SIAM J. COMPUT.
Vol. 28, No. 4, pp. 1284–1297

LINEAR TIME ALGORITHMS FOR DOMINATING PAIRS IN ASTEROIDAL TRIPLE-FREE GRAPHS*

DEREK G. CORNEIL†, STEPHAN OLARIU‡, AND LORNA STEWART§

Abstract. An independent set of three vertices is called an *asteroidal triple* if between each pair in the triple there exists a path that avoids the neighborhood of the third. A graph is asteroidal triple-free (AT-free) if it contains no asteroidal triple. The motivation for this investigation is provided, in part, by the fact that AT-free graphs offer a common generalization of interval, permutation, trapezoid, and cocomparability graphs.

Previously, the authors have given an existential proof of the fact that every connected AT-free graph contains a dominating pair, that is, a pair of vertices such that every path joining them is a dominating set in the graph. The main contribution of this paper is a constructive proof of the existence of dominating pairs in connected AT-free graphs. The resulting simple algorithm, based on the well-known lexicographic breadth-first search, can be implemented to run in time linear in the size of the input, whereas the best algorithm previously known for this problem has complexity $O(|V|^3)$ for input graph $G = (V, E)$. In addition, we indicate how our algorithm can be extended to find, in time linear in the size of the input, all dominating pairs in a connected AT-free graph with diameter greater than 3. A remarkable feature of the extended algorithm is that, even though there may be $O(|V|^2)$ dominating pairs, the algorithm can compute and represent them in linear time.

Key words. algorithms, dominating pairs, asteroidal triple-free graphs, lexicographic breadth-first search

AMS subject classifications. 05C85, 68R10

PII. S0097539795282377

1. Introduction.

Considerable attention has been paid to exploiting algorithmically different aspects of the linear structure exhibited by various families of graphs. Examples of such families include interval graphs [15], permutation graphs [11], trapezoid graphs [6, 10], and cocomparability graphs [13].

The linearity of these four classes is usually described in terms of ad hoc properties of each of these classes of graphs. For example, in the case of interval graphs, the linearity property is traditionally expressed in terms of a linear order on the set of maximal cliques [4, 5]. For permutation graphs the linear behavior is explained in terms of the underlying partial order of dimension 2 [1]; for cocomparability graphs the linear behavior is expressed in terms of topological orderings of transitive orientations of comparability graphs [14], and so on.

As it turns out, the classes mentioned above are all subfamilies of a class of graphs called the asteroidal triple-free graphs (AT-free graphs). An independent set of three vertices is called an *asteroidal triple* if between any pair in the triple there exists a path that avoids the neighborhood of the third. AT-free graphs were introduced over

*Received by the editors March 3, 1995; accepted for publication (in revised form) June 20, 1997; published electronically March 22, 1999.

http://www.siam.org/journals/sicomp/28-4/28237.html

†Department of Computer Science, University of Toronto, Toronto, ON M5S 1A7, Canada (dgc@cs.utoronto.ca). The research of this author was supported by financial assistance from the Natural Sciences and Engineering Council of Canada.

‡Department of Computer Science, Old Dominion University, Norfolk, VA 23529-0162 (olariu@cs.odu.edu). The research of this author was supported in part by National Science Foundation grants CCR-9407180 and CCR-9522093 and by ONR grant N00014-97-1-0526.

§Department of Computing Science, University of Alberta, Edmonton, AB T6G 2H1, Canada (stewart@cs.ualberta.ca). The research of this author was supported by financial assistance from the Natural Sciences and Engineering Council of Canada.

three decades ago by Lekkerkerker and Boland [15], who showed that a graph is an interval graph if and only if it is chordal and AT-free. Thus, Lekkerkerker and Boland's result may be viewed as showing that the absence of asteroidal triples imposes the linear structure on chordal graphs that results in interval graphs. Recently, we have studied AT-free graphs with the stated goal of identifying the "agent" responsible for the linear behavior observed in the four subfamilies. Specifically, in [9] we presented evidence that the property of being AT-free is what is enforcing the linear behavior of these classes.

One strong "certificate" of linearity is the existence of a *dominating pair* of vertices, that is, a pair of vertices with the property that every path connecting them is a dominating set. In [9], we gave an existential proof of the fact that every connected AT-free graph contains a dominating pair.

The main contribution of this paper is a constructive proof of the existence of dominating pairs in connected AT-free graphs. A remarkable feature of our approach is that the resulting simple algorithm, based on the well-known lexicographic breadth-first search of [16], can easily be implemented to run in time $O(|V| + |E|)$, where the input is a connected AT-free graph $G = (V, E)$. In addition, our algorithm can be extended to find, in time linear in the size of the input, all dominating pairs in a connected AT-free graph with diameter greater than 3.

