, then let $z_1 := u_{j+1}$ and $z_2 := x$.
  - If $g(x) \leq g(u_{j+1})$, then let $z_1 := x$ and $z_2 := w_{j+1}$.

Subcase 3.4. B_j and B_{j+1} are 2-connected (Subcase 3.4 in Algorithm Trees).

Since G is 4-connected and $(B_j^+, v_{j-1}, u_j, v_j, w_j)$ and $(B_{j+1}^+, v_j, u_{j+1}, v_{j+1}, w_{j+1})$ are both planar, $v_j \notin N_{B_j}(v_{j-1}) \cup N_{B_{j+1}}(v_{j+1})$. So by Case 3 of Algorithm Numbering g, $v_j \notin D$. Note that $g(u_j) < g(w_{j+1})$ or $g(u_{j+1}) < g(w_j)$.

- If $g(u_j) < g(w_{j+1})$, then Algorithm Trees chooses neighbors p_1, p_3 of v_j in B_j such that $T^j_{u_j}[u_j, p_1]$ and $T^j_{v_{j-1}}[v_{j-1}, p_3]$ are disjoint and neighbors

p_2, p_4 of v_j in B_{j+1} such that $T^{j+1}_{w_{j+1}}[w_{j+1}, p_2]$ and $T^{j+1}_{v_{j+1}}[v_{j+1}, p_4]$ are disjoint. By construction, $T_1[u_j, v_j] = T^j_{u_j}[u_j, p_1] + \{v_j, v_jp_1\}$, $T_3[v_{j-1}, v_j] = T^j_{v_{j-1}}[v_{j-1}, p_3] + \{v_j, v_jp_3\}$, $T_2[w_{j+1}, v_j] = T^{j+1}_{w_{j+1}}[w_{j+1}, p_2] + \{v_j, v_jp_2\}$, and $T_4[v_{j+1}, v_j] = T^{j+1}_{v_{j+1}}[v_{j+1}, p_4] + \{v_j, v_jp_4\}$. Moreover, $T_3[v_0, v_{j-1}] \subseteq \bigcup_{l=1}^{j-1} B_l$ and $T_4[v_k, v_{j+1}] \subseteq \bigcup_{l=j+2}^{k} B_l$. Thus, $T_1[u_j, v_j], T_2[w_{j+1}, v_j], T_3[v_0, v_j]$, and $T_4[v_k, v_j]$ are internally disjoint. Let $z_1 := u_j$ and $z_2 := w_{j+1}$. Clearly, (1)–(3) hold.

- If $g(u_j) \geq g(w_{j+1})$, then $g(u_{j+1}) < g(w_j)$, and Algorithm Trees chooses neighbors p_2, p_3 of v_j in B_j such that $T^j_{w_j}[w_j, p_2]$ and $T^j_{v_{j-1}}[v_{j-1}, p_3]$ are disjoint and neighbors p_1, p_4 of v_j in B_{j+1} such that $T^{j+1}_{u_{j+1}}[u_{j+1}, p_1]$ and $T^{j+1}_{v_{j+1}}[v_{j+1}, p_4]$ are disjoint. Let $z_1 := u_{j+1}$ and $z_2 := w_j$. One can show as in the above paragraph that $T_1[u_{j+1}, v_j]$, $T_2[w_j, v_j]$, $T_3[v_0, v_j]$, and $T_4[v_k, v_j]$ are internally disjoint and (1)–(3) hold.

Case 4. H_i is a triangle G_{i-1}-chain in G.

Let $I(H_i) := \{v_1, v_2, v_3\}$, let $y_1, y_2, y_3 \in V(\bar{G}_i)$ such that $y_1v_1, y_2v_2, y_3v_3 \in E(G)$, and let v_jx_j $(j = 1, 2, 3)$ be the legs of H_i. Assume that v_1, v_2, v_3 are labeled so that $g(x_1) < g(x_2) < g(x_3)$.

The proof can be done by inspecting a small number of cases (Case 4 in Algorithm Trees) and using Lemma 6.7 and Case 4 of Algorithm Numbering g and Algorithm Numbering f. For the sake of completeness, we list for each case the choice for z_1, z_2, z_3, and z_4. The verification that they satisfy (1)–(3) is straightforward, and we omit it.

- If $f(y_1) < f(y_2)$ and $f(y_1) < f(y_3)$, then let $z_2 := x_3$ and $z_3 := y_1$.
  If $v = v_1$, then let $z_1 := x_1$ and $z_4 := y_2$.
  If $v = v_2$, then let $z_1 := x_2$ and $z_4 := y_2$.
  If $v = v_3$, then let $z_1 := x_2$ and $z_4 := y_3$.
- If $f(y_2) < f(y_1)$ and $f(y_2) < f(y_3)$, then let $z_2 := x_3$ and $z_3 := y_2$.
  If $v = v_1$, then let $z_1 := x_1$ and $z_4 := y_1$.
  If $v = v_2$, then let $z_1 := x_2$ and $z_4 := y_1$.
  If $v = v_3$, then let $z_1 := x_1$ and $z_4 := y_3$.
- If $f(y_3) < f(y_1) < f(y_2)$, then let $z_2 := x_3$ and $z_4 := y_2$.
  If $v = v_1$, then let $z_1 := x_1$ and $z_3 := y_1$.
  If $v = v_2$, then let $z_1 := x_2$ and $z_3 := y_1$.
  If $v = v_3$, then let $z_1 := x_1$ and $z_3 := y_3$.
- If $f(y_3) < f(y_2) < f(y_1)$, then let $z_2 := x_3$ and $z_4 := y_1$.
  If $v = v_1$, then let $z_1 := x_1$ and $z_3 := y_2$.
  If $v = v_2$, then let $z_1 := x_2$ and $z_3 := y_2$.
  If $v = v_3$, then let $z_1 := x_2$ and $z_3 := y_3$.