It should be noted that the fastest algorithm known to us, which recognizes whether or not a graph $G = (V, E)$ is AT-free, runs in time $O(|V|^3)$.

To put our result in perspective, we observe that previously, the most efficient algorithm for finding a dominating pair in a graph $G = (V, E)$ was the straightforward $O(|V|^3)$ algorithm described in [2].

For each of the four families mentioned above, vertices that occupy the extreme positions in the corresponding intersection model [12] constitute a dominating pair. It is interesting to note, however, that a linear time algorithm for finding a dominating pair was not previously known, even for cocomparability graphs, a strict subclass of AT-free graphs.

The remainder of this paper is organized as follows. Section 2 contains some relevant terminology and background. Section 3 is a description of the lexicographic breadth-first search algorithm of [16] along with some properties of that algorithm. In section 4 we present an algorithm which finds a dominating pair in a connected AT-free graph. In sections 5 and 6, we show how to extend the dominating pair algorithm to find all dominating pairs in a connected AT-free graph with sufficiently large diameter. Section 7 contains our conclusions.

2. Background. All the graphs in this work are finite with no loops or multiple edges. In addition to standard graph theoretic terminology compatible with [3], we shall define some new terms. We let $d(v)$ denote the *degree* of vertex v; $d(u, v)$ denotes the *distance* between vertices u and v in a graph, that is, the number of edges on a shortest path joining u and v. In addition, we let $\mathrm{diam}(G)$ denote the *diameter* of the graph G, that is, $\max_{u,v \in G} d(u, v)$. Two vertices u and v with $d(u, v) = \mathrm{diam}(G)$ are said to *achieve the diameter*. Given a graph $G = (V, E)$ and a vertex x, we let $N(x)$ denote the set of neighbors of x; $N'(x)$ denotes the set of neighbors of x in the complement $\overline{G}$ of G.

Let $\pi = v_1, v_2, \ldots, v_k$ be a path of graph G. If the subgraph of G induced by $\{v_1, v_2, \ldots, v_k\}$ has exactly $k - 1$ edges, i.e., none other than the edges of the path, then π is said to be an *induced*, or *chordless*, path. All the paths in this work are assumed to be induced unless stated otherwise. We refer to a path joining vertices x

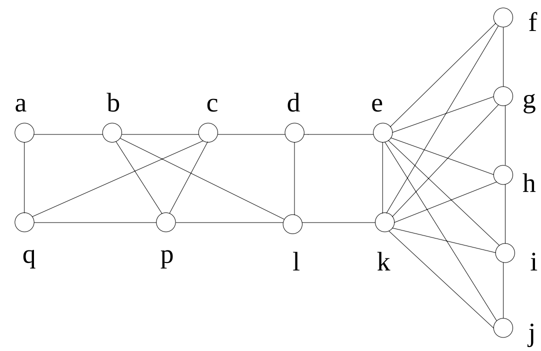

FIG. 1. *A connected AT-free graph G.*

and y as an x,y-path. We say that a vertex u *intercepts a path* π if u is adjacent to at least one vertex on π; otherwise, u is said to *miss* π. Let $G = (V, E)$ be a graph, π a path in G, x a vertex of G, and X a subset of V. Let $V(\pi)$ be the vertices of G that are on the path π. We shall use the following notation: $\pi - x$ refers to the subgraph of G induced by the vertices $V(\pi) - \{x\}$, $\pi + x$ refers to the subgraph of G induced by the vertices $V(\pi) \cup \{x\}$, and $\pi \cup X$ refers to the subgraph of G induced by the vertices $V(\pi) \cup X$.

For a connected AT-free graph with a pair of vertices x, y we let $D(x, y)$ denote the set of vertices that intercept all x,y-paths. Note that (x, y) is a dominating pair if and only if $D(x, y) = V$. We say that vertices u and v are *unrelated with respect to* x if $u \notin D(v, x)$ and $v \notin D(u, x)$. A vertex x of an AT-free graph G is called *pokable* if the graph obtained from G by adding a pendant vertex adjacent to x is AT-free. It is not hard to see that if an AT-free graph G contains no unrelated vertices with respect to x, then x is pokable. A *pokable dominating pair* is a dominating pair such that both vertices are pokable. A vertex x is a *pokable dominating pair vertex* if x is pokable and there exists y such that (x, y) is a dominating pair. To illustrate these definitions, consider the graph $G = (V, E)$ of Figure 1. In this graph, $D(c, l) = \{b, c, d, k, l, p, q\}$, $D(c, e) = V\backslash\{a\}$, and $D(a, e) = D(q, i) = V$. Any pair consisting of one vertex from $\{a, q\}$ and one vertex from $\{e, f, g, h, i, j, k\}$ is a dominating pair; a is pokable and h is not pokable (adding a pendant vertex h' adjacent to h would create the AT $\{f, j, h'\}$).

3. Lexicographic breadth-first search. Our dominating pair algorithm invokes Procedure LBFS (short for lexicographic breadth-first search), which, when given a connected graph G and a vertex x of G, returns a numbering of the vertices of G. We reproduce below the details of LBFS from [16].

PROCEDURE LBFS(G, x).
{Input: a connected graph $G = (V, E)$ and a distinguished vertex x of G;
Output: a numbering σ of the vertices of G}
begin
  label(x) $\leftarrow |V|$;
  for each vertex v in $V - \{x\}$ **do**
    label(v) $\leftarrow \Lambda$;
  for $i \leftarrow |V|$ **downto** 1 **do begin**
    pick an unnumbered vertex v with (lexicographically) the largest label;
    $\sigma(v) \leftarrow i$; {assign to v number i}
    for each unnumbered vertex u in $N(v)$ **do**

append i to label(u)
end
end; {LBFS}

Notice that the numbering returned by LBFS is not unique. One numbering that could result from LBFS(G, q), where G is the graph of Figure 1, is $\sigma(q) = 14$, $\sigma(p) = 13$, $\sigma(c) = 12$, $\sigma(a) = 11$, $\sigma(b) = 10$, $\sigma(l) = 9$, $\sigma(d) = 8$, $\sigma(k) = 7$, $\sigma(e) = 6$, $\sigma(i) = 5$, $\sigma(h) = 4$, $\sigma(j) = 3$, $\sigma(g) = 2$, $\sigma(f) = 1$.

A few definitions relating to LBFS are in order at this point. Let x be an arbitrary vertex of a connected graph G, and consider running LBFS(G, x). For vertices a, b of G we write $a \prec b$ whenever $\sigma(a) < \sigma(b)$, and we shall say that b is *larger* than a. To make the notation more manageable we shall sometimes write $v_1 \prec v_2 \prec \cdots \prec v_k$ as shorthand for $v_1 \prec v_2, v_2 \prec v_3, \ldots, v_{k-1} \prec v_k$. We shall denote by $\lhd$ the lexicographic total order of the set of LBFS labels. We let $\lambda(a, b)$ denote the label of a when b was about to be numbered. Given vertices a, b, c with $a \prec c$ and $b \prec c$, we shall say that a and b *are tied at* c if $\lambda(a, c) = \lambda(b, c)$. Given a vertex y, an a,b-path is said to be y-*majorizing* if all the vertices on the path are larger than y.

We assume that $G = (V, E)$ is an arbitrary connected graph and that LBFS(G, x) has been invoked, where x is an arbitrary vertex of G. The following fundamental properties of LBFS will be used later.

PROPOSITION 3.1. *Let a, b, and c be vertices of G satisfying $a \prec b$, $b \prec c$, $ac \in E$, and $bc \notin E$. Then there exists a vertex d in G adjacent to b but not to a and such that $c \prec d$.*

Proof. The existence of d follows immediately from the observation that when b was about to be processed by LBFS it could not have been tied with a. Since a inherited c's label, b must have inherited the label of a larger vertex nonadjacent to a; this is d. $\square$

PROPOSITION 3.2 (monotonicity property). *Let a, b, c, and d be vertices of G such that $a \prec c$ or $a = c$, $b \prec c$ or $b = c$, and $c \prec d$. If $\lambda(a, d) \lhd \lambda(b, d)$, then $\lambda(a, c) \lhd \lambda(b, c)$.*

Proof. The proof follows directly from the lexicographic ordering of labels. $\square$

LEMMA 3.3. *Let a, b, b', and c be vertices of G such that $a \prec b \prec c \prec b'$, $bb' \in E$, and $b'c \notin E$. Then a and c cannot be tied at b'.*

Proof. By Proposition 3.1 applied to vertices b, c, and b' we find a vertex c' adjacent to c but not to b and such that $b' \prec c'$. Write $C = \{t \mid tc \in E, tb \notin E, b' \prec t\}$. Clearly, $c' \in C$. In fact, we select c' to be the *largest* vertex in C.

If the statement is false, then a and c are tied at b'. Since $b' \prec c'$ and since $cc' \in E$, we must have $ac' \in E$. Now, Proposition 3.1 applied to vertices a, b, *and* c' yields a vertex b'' adjacent to b but not to a such that $c' \prec b''$.