This completes the proof of Lemma 6.11. ∎

LEMMA 6.12. *Let $i \in \{1, \ldots, t-1\}$. Then for any $u, v \in D_i$, with $g(u) < g(v)$, $T_1[r, u]$ and $T_2[r, v]$ are internally disjoint paths in G_i.*

Proof. We will prove the lemma by induction on i. The basis of induction is $i = 0$ with $D_0 := \{r\}$ and $G_0 := (\{r\}, \emptyset)$. So assume that $i > 0$ and the lemma holds for $i - 1$. We consider the four cases of Algorithm Numbering g.

Case 1. H_i is an elementary G_{i-1}-chain in G.

Let $H_i := v_0B_1v_1B_2v_2$, with $g(v_0) < g(v_2)$. By (1) of Lemma 6.2, $E(T_1 \cap H_i) = \{v_0v_1\}$ and $E(T_2 \cap H_i) = \{v_1v_2\}$. Recall that $D_i = D_{i-1} \cup \{v_1\}$.

If $u, v \in D_{i-1}$ and $g(u) < g(v)$, then by the induction hypothesis, $T_1[r, u]$ and $T_2[r, v]$ are internally disjoint paths in G_{i-1}. Thus, it suffices to prove the following:

for any $u, v \in D_i$, with $g(u) < g(v)$ and $v_1 \in \{u, v\}$, $T_1[r, u]$ and $T_2[r, v]$ are internally disjoint paths in G_i.

Assume first that $u = v_1$. Then $v = v \in D_{i-1}$. Since $g(v_0) < g(v_1) < g(v)$, it follows from the induction hypothesis that $T_1[r, v_0]$ and $T_2[r, v]$ are internally disjoint paths in G_{i-1}. Therefore, $T_1[r, v_1] = T_1[r, v_0] + \{v_1, v_1v_0\}$ and $T_2[r, v]$ are internally disjoint paths in G_i.

Now suppose $v = v_1$. Then $u \in D_{i-1}$. Since $g(u) < g(v_1) < g(v_2)$, it follows from the induction hypothesis that $T_1[r, u]$ and $T_2[r, v_2]$ are internally disjoint paths in G_{i-1}. Therefore, $T_1[r, u]$ and $T_2[r, v_1] = T_2[r, v_2] + \{v_1, v_1v_2\}$ are internally disjoint paths in G_i.

Case 2. $i = 1$, or H_i is an up G_{i-1}-chain in G but not an elementary G_{i-1}-chain in G.

Let $H_i := v_0B_1v_1 \dots v_{k-1}B_kv_k$, with $v_0 = v_k = r$ when $i = 1$, and $g(v_0) < g(v_k)$ when $i \neq 1$. For each 2-connected B_j, let u_j, w_j denote the terminals of B_j^+ other than v_{j-1}, v_j, with $f(u_j) < f(w_j)$, and let $T^j_{v_{j-1}}, T^j_{v_j}, T^j_{u_j}, T^j_{w_j}$ denote the trees rooted, respectively, at v_{j-1}, v_j, u_j, w_j in the independent spanning $\{v_{j-1}, v_j, u_j, w_j\}$-system of B_j^+ computed in Assumption 4.5. This is the same as in Case 2 of Algorithm Trees. Let $u, v \in D_i$ with $g(u) < g(v)$.

If $u, v \in D_{i-1}$, then by the induction hypothesis, $T_1[r, u]$ and $T_2[r, v]$ are internally disjoint paths in G_{i-1}.

If $u \in D_i - D_{i-1}$ and $v \in D_{i-1}$, then by the construction in Case 2 of Algorithm Numbering g, $g(v_0) < g(u) < g(v)$. By the induction hypothesis, $T_1[r, v_0]$ and $T_2[r, v]$ are internally disjoint paths in G_{i-1}. Since $T_1[v_0, u]$ is a path in $H_i - v_k$ by (1) of Lemma 6.3 when $i = 1$, or by (1) of Lemma 6.4 when $i \neq 1$, $T_1[r, v]$ and $T_2[r, v]$ are internally disjoint paths in G_i.

If $u \in D_{i-1}$ and $v \in D_i - D_{i-1}$, then by the construction in Case 2 of Algorithm Numbering g, $g(u) < g(v) < g(v_k)$. By the induction hypothesis, $T_1[r, u]$ and $T_2[r, v_k]$ are internally disjoint paths in G_{i-1}. Since $T_2[v_k, v]$ is a path in $H_i - v_0$ by (2) of Lemma 6.3 when $i = 1$, or by (2) of Lemma 6.4 when $i \neq 1$, $T_1[r, v]$ and $T_2[r, v]$ are internally disjoint paths in G_i.

So we may assume that $u, v \in D_i - D_{i-1}$. Let g^i denote the function g at the start of iteration i of Algorithm Numbering g (when it examines H_i in Case 2). Recall that for each $j = 1, \dots, k$ the algorithm computes a sequence σ_j as follows. If B_j is 2 connected, then σ_j is a $(T^j_{v_{j-1}}, T^j_{v_j})$-ordering of $N_{B_j^+}(\{u_j, w_j\}) - \{v_{j-1}, v_j\}$. If B_j is trivial, then σ_j is the empty sequence. Moreover, the algorithm extends g^i to $\sigma := \sigma_1, v_1, \sigma_2, \dots, v_{k-1}, \sigma_k$ from v_0 and set $D_i := D_{i-1} \cup \{\sigma\}$. Thus, $u, v \in \{\sigma\}$. Note that since $g(u) < g(v)$, u precedes v in the sequence σ.