Since $b' \prec b''$, the assumption that a and c are tied at b' guarantees that b'' is not adjacent to c. Therefore, Proposition 3.1 can be applied to vertices b, c, and b'', yielding a vertex c'' adjacent to c but not to b and such that $b'' \prec c''$. Since $b' \prec c' \prec b'' \prec c''$, it must be that $c'' \in C$, contradicting that c' is the largest vertex in C. $\square$

LEMMA 3.4. *Let y, a, and b be pairwise nonadjacent vertices of G such that $y \prec a$ and $a \prec b$. If a and y are not tied at b, then y misses a y-majorizing a,b-path.*

Proof. Assume that a and y are not tied at b. We must exhibit a y-majorizing a,b-path missed by y.

Since $y \prec a$, $\lambda(y, a) \lhd \lambda(a, a)$ or $\lambda(y, a) = \lambda(a, a)$. Therefore, by the monotonicity property (Proposition 3.2), $\lambda(y, b) \lhd \lambda(a, b)$ or $\lambda(y, b) = \lambda(a, b)$. Now, since a and y

are not tied at b, $\lambda(y,b) \lhd \lambda(a,b)$. Consequently, we find a vertex a_1 adjacent to a but not to y and such that $b \prec a_1$. (Vertex a_1 is chosen to be the largest satisfying these conditions.) We may assume that a_1 is not adjacent to b, since otherwise the path a, a_1, b is the desired y-majorizing path.

Now, Lemma 3.3 guarantees that b and y cannot be tied at a_1. Thus, we find a vertex b_1 adjacent to b but not to y and such that $a_1 \prec b_1$. (As before, we select as b_1 the largest vertex with this property.) Trivially, we may assume that b_1 is adjacent to neither a nor a_1; else we have the desired y-majorizing path. Again, Lemma 3.3 tells us that a_1 and y cannot be tied at b_1 and so we find a vertex a_2 adjacent to a_1 but not to y and such that $b_1 \prec a_2$. (As before, we select as a_2 the largest vertex with this property.) It is easy to verify that a_2 is not adjacent to a (by the choice of a_1), b, or b_1.

Continuing as above, we obtain two chordless y-majorizing paths $a = a_0, a_1, a_2, \ldots$ and $b = b_0, b_1, b_2, \ldots$, both missed by y. If no vertex on the first path is adjacent to a vertex on the second one, then the paths are infinite, contradicting that G is finite. Therefore, such an adjacency must exist, yielding the desired a,b-path. ∎

4. The dominating pair algorithm. Our dominating pair algorithm takes as input a connected AT-free graph G and returns a pokable dominating pair of G. The algorithm provides a constructive proof of the existence of pokable dominating pairs in connected AT-free graphs. (An existential proof of this fact was given in [9].)

The four properties of LBFS specified in the preceding section hold for every connected graph G. The proof of correctness of the dominating pair algorithm relies on two additional properties of LBFS which hold when the input graph is a connected AT-free graph. We present these properties next.

THEOREM 4.1. *Let $G = (V, E)$ be a connected AT-free graph and let x and y be arbitrary vertices of G. Let $\prec$ be the vertex ordering corresponding to a numbering produced by LBFS(G, x). The subgraph of G induced by y and all vertices z with $y \prec z$ contains no unrelated vertices with respect to y.*

Proof. First, we give an overview of the proof. The existence of such unrelated vertices, u and v, and the fact that they are numbered before y in an LBFS from x, would imply that u and v are connected by a path through x. If y misses such a path, then $\{y, u, v\}$ is an AT.

In particular, if the statement is false, we find a vertex y and vertices u, v with $y \prec u \prec v$, such that u and v are unrelated with respect to y. This implies the existence of chordless paths $\pi(y,u) : y = u_1, u_2, \ldots, u_p = u$ missed by v, and $\pi(y,v) : y = v_1, v_2, \ldots, v_q = v$ missed by u, with the vertices on both paths, except for y, numbered by LBFS before y. We claim that

$$u \prec v_3. \tag{4.1}$$

If (4.1) is false, then $v_3 \prec u$ and $u \prec v$, and we must find a subscript i ($3 \leq i \leq q-1$) such that $v_i \prec u$ and $u \prec v_{i+1}$. Now, Lemma 3.3 tells us that u and y cannot be tied at v_{i+1}. In turn, Lemma 3.4 guarantees the existence of a y-majorizing u,v_{i+1}-path missed by y. This path extends trivially to a y-majorizing u,v-path, implying that $\{y, u, v\}$ is an AT. Thus, (4.1) must hold.

Next, we claim that

$$u \text{ and } y \text{ are tied at } v_3. \tag{4.2}$$

The contrary would imply, by virtue of Lemma 3.4, the existence of a y-majorizing

u,v_3-path missed by y. This extends easily into a y-majorizing u,v-path missed by y, implying that $\{y, u, v\}$ is an AT. Thus, (4.2) must hold.

We note that $v_2 \prec v_3$; otherwise, since v_2 is adjacent to y and not to u, we would contradict (4.2). Further, we claim that

$$u \prec v_2. \tag{4.3}$$

Otherwise, by (4.1) we have $v_2 \prec u$ and $u \prec v_3$. Now, Lemma 3.3 specifies that u and y cannot be tied at v_3, contradicting (4.2). Thus, (4.3) must be true.

Proposition 3.1 applied to vertices $y \prec u$ and $u \prec v_2$ guarantees the existence of a vertex u' adjacent to u but not to y and such that $v_2 \prec u'$. Since u and y are tied at v_3, it must be the case that $y \prec u$, $u \prec v_2$, $v_2 \prec u'$, and $u' \prec v_3$. If u' is adjacent to v_3, then we have a u,v-path missed by y, contradicting that the graph is AT-free. Thus, u' is not adjacent to v_3. But now, Lemma 3.3 guarantees that y and u' cannot be tied at v_3. Further, Lemma 3.4 tells us that there must exist a y-majorizing u',v_3-path missed by y. This path extends in the obvious way to a y-majorizing u,v-path missed by y, contradicting that the graph is AT-free. This completes the proof of Theorem 4.1. ∎

We observe that, if G contains no unrelated vertices with respect to vertex v, then v is pokable. This observation and Theorem 4.1 combined imply that each vertex y of G is pokable in the subgraph of G induced by y and all vertices z with $y \prec z$. In particular, the last vertex numbered by LBFS(G, x) is pokable in G.

One additional theorem about LBFS, specialized to connected AT-free graphs, will lead to the dominating pair algorithm.

THEOREM 4.2. *Let $G = (V, E)$ be a connected AT-free graph and suppose that G contains no vertices unrelated with respect to vertex x of G. Let $\prec$ be a vertex ordering corresponding to a numbering produced by LBFS(G, x). Then, for all vertices u, v in V with $u \prec v$, $v \in D(u, x)$.*

Proof. The argument proceeds by noting that if $v \notin D(u, x)$, then there is a u,x-path missed by v and no v,x-path missed by u (since u and v cannot be unrelated with respect to x). However, an LBFS from x would number u before v, contradicting the conditions of the theorem.

Assume that the theorem is false and let v be the largest vertex in V for which there exists a vertex u with $u \prec v$ and $v \notin D(u, x)$. We now select a specific path π and a vertex u with $u \prec v$ such that π is a u,x-path missed by v. Let U be the set of all vertices u such that $u \prec v$ and $v \notin D(u, x)$ and let $\mathcal{P}$ be the set of all chordless u,x-paths in G that are missed by v. Among all minimum length paths in $\mathcal{P}$, we choose π to be the one that extends to the largest possible vertex at each step. Now u is the endpoint of π that is in the set U.

Formally, let $\mathcal{P}_\mathcal{M}$ be the subset of $\mathcal{P}$ consisting of all minimum length paths of $\mathcal{P}$. For paths $P = p_1, p_2, \ldots, p_k$ and $P' = p'_1, p'_2, \ldots, p'_k$ in $\mathcal{P}_\mathcal{M}$, we say that P' is greater than P if there exists a subscript i, $1 \le i \le k$, such that $\sigma(p'_j) = \sigma(p_j)$ for all $1 \le j < i$ and $\sigma(p'_i) > \sigma(p_i)$. Clearly, "greater than" is a total order on $\mathcal{P}_\mathcal{M}$. We choose $\pi : u = u_1, u_2, \ldots, u_k = x$ to be the unique greatest element of $\mathcal{P}_\mathcal{M}$.

Observe that $u_1 \prec v$ and $v \prec u_2$; otherwise we contradict the fact that π is in $\mathcal{P}_\mathcal{M}$. Now Proposition 3.1 guarantees the existence of a vertex v_2 adjacent to $v = v_1$ but not to u_1 and such that $u_2 \prec v_2$.

It is easily seen that v_2 is nonadjacent to u_i for all $i > 2$, since otherwise u and v are unrelated with respect to x. This immediately implies that $v_2 \prec u_3$ (otherwise we contradict the choice of v) and $u_2v_2 \in E$ (otherwise we contradict the choice of both u and v).

Now apply Proposition 3.1 to vertices u_2, v_2, and u_3; we find a vertex v_3 adjacent to v_2 but not to u_2 and such that $u_3 \prec v_3$.

Since $u_2 \prec v_3$, v_3 cannot miss the path $u_2, u_3, \ldots, u_k = x$. Let t ($t \geq 3$) be the largest subscript for which $v_3 u_t \in E$. Now v_3 must be adjacent to u_1; else u_1 and v_1 are unrelated with respect to x. (The v_1,x-path missed by u_1 is $v_1, v_2, v_3, u_t, \ldots, x$.) Note also that v_3 must be adjacent to v_1; otherwise we contradict the choice of π. (To see this, note that u_1, v_3 extends to a minimum length chordless u,x-path via u_t.) Now, $t = 3$; else the assignment $v \leftarrow u_2$ and $u \leftarrow v$ contradicts the initial choice of u and v.