First, suppose that there exists no $j \in \{1, \dots, k-1\}$ such that $u, v \in \{\sigma_j\}$. Hence, there is some $j \in \{1, \dots, k-1\}$ such that either

- u appears in the sequence $\sigma_1, v_1, \dots, \sigma_j, v_j$ and v appears in the sequence $\sigma_{j+1}, v_{j+1}, \dots, v_{k-1}, \sigma_k$ or
- u appears in the sequence $\sigma_1, v_1, \dots, \sigma_j$ and v appears in the sequence $v_j, \sigma_{j+1}, v_{j+1}, \dots, v_{k-1}, \sigma_k$.

By (1) and (2) of Lemma 6.3 when $i = 1$ or by (1) and (2) of Lemma 6.4 when $i \neq 1$, $T_1[v_0, u]$ and $T_2[v_k, v]$ are internally disjoint paths in H_i, and by the induction hypothesis, $T_1[r, v_0]$ and $T_2[r, v_k]$ are internally disjoint paths in G_{i-1}. Therefore, $T_1[r, u]$ and $T_2[r, v]$ are internally disjoint paths in G_i.

So, we may assume that there exists some $j \in \{1, \dots, k-1\}$ such that u, v are in the sequence σ_j. Since the sequence σ_j is $(T^j_{v_{j-1}}, T^j_{v_j})$-ordered and u precedes v in

σ_j, $T^j_{v_{j-1}}[v_{j-1}, u]$ and $T^j_{v_j}[v_j, v]$ are disjoint. By the construction in Algorithm Trees, $T_1[v_{j-1}, u] = T^j_{v_{j-1}}[v_{j-1}, u]$ and $T_2[v_j, v] = T^j_{v_j}[v_j, v]$. By (1) and (2) of Lemma 6.3 when $i = 1$, or by (1) and (2) of Lemma 6.4 when $i \neq 1$, $T_1[v_0, v_{j-1}]$ and $T_2[v_k, v_j]$ are internally disjoint paths in H_i. Moreover, by the induction hypothesis, $T_1[r, v_0]$ and $T_2[r, v_k]$ are internally disjoint paths in G_{i-1}. Therefore, $T_1[r, u]$ and $T_2[r, v]$ are internally disjoint paths in G_i.

Case 3. $i = t$, or H_i is a down G_{i-1}-chain in G but not an elementary G_{i-1}-chain in G.

Let $H_i := v_0 B_1 v_1 \ldots v_{k-1} B_k v_k$, with $v_0 = v_k = r$ when $i = t$, and $f(v_0) < f(v_k)$ when $i \neq t$. For each 2-connected B_j, let u_j, w_j denote the terminals of B_j^+ other than v_{j-1}, v_j, with $g(u_j) < g(w_j)$, and let $T^j_{v_{j-1}}, T^j_{v_j}, T^j_{u_j}, T^j_{w_j}$ denote the trees rooted, respectively, at v_{j-1}, v_j, u_j, w_j in the independent spanning $\{v_{j-1}, v_j, u_j, w_j\}$-system of B_j^+ computed in Assumption 4.5. This is the same as in Case 3 of Algorithm Trees. Let $u, v \in D_i$ with $g(u) < g(v)$. Recall that $D_i = D_{i-1} \cup N_{B_1}(v_0) \cup N_{B_k}(v_k)$.

If $u, v \in D_{i-1}$, then by the induction hypothesis, $T_1[r, u]$ and $T_2[r, v]$ are internally disjoint paths in $G_{i-1} \subset G_i$.

If $u \in D_i - D_{i-1}$ and $v \in D_{i-1}$, then $u \in N_{B_1}(v_0) \cup N_{B_k}(v_k)$. By (1) and (3) of Lemma 6.11, there exists $z_1 \in V(G_{i-1})$ such that $g(z_1) < g(u)$ and $V(T_1[z_1, u] - z_1) \subseteq I(H_i)$. Since $z_1, v \in D_{i-1}$ and $g(z_1) < g(u) < g(v)$, it follows from the induction hypothesis that $T_1[r, z_1]$ and $T_2[r, v]$ are internally disjoint paths in G_{i-1}. Therefore, $T_1[r, u]$ and $T_2[r, v]$ are internally disjoint paths in G_i.

If $u \in D_{i-1}$ and $v \in D_i - D_{i-1}$, then $v \in N_{B_1}(v_0) \cup N_{B_k}(v_k)$. By (1) and (3) of Lemma 6.11, there exists $z_2 \in V(G_{i-1})$ such that $g(v) < g(z_2)$ and $V(T_2[z_2, v] - z_2) \subseteq I(H_i)$. Since $z_2, u \in D_{i-1}$ and $g(u) < g(v) < g(z_2)$, it follows from the induction hypothesis that $T_1[r, u]$ and $T_2[r, z_2]$ are internally disjoint paths in G_{i-1}. Therefore, $T_1[r, u]$ and $T_2[r, v]$ are internally disjoint paths in G_i.

So, we need only to prove the case when $u, v \in D_i - D_{i-1}$. Let g^i denote the function g at the start of iteration i of Algorithm Numbering g (when it examines H_i in Case 3). Now we consider the three subcases of Case 3 of Algorithm Numbering g.

Subcase 3.1. $k = 1$ (thus, B_1 is 2-connected).