Now assume that we have constructed a sequence $v = v_1, v_2, \ldots, v_i$ of vertices such that v_i ($i \geq 3$) satisfies the following conditions:

(a) $v_i v_{i-1}, v_i u_i \in E$;

(b) $u_i \prec v_i$ and $v_i u_{i-1} \notin E$;

(c) $v_i u_j \notin E$ for $j > i$;

(d) $v_i u_1, v_i v_1 \in E$.

We argue that there exists a vertex v_{i+1} satisfying conditions (a)–(d) with $i+1$ in place of i. For this purpose, note that u_{i+1} exists since $u_i \prec v_i$ implies that $u_i \neq x$. Now, $v_i \prec u_{i+1}$ since otherwise the assignment $v \leftarrow v_i$ and $u \leftarrow u_{i+1}$ contradicts the initial choice of u and v.

Now, by (c), Proposition 3.1 applied to vertices u_i, v_i, and u_{i+1} guarantees the existence of a vertex v_{i+1} adjacent to v_i but not to u_i and such that $u_{i+1} \prec v_{i+1}$. Thus, (b) is verified. Let t be the largest subscript for which v_{i+1} is adjacent to u_t. (t exists and $t \geq i+1$, since otherwise the assignment $v \leftarrow v_{i+1}$ and $u \leftarrow u_i$ contradicts the initial choice of u and v.)

Note that u_{i-1} is adjacent to v_{i+1}, since if it is not, then u_{i-1} and v_1 are unrelated with respect to x. (By (b) and (d), the path contained in $v_1, v_i, v_{i+1}, u_t, u_{t+1}, \ldots, x$ is missed by u_{i-1}.) Also v_1 is adjacent to v_{i+1}, since otherwise we contradict the choice of π by going down π and picking the first edge $u_j v_{i+1}$, which we know exists (in particular, we know that $u_{i-1} v_{i+1} \in E$) and then going to u_t and on to x. Now u_1 is adjacent to v_{i+1}, since otherwise u_1 and v_1 are unrelated with respect to x (the path $v_1, v_{i+1}, u_t, \ldots, x$ would be missed by u_1). Thus, (d) holds.

Note that $t = i+1$; otherwise we should have picked u_i instead of v (for u_i misses the path u_1, v_{i+1}, u_t, etc.). Thus, both (a) and (c) hold.

But now we have reached a contradiction: v_k must exist and it must be that $x = u_k \prec v_k$, which is absurd. ∎

Theorem 4.2 implies that if G contains no vertices unrelated with respect to x, then (x, y) is a dominating pair in the subgraph of G induced by y and all vertices z with $y \prec z$. In particular, x and the last vertex numbered by LBFS(G, x) constitute a dominating pair of G.

We are now in a position to spell out the details of the dominating pair algorithm.

PROCEDURE DP(G).

{Input: a connected AT-free graph G;

Output: (y, z) a pokable dominating pair of G}

begin

choose an arbitrary vertex x of G;

if $N'(x) = \emptyset$ **then** return (x, x);

LBFS(G, x);

let y be the vertex numbered last by LBFS(G, x);

LBFS(G, y);

let z be the vertex numbered last by LBFS(G, y);
return(y, z)
end; {DP}

As an example, we refer again to the graph G of Figure 1. We saw earlier that a possible numbering resulting from LBFS(G, q) corresponds to the ordering

$$f \prec g \prec j \prec h \prec i \prec e \prec k \prec d \prec l \prec b \prec a \prec c \prec p \prec q.$$

LBFS(G, f) may produce the ordering

$$a \prec q \prec b \prec p \prec c \prec l \prec d \prec j \prec i \prec h \prec g \prec k \prec e \prec f.$$

Thus, DP(G) may output the pokable dominating pair (a, f).

Finally, we state the following result.

THEOREM 4.3. *Procedure DP finds a pokable dominating pair in a connected AT-free graph, $G = (V, E)$, in $O(|V| + |E|)$ time.*

Proof. Clearly, (x, x) is a pokable dominating pair of G if $N'(x) = \emptyset$. Otherwise, by Theorem 4.1, G contains no unrelated vertices with respect to y and, hence, by Theorem 4.2, (y, z) is a dominating pair of G. In addition, Theorem 4.1 implies that both y and z are pokable in G. It is clear that a linear time implementation is possible (see [16] for details of a linear time implementation of LBFS). □

5. Computing dominated sets. Since dominating pairs play an important role in the study of AT-free graphs and, intuitively, correspond to the extreme endpoints of the linear structure of the graph, it is interesting to ask whether the above algorithm can be the basis of an efficient algorithm to find all of the dominating pairs in a connected AT-free graph. It turns out that we can indeed extend the algorithm to efficiently find all dominating pairs in a connected AT-free graph provided that the graph has diameter greater than 3. The diameter restriction is a consequence of Theorem 6.1, which states that the set of dominating pairs is precisely the Cartesian product of two subsets of vertices X and Y, provided the diameter of the graph is greater than 3. Thus, by computing X and Y, we have a linear-sized implicit representation of all dominating pairs. Such representations do not seem to hold for AT-free graphs with diameter less than 4.