Since $(B_1^+, v_0, u_1, v_1, w_1)$ is planar and G is 4-connected, $v_0, v_1 \notin N_{B_1}(v_0) \cup N_{B_1}(v_1)$. Hence, in this case, $D_i - D_{i-1} = N_{B_1^+}(\{v_0, v_1\}) - \{u_1, w_1\} = N_{B_1}(v_0) \cup N_{B_1}(v_1)$. Moreover, Algorithm Numbering g produces a $(T^1_{u_1}, T^1_{w_1})$-ordering σ of $N_{B_1^+}(\{v_0, v_1\}) - \{u_1, w_1\}$ and extends g^i to σ from u_1.

Let $u, v \in D_i - D_{i-1}$, with $g(u) < g(v)$. Then both u and v are in the sequence σ, and u precedes v in σ. Since σ is $(T^1_{u_1}, T^1_{w_1})$-ordered, $T^1_{u_1}[u_1, u]$ and $T^1_{w_1}[w_1, v]$ are disjoint. By the construction in Case 3 of Algorithm Trees, $T_1[u_1, u] = T^1_{u_1}[u_1, u]$ and $T_2[w_1, v] = T^1_{w_1}[w_1, v]$. By the induction hypothesis, $T_1[r, u_1]$ and $T_2[r, w_1]$ are internally disjoint paths in G_{i-1}. Therefore, $T_1[r, u]$ and $T_2[r, v]$ are internally disjoint paths in G_i.

Subcase 3.2. $k = 2$, and B_1 or B_2 is trivial.

By symmetry we assume that B_2 is trivial (the arguments are analogous if B_1 is trivial). Note that B_1 is 2-connected because H_i is not an elementary G_{i-1}-chain in G. Thus, $D_i - D_{i-1} = N_{B_1}(v_0) \cup \{v_1\}$.

- If v_1 has no neighbor in $V(G_{i-1})$, then Algorithm Numbering g chooses neighbors q_1, q_2, q_3 of v_1 in B_1 such that $T^1_{u_1}[u_1, q_1], T^1_{v_0}[v_0, q_2]$, and $T^1_{w_1}[w_1, q_3]$ are disjoint and then computes a $(T^1_{u_1} + \{v_1, v_1q_1\}, T^1_{w_1} + \{v_1, v_1q_3\})$-ordering σ of $N_{B_1}(v_0) \cup \{v_1\}$ in $B_1^+ \cup B_2$ (recall that $(B_1^+ \cup B_2, v_0, u_1, v_2, w_1)$ is planar). Then Algorithm Numbering g extends g^i to σ from u_1.

Let $u, v \in D_i - D_{i-1}$, with $g(u) < g(v)$. Then both u and v are in the sequence σ, and u precedes v in σ.

Let us consider first the case when $u \neq v_1$ and $v \neq v_1$. Thus, $u, v \in N_{B_1}(v_0)$. Since σ is $(T^1_{u_1} + \{v_1, v_1q_1\}, T^1_{w_1} + \{v_1, v_1q_3\})$-ordered and u precedes v in σ, $T^1_{u_1}[u_1, u]$ and $T^1_{w_1}[w_1, v]$ are disjoint. By construction (Case 3 of Algorithm Trees), $T_1[u_1, u] = T^1_{u_1}[u_1, u]$ and $T_2[w_1, v] = T^1_{w_1}[w_1, v]$. By the induction hypothesis, $T_1[r, u_1]$ and $T_2[r, w_1]$ are internally disjoint paths in G_{i-1}. Therefore, $T_1[r, u]$ and $T_2[r, v]$ are internally disjoint paths in G_i.

Now suppose that $u = v_1$. Since σ is $(T^1_{u_1} + \{v_1, v_1q_1\}, T^1_{w_1} + \{v_1, v_1q_3\})$-ordered and u precedes v in σ, $T^1_{u_1}[u_1, q_1] + \{v_1, v_1q_1\}$ and $T^1_{w_1}[w_1, v]$ are disjoint. By construction (Case 3 of Algorithm Trees), $T_1[u_1, v_1] = T^1_{u_1}[u_1, q_1] + \{v_1, v_1q_1\}$ and $T_2[w_1, v] = T^1_{w_1}[w_1, v]$. By the induction hypothesis, $T_1[r, u_1]$ and $T_2[r, w_1]$ are internally disjoint paths in G_{i-1}. Therefore, $T_1[r, u]$ and $T_2[r, v]$ are internally disjoint paths in G_i.

So assume $v = v_1$. Since σ is $(T^1_{u_1} + \{v_1, v_1q_1\}, T^1_{w_1} + \{v_1, v_1q_3\})$-ordered and u precedes v in σ, $T^1_{u_1}[u_1, u]$ and $T^1_{w_1}[w_1, q_3] + \{v_1, v_1q_3\}$ are disjoint. By construction (Case 3 of Algorithm Trees), $T_1[u_1, u] = T^1_{u_1}[u_1, u]$ and $T_2[w_1, v_1] = T^1_{w_1}[w_1, q_3] + \{v_1, v_1q_3\}$. By the induction hypothesis, $T_1[r, u_1]$ and $T_2[r, w_1]$ are internally disjoint paths in G_{i-1}. Therefore, $T_1[r, u]$ and $T_2[r, v]$ are internally disjoint paths in G_i.

- If v_1 has a neighbor in $V(G_{i-1})$, then Algorithm Numbering g chooses a vertex $x \in (N_G(v_1) \cap V(G_{i-1}))$ with $g^i(x)$ minimum and computes a $(T^1_{u_1}, T^1_{w_1})$-ordering σ of $N_{B_1^+}(v_0) - \{u_1, w_1\}$.

  Let $u, v \in D_i - D_{i-1}$, with $g(u) < g(v)$.

  Let us consider first the case when $u \neq v_1$ and $v \neq v_1$. Then both u and v are in σ, and u precedes v in σ. Since σ is $(T^1_{u_1}, T^1_{w_1})$-ordered and u precedes v in σ, $T^1_{u_1}[u_1, u]$ and $T^1_{w_1}[w_1, v]$ are disjoint. By construction (Case 3 of Algorithm Trees), $T_1[u_1, u] = T^1_{u_1}[u_1, u]$ and $T_2[w_1, v] = T^1_{w_1}[w_1, v]$. By the induction hypothesis, $T_1[r, u_1]$ and $T_2[r, w_1]$ are internally disjoint paths in G_{i-1}. Therefore, $T_1[r, u]$ and $T_2[r, v]$ are internally disjoint paths in G_i.

  Now suppose that $u = v_1$. Thus, v is in the sequence σ. Recall how Algorithm Numbering g extends g^i in Subcase 3.2 of Algorithm Numbering g.

  If $g(x) > g(u_1)$, then $g^i(x) > g^i(u_1)$, and Algorithm Numbering g extends g^i to σ, v_1 from u_1. But then $g(v) < g(v_1) = g(u)$, contradicting the assumption that $g(u) < g(v)$.

  If $g(x) \leq g(u_1)$, then $g^i(x) \leq g^i(u_1)$, and Algorithm Numbering g extends g^i to v_1, σ from x. By construction (Subcase 3.2 of Algorithm Trees with $j = 1$), $xv_1 \in E(T_1)$ and $T_2[w_1, v] = T^1_{w_1}[w_1, v]$. Since $g(x) < g(w_1)$, by the induction hypothesis, $T_1[r, x]$ and $T_2[r, w_1]$ are internally disjoint paths in G_{i-1}. Therefore, $T_1[r, v_1]$ and $T_2[r, v]$ are internally disjoint paths in G_i.

  The case $v = v_1$ can be treated analogously ($g(x) \leq g(u_1)$ cannot occur).

Subcase 3.3. $k \geq 3$, or $k = 2$ and both B_1, B_2 are 2-connected.

In this case, $D_i - D_{i-1} = N_{B_1}(v_0) \cup N_{B_k}(v_k)$. Let $u, v \in D_i - D_{i-1}$ with $g(u) < g(v)$.

Let us consider first the case when $u, v \in N_{B_1}(v_0)$. Thus, B_1 is 2-connected, and Algorithm Numbering g (Subcase 3.3) computes a $(T^1_{u_1}, T^1_{w_1})$-ordering σ of $N_{B_1^+}(v_0) - \{u_1, w_1\} = N_{B_1}(v_0)$ and extends g^i to σ from u_1. Thus, $g(u_1) < g(u) < g(v)$. Since σ is $(T^1_{u_1}, T^1_{w_1})$-ordered and u precedes v in σ, $T^1_{u_1}[u_1, u]$ and $T^1_{w_1}[w_1, v]$ are disjoint. By construction (Case 3 of Algorithm trees), $T_1[u_1, u] = T^1_{u_1}[u_1, u]$ and

$T_2[w_1, v] = T^1_{w_1}[w_1, v]$. Since $g(u_1) < g(w_1)$ and by the induction hypothesis, $T_1[r, u_1]$ and $T_2[r, w_1]$ are internally disjoint paths in G_{i-1}. Therefore, $T_1[r, u]$ and $T_2[r, v]$ are internally disjoint paths in G_i.

Suppose, now, that $u, v \in N_{B_k}(v_k)$. Then B_k is 2-connected, and Algorithm Numbering g (Subcase 3.3) computes a $(T^k_{u_k}, T^k_{w_k})$-ordering of $N_{B^+_k}(v_k) - \{u_k, w_k\} = N_{B_k}(v_k)$ and extends g^i to σ from u_k. Thus, $g(u_k) < g(u) < g(v)$. Since σ is $(T^k_{u_k}, T^k_{w_k})$-ordered and u precedes v in σ, $T^k_{u_k}[u_k, u]$ and $T^k_{w_k}[w_k, v]$ are disjoint. By construction (Case 3 of Algorithm Trees), $T_1[u_k, u] = T^k_{u_k}[u_k, u]$ and $T_2[w_k, v] = T^k_{w_k}[w_k, v]$. Since $g(u_k) < g(w_k)$ and by the induction hypothesis, $T_1[r, u_k]$ and $T_2[r, w_k]$ are internally disjoint paths in G_{i-1}. Therefore, $T_1[r, u]$ and $T_2[r, v]$ are internally disjoint paths in G_i.

So we may assume that $u \in N_{B_1}(v_0)$ and $v \in N_{B_k}(v_k)$, or $u \in N_{B_k}(v_k)$ and $v \in N_{B_1}(v_0)$. By symmetry, assume that $u \in N_{B_1}(v_0)$ and $v \in N_{B_k}(v_k)$. We will prove that there exist vertices $z_1, z_2 \in V(G_{i-1})$, with $g(z_1) < g(z_2)$, such that $T_1[z_1, u]$ and $T_2[z_2, v]$ are internally disjoint paths in G, $V(T_1[z_1, u] - z_1) \subseteq I(H_i)$, and $V(T_2[z_2, v] - z_2) \subseteq I(H_i)$.

Consider the following cases for u and B_1.