Perhaps even more interesting in its own right, and a step in the direction of computing all dominating pairs, is a method that, given a connected AT-free graph G and a pokable dominating pair vertex x of G, computes the sets $D(v, x)$ for all vertices v of G. (Recall that $D(v, x)$ denotes the set of vertices that intercept all v,x-paths.) We describe this method first and, in the next section, we show how the information obtained can be used to compute all the dominating pairs in a connected AT-free graph with diameter greater than 3.

In order to understand our approach, which relies on a variant of LBFS, let us examine a few details of an efficient LBFS implementation. We use an adjacency list representation of a graph. Additionally, unnumbered vertices are stored in another data structure, specifically, a list of lists. At each stage of the algorithm, each list contains unnumbered vertices having the same label (i.e., vertices that are tied at the current stage), and lists are stored in decreasing lexicographic order of the corresponding labels. Thus, the largest label can be found in constant time. Let us examine the evolution of the list of lists during the execution of LBFS(G, x) where $G = (V, E)$. Initially, there are two lists: one contains the vertex x and corresponds to the label $|V|$, and the other contains all other vertices of G and corresponds to the label Λ.

Each time a new vertex u is numbered, it is removed from its list and its number is appended to the labels of its unnumbered neighbors. Each list that contains both an unnumbered neighbor of u and a vertex that is not adjacent to u, is split into two lists, one for the original label and one corresponding to the original label with $\sigma(u)$ appended. The first list follows the second in the ordered list of lists. It is important to note that, by the monotonicity property (Proposition 3.2), the relative order of the lists never changes. In order to access and move the neighbors of u in $O(d(u))$ time, an array of $|V|$ pointers indicates the location of each unnumbered vertex within the list of lists, and the lists are doubly linked.

We now return to the problem at hand, namely, given a connected AT-free graph $G = (V, E)$ and a pokable dominating pair vertex x of G, we wish to compute the sets $D(v, x)$ for all vertices v of G. We will modify LBFS to obtain a linear time algorithm for this problem. To begin, we observe that the sum of the cardinalities of the sets $D(v, x)$, for all $v \in V$, may be $O(|V|^2)$, and hence, a linear time algorithm must use an implicit representation of these sets. We handle this as follows: for each vertex v we compute a number, $\mathrm{span}(v)$, $1 \leq \mathrm{span}(v) \leq \sigma(v)$, with the property that $D(v, x) = \{u|\sigma(u) \geq \mathrm{span}(v)\} \cup N^-(v)$, where σ is a numbering resulting from LBFS(G, x) and $N^-(v) = N(v) \cap \{w|\sigma(w) < \sigma(v)\}$. Thus, $\mathrm{span}(v)$ indicates an interval, with respect to σ, of vertices to be included in $D(v, x)$. When all vertices of a set $W \subseteq V$ have the same span value, we refer to that value as $\mathrm{span}(W)$.

The values of $\mathrm{span}(v)$ for all vertices v are computed incrementally. It is not necessary to update the values individually because all vertices on the same list will have the same span value. Thus, we store span values for each list, rather than for each vertex. The two initial lists have span values of $|V|$. Just before a vertex is numbered, the span value of its list is updated. When a list is split, the two new lists inherit the span value of the original list. A span value is assigned to an individual vertex when that vertex is finally numbered. As we maintain the lists, we store the size of each list. Thus, for the list W in each iteration, W, $|W|$, and $\mathrm{span}(W)$ can be accessed in constant time. Furthermore, over all iterations, all updates of span values can be accomplished in linear time. Thus, the overall complexity of the algorithm below is $O(|V| + |E|)$.

Procedure DSETS is a modified LBFS which computes implicit representations of $D(v, x)$ for all $v \in V$.

PROCEDURE DSETS(G, x).