- B_1 is 2-connected. Then, by construction in Algorithm Trees, $T_1[u_1, u] = T^1_{u_1}[u_1, u]$, and let $z_1 := u_1$.
- B_1 is trivial. Thus, $u = v_1$. If B_2 is trivial, then by construction in Subcase 3.1 of Algorithm Trees (with $j = 1$), there exists a neighbor p_1 of v_1 in $V(G_{i-1})$ such that $g(p_1)$ is minimum and $p_1 v_1 \in E(T_1)$. In this case, let $z_1 := p_1$.
  So assume that B_2 is 2-connected.
  - If v_1 has no neighbor in $V(G_{i-1})$, then by construction in Subcase 3.3 (with $j = 1$) of Algorithm Trees, there exists a neighbor p_1 of v_1 in B_2 such that $T_1[u_2, v_1] = T^2_{u_2}[u_2, p_1] + \{v_1, v_1 p_1\}$. In this case, let $z_1 := u_2$.
  - If v_1 has a neighbor in $V(G_{i-1})$, then Algorithm Trees in Subcase 3.3 (with $j = 1$) chooses $x \in N_G(v_1) \cap V(G_{i-1})$ with $g(x)$ minimum. If $g(x) > g(u_2)$, then by construction there exists a neighbor p_1 of v_1 in B_2 such that $T_1[u_2, v_1] = T^2_{u_2}[u_2, p_1] + \{v_1, v_1 p_1\}$. In this case, let $z_1 := u_2$. If $g(x) \le g(u_2)$, then $x v_1 \in E(T_1)$. In this case, let $z_1 := x$.

Consider the analogous cases for v and B_k.

- B_k is 2-connected. Then by construction in Algorithm Trees, $T_2[w_k, v] = T^k_{w_k}[w_k, v]$, and let $z_2 := w_k$.
- B_k is trivial. Thus, $v = v_{k-1}$. If B_{k-1} is trivial, then by construction in Subcase 3.1 of Algorithm Trees (with $j = k-1$), there exists a neighbor p_2 of v_{k-1} in $V(G_{i-1})$ such that $g(p_2)$ is not minimum and $p_2 v_{k-1} \in E(T_2)$. In this case, let $z_2 := p_2$.
  So assume that B_{k-1} is 2-connected.
  - If v_{k-1} has no neighbor in $V(G_{i-1})$, then by construction in Subcase 3.2 (with $j = k-1$) of Algorithm Trees, there exists a neighbor p_2 of v_{k-1} in B_{k-1} such that $T_2[w_{k-1}, v_{k-1}] = T^{k-1}_{w_{k-1}}[w_{k-1}, p_2] + \{v_{k-1}, v_{k-1} p_2\}$. In this case, let $z_2 := w_{k-1}$.
  - If v_{k-1} has a neighbor in $V(G_{i-1})$, then Algorithm Trees in Subcase 3.2 chooses $x \in N_G(v_{k-1}) \cap V(G_{i-1})$ with $g(x)$ minimum. If $g(x) > g(u_{k-1})$, then $x v_{k-1} \in E(T_2)$. In this case, let $z_2 := x$. If $g(x) \le g(u_{k-1})$, then by construction there exists a neighbor p_2 of v_{k-1} in B_{k-1} such that $T_2[w_{k-1}, v_{k-1}] = T^{k-1}_{w_{k-1}}[w_{k-1}, p_2] + \{v_{k-1}, v_{k-1} p_2\}$. In this case, let $z_2 := w_{k-1}$ (this is the same as in the previous paragraph).

So $T_1[z_1, u]$ either is contained in B_1^+, or is contained in B_2^+, or is induced by a single edge. Hence, $g(z_1) < g(u)$. Similarly, $T_2[z_2, v]$ either is contained in B_k^+, or is contained in B_{k-1}^+, or is induced by a single edge. So $g(v) < g(z_2)$. Since $g(u) < g(v)$, $g(z_1) < g(z_2)$.

Note that if $k = 3$, B_2 is 2-connected, and both paths $T_1[z_1, u]$ and $T_2[z_2, v]$ are contained in B_2^+, then $u = v_1$, $v = v_2 = v_{k-1}$, $T_1[u_2, u] = T^2_{u_2}[u_2, p_1] + \{v_1, v_1p_1\}$ for some neighbor p_1 of v_1 in B_2, and $T_2[w_2, v] = T^2_{w_2}[w_2, p_2] + \{v_2, v_2p_2\}$ for some neighbor p_2 of v_2 in B_2. In this case, since u, v are $(T^2_{u_2}, T^2_{w_2})$-ordered, $T_1[u_2, u]$ and $T_2[w_2, v]$ are disjoint.

Therefore, since $k \geq 3$, it is not hard to see that $T_1[z_1, u]$ and $T_2[z_2, v]$ are disjoint paths in G, $V(T_1[z_1, u] - z_1) \subseteq I(H_i)$, and $V(T_2[z_2, v] - z_2) \subset I(H_i)$.

Since $g(z_1) < g(z_2)$, by the induction hypothesis, $T_1[r, z_1]$ and $T_2[r, z_2]$ are internally disjoint paths in G_{i-1}. Therefore, $T_1[r, u]$ and $T_2[r, v]$ are internally disjoint paths in G_i.

Case 4. H_i is a triangle G_{i-1}-chain in G.

By Algorithm Numbering g, $D_i - D_{i-1} = \{v_1, v_2, v_3\}$ and $g(v_1) < g(v_2) < g(v_3)$. Thus, it suffices to show that the following pairs are internally disjoint: $T_1[r, v_1]$ and $T_2[r, v_2]$, $T_1[r, v_2]$ and $T_2[r, v_3]$, and $T_1[r, v_1]$ and $T_2[r, v_3]$. This can be done by inspecting Case 4 of Algorithm Trees. ∎

Recall that Algorithm Numbering f with input $\mathcal{C} := (H_1, \ldots, H_t)$ computes a numbering f and sets $D'_{t+1}, D'_t, D'_{t-1}, \ldots, D'_2$. The next lemma can be proved, analogously to Lemma 6.12. We give only some detail for Case 4, as f and g are not symmetric in that case.

LEMMA 6.13. *Let $i \in \{1, \ldots, t\}$. Then for any $u, v \in D'_i$ with $f(u) < f(v)$, $T_3[r, u]$ and $T_4[r, v]$ are internally disjoint paths in $\bar{G}_i$.*

Proof. We use the notation in the proof of Lemma 6.12 and assume H_i is a triangle G_{i-1}-chain in G. By inspecting Case 4 of Algorithm Numbering f and Algorithm Trees, we have the following.

- If $f(y_1) < f(y_2)$ and $f(y_1) < f(y_3)$, then $f(v_1) < f(v_2) < f(v_3)$. So we can show that $T_3[r, v_1]$ and $T_4[r, v_2]$ are internally disjoint, $T_3[r, v_1]$ and $T_4[r, v_3]$ are internally disjoint, and $T_3[r, v_2]$ and $T_4[r, v_3]$ are internally disjoint.
- If $f(y_2) < f(y_1)$ and $f(y_2) < f(y_3)$, then $f(v_2) < f(v_1) < f(v_3)$. So we can show that $T_3[r, v_2]$ and $T_4[r, v_1]$ are internally disjoint, $T_3[r, v_2]$ and $T_4[r, v_3]$ are internally disjoint, and $T_3[r, v_1]$ and $T_4[r, v_3]$ are internally disjoint.
- If $f(y_3) < f(y_1) < f(y_2)$, then $f(v_3) < f(v_1) < f(v_2)$. So we can show that $T_3[r, v_3]$ and $T_4[r, v_1]$ are internally disjoint, $T_3[r, v_3]$ and $T_4[r, v_2]$ are internally disjoint, and $T_3[r, v_1]$ and $T_4[r, v_2]$ are internally disjoint.
- If $f(y_3) < f(y_2) < f(y_1)$, then $f(v_3) < f(v_2) < f(v_1)$. So we can show that $T_3[r, v_3]$ and $T_4[r, v_2]$ are internally disjoint, $T_3[r, v_3]$ and $T_4[r, v_1]$ are internally disjoint, and $T_3[r, v_2]$ and $T_4[r, v_1]$ are internally disjoint. ∎

THEOREM 6.14. *Given a 4-connected graph G, $r \in V(G)$, and a nonseparating chain decomposition $\mathcal{C} := (H_1, \ldots, H_t)$ of G rooted at r, Algorithm Trees computes four independent spanning trees rooted at r.*

Proof. By Corollary 6.10, T_1, T_2, T_3, T_4 are spanning trees of G. Let us prove that they are independent with r as root. Let $v \in V(G) - \{r\}$. Suppose that v is an internal vertex of a good chain H_i in the decomposition $\mathcal{C}$. By Lemma 6.11 there exist $z_1, z_2, z_3, z_4 \in V(G)$ such that

(i) $z_1, z_2 \in V(G_{i-1})$, and either $g(z_1) < g(z_2)$ or $z_1 = z_2 = r$,
(ii) $z_3, z_4 \in V(\bar{G}_i)$, and either $f(z_3) < f(z_4)$ or $z_3 = z_4 = r$, and

(iii) $T_i[z_i, v]$, $i = 1, 2, 3, 4$, are internally disjoint paths and $V(T_i[z_i, v] - z_i) \subseteq I(H_i)$.

By Lemma 6.12, if $g(z_1) < g(z_2)$, then $T_1[r, z_1]$ and $T_2[r, z_2]$ are internally disjoint paths in G_{i-1}. Obviously, the same holds if $z_1 = z_2 = r$. Similarly, by Lemma 6.13, if $f(z_3) < f(z_4)$, then $T_3[r, z_3]$ and $T_4[r, z_4]$ are internally disjoint paths in $\bar{G}_i$, and the same holds if $z_3 = z_4 = r$. Therefore, $T_1[r, v], T_2[r, v], T_3[r, v]$, and $T_4[r, v]$ are internally disjoint. Hence, T_1, T_2, T_3, and T_4 are independent spanning trees of G rooted at r. ◻

LEMMA 6.15. *Algorithm Trees runs in $O(|V(G)|^3)$ time.*

Proof. By Lemmas 4.6 and 4.7, given $\mathcal{C}$ we can compute numberings g and f in $O(|V(G)|^3)$ time. By Theorem 3.2 we can compute independent spanning systems for all planar sections in $\mathcal{C}$ in $O(|V(G)| + |E(G)|)$ time.

We will show that at each iteration the time spent by Algorithm Trees is $O(|V(G)|^2)$ time. Since the number of iterations is at most $|V(G)|$, this implies the result.

Suppose we are at iteration i of Algorithm Trees.

One can see easily that if Case 1 or Case 4 occurs, then Algorithm Trees uses constant time. Thus, we may assume that Case 2 or Case 3 occurs.