{Input: a connected AT-free graph $G = (V, E)$ and a pokable dominating pair vertex x of G;

Output: a numbering σ of the vertices of G and, for each vertex v, $\mathrm{span}(v)$ such that $D(v, x) = \{u|\sigma(u) \geq \mathrm{span}(v)\} \cup N^-(v)$ }

begin

  label$(x) \leftarrow |V|$;

  for each vertex v in $V - \{x\}$ **do**

    label$(v) \leftarrow \Lambda$;

  $W_1 \leftarrow \{x\}$; $W_2 \leftarrow V - \{x\}$; {Initialize two lists}

  $\mathrm{span}(W_1) \leftarrow \mathrm{span}(W_2) \leftarrow |V|$;

  for $i \leftarrow |V|$ **downto** 1 **do begin** {main for loop}

    pick an unnumbered vertex v with (lexicographically) the largest label;

    let W be the list containing v and all vertices tied with v;

    $\mathrm{span}(W) \leftarrow \min\ \{\mathrm{span}(W),\ i + 1 - |W|\ \}$;

    remove v from W;

$\sigma(v) \leftarrow i$; {assign to v number i}
span(v) $\leftarrow$ span(W);
for each unnumbered vertex u in $N(v)$ **do**
  append i to label(u);
split lists as necessary so that there is a one-to-one correspondence between the resulting set of lists and the vertex labels
end {main for loop}
end; {DSETS}

Let us look again at the graph G of Figure 1. Suppose that the numbering returned by DSETS(G, q) corresponds to the ordering

$$f \prec g \prec j \prec h \prec i \prec e \prec k \prec d \prec l \prec b \prec a \prec c \prec p \prec q.$$

Now the span values computed by DSETS(G, q) will be

v	a	b	c	d	e	f	g	h	i	j	k	l	p	q
span(v)	11	10	11	8	6	1	1	1	1	1	7	9	11	14

It is easy to verify that the corresponding sets are correctly represented in this case. For example, $D(q, q) = \{a, c, p, q\}$, $D(l, q) = \{a, b, c, d, k, l, p, q\}$, and $D(e, q) = V$.

Before presenting the proof of correctness of Procedure DSETS, we examine the relationship between vertices x (such that there are no vertices unrelated with respect to x) and pokable dominating pair vertices. The following lemma acts as a bridge between the results of section 4 and the subsequent results of this section.

LEMMA 5.1. *Let G be a connected AT-free graph and let x be an arbitrary vertex of G. Then G contains no vertices unrelated with respect to x if and only if x is a pokable dominating pair vertex of G.*

Proof. The "only if" part follows from Theorem 4.2 and the fact that if G contains no vertices unrelated with respect to x, then x is pokable (since no AT can be created by adding a pendant vertex adjacent to x). To prove the "if" part, let y be a vertex of G such that (x, y) is a dominating pair of G, and consider unrelated vertices u and v with respect to x. Since (x, y) is a dominating pair, u and v intercept every path joining x and y. Let π be an x,y-path and let u' and v' be vertices on π adjacent to u and v, respectively. Trivially, both u' and v' are distinct from x. But now there exists a u,v-path in G that does not contain x (this path contains vertices u', v' and a subpath of π), implying that x is not pokable. ∎

The correctness of Procedure DSETS relies on the following theorem.

THEOREM 5.2. *Let x be a pokable dominating pair vertex of a connected AT-free graph $G = (V, E)$. For every vertex v of G, $D(v, x) = \{u|\sigma(u) \geq \text{span}(v)\} \cup N^-(v)$.*

Proof. Informally, notice that span(v) is the smallest numbered vertex that is tied with v at any point in the algorithm. Intuitively, all vertices in $\{u|\sigma(u) \geq \text{span}(v)\}$, as well as all neighbors of v, are in $D(v, x)$. The proof demonstrates that $D(v, x)$ is exactly equal to this set of vertices.

Formally, let $\sigma : V \mapsto \{1, 2, \ldots, n\}$ be a numbering returned by DSETS(G, x), let v be an arbitrary vertex of V, and let $D(v) = \{u|\sigma(u) \geq \text{span}(v)\} \cup N^-(v)$.

Our plan is to prove that $D(v) = D(v, x)$. To implement this plan we first prove that $D(v) \subseteq D(v, x)$. Suppose that $D(v) \not\subseteq D(v, x)$, and let u be a vertex in $D(v)$ but not in $D(v, x)$. Now $uv \notin E$ and, by Theorem 4.2 and Lemma 5.1, $\sigma(u) < \sigma(v)$. Thus, by the algorithm, span(v) $\leq \sigma(u) < \sigma(v)$. (To see this, notice that during the iteration of the main for loop in which vertex v is numbered, the list W that contains v receives a span value less than or equal to $i+1-|W|$. Since W contains v, $|W| \geq 1$ and hence $\mathrm{span}(W) \leq i$. Now, the desired inequality follows, since $\sigma(v)$ is assigned the value i and $\mathrm{span}(v)$ is assigned the value $\mathrm{span}(W)$.)

Let