Suppose that Case 2 occurs. The initial updating of T_1, T_2, T_3, T_4 (before Subcases 2.1–2.4 are dealt with) can be done in $O(|V(G)|)$ time. Then for each $j \in \{1, \dots, k\}$ the algorithm inserts v_j into the subgraphs T_1, T_2, T_3, T_4 according to Subcases 2.1–2.4. One can see that Subcase 2.1 can be executed in $O(1)$ time. In the other cases, the algorithm has to solve one of the following problems (at most twice).

(1) Given a planar graph (B, v', u, v, w) and an independent spanning $\{v', u, v, w\}$-system $\{T_{v'}, T_u, T_v, T_w\}$ of B (with $T_{v'}, T_u, T_v, T_w$ rooted, respectively, at v', u, v, w), find three neighbors p_1, p_2, p_3 of v in B such that $T_{v'}[v', p_1], T_u[u, p_2]$, and $T_w[w, p_3]$ are disjoint.

(2) Given a planar graph (B, v', u, v, w) and an independent spanning $\{v', u, v, w\}$-system $\{T_{v'}, T_u, T_v, T_w\}$ of B (with $T_{v'}, T_u, T_v, T_w$ rooted, respectively, at v', u, v, w), find two neighbors p_1, p_2 of v in B such that $T_{v'}[v', p_1]$ and $T_u[u, p_2]$ are disjoint.

By Lemmas 3.6 and 3.7, both problems can be solved in $O(|V(B)|)$ time. Thus, it is not hard so see that the time spent by Algorithm Trees in Case 2 is $O(|V(G)|^2)$.

Case 3 is analogous to Case 2, and by an argument similar to the last paragraph, one can show that Algorithm Trees uses $O(|V(G)|^2)$ time in this case as well. ◻

Now we are almost ready to prove Theorem 1.1, except that if we apply Theorem 2.8 directly to a 4-connected graph G to find a nonseparating chain decomposition of G, we spend $O(|V(G)|^2|E(G)|)$ time. We can obtain an $O(|V(G)|^3)$ algorithm by using the following result of Ibaraki and Nagamochi [10].

THEOREM 6.16. *Let G be a k-connected graph for some integer $k \geq 1$. Then one can find in $O(|V(G)| + |E(G)|)$ time a spanning k-connected subgraph of G with $O(|V(G)|)$ edges.*

Proof of Theorem 1.1. Let G be a 4-connected graph, and let $r \in V(G)$. Apply Theorem 6.16 to G, and let G' be the resulting spanning 4-connected subgraph of G.

Applying Theorem 2.8 to G', we can find a nonseparating chain decomposition $\mathcal{C}$ of G' in $O(|V(G')|^3)$ time (and hence in $O(|V(G)|^3)$ time).

Finally, apply Theorem 6.14 to $G, \mathcal{C}$ and find four independent spanning trees T_1, T_2, T_3, T_4 of G' rooted at r. By Lemma 6.15, this is done in $O(|V(G')|^3)$ time, and hence in $O(|V(G)|^3)$ time. Clearly, T_1, T_2, T_3, T_4 are independent spanning trees of G rooted at r. ◻

Acknowledgment. We thank an anonymous referee for correcting our mistakes and bringing references [6] and [14] to our attention.

REFERENCES

[1] J. Cheriyan and S. N. Maheshwari, *Finding nonseparating induced cycles and independent spanning trees in* 3*-connected graphs*, J. Algorithms, 9 (1988), pp. 507–537.

[2] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and S. Clifford, *Introduction to Algorithms*, 2nd ed., McGraw–Hill, Boston, MA, 2001.

[3] S. Curran, *Independent Trees in* 4*-Connected Graphs*, Ph.D. thesis, Georgia Institute of Technology, Atlanta, GA, 2003.

[4] S. Curran, O. Lee, and X. Yu, *Chain decomposition of* 4*-connected graphs*, SIAM J. Discrete Math., 19 (2005), pp. 848–880.

[5] D. Dolev, J. Y Halpern, B. Simons, and R. Strong, *A new look at fault tolerant network routing*, in Proceedings of the 16th Annual ACM Symposium on Theory of Computing, 1984, pp. 526–535.

[6] A. Frank, *Connectivity and network flows*, in Handbook of Combinatorics 1, R. L. Graham, M. Grötschel, and L. Lovász, eds., Elsevier, Amsterdam, 1995, pp. 111–177.

[7] J. E. Hopcroft and R. E. Tarjan, *Efficient planarity testing*, J. Assoc. Comput. Mach., 21 (1974), pp. 549–568.

[8] W.-L. Hsu and W.-K. Shih, *A new planarity test*, Theoret. Comput. Sci., 223 (1999), pp. 179–191.

[9] A. Huck, *Independent trees in graphs*, Graphs Combin., 10 (1994), pp. 29–45.

[10] T. Ibaraki and H. Nagamochi, *A linear-time algorithm for finding a sparse k-connected spanning subgraph of a k-connected graph*, Algorithmica, 7 (1992), pp. 583–596.

[11] A. Itai and M. Rodeh, *The multi-tree approach to reliability in distributed networks*, in Proceedings of the 25th Annual IEEE Symposium on the Foundations of Computer Science, 1984, pp. 137–147.

[12] A. Itai and A. Zehavi, *Three tree-paths*, J. Graph Theory, 13 (1989), pp. 175–188.

[13] K. Miura, S. Nakano, T. Nishizeki, and D. Takahashi, *A linear-time algorithm to find four independent spanning trees in four connected planar graphs*, Internat. J. Found. Comput. Sci., 10 (1999), pp. 195–210.

[14] A. Schrijver, *Problem* 32, in Combinatorial Optimization Volume C, Springer-Verlag, Berlin, 2003, pp. 1456–1457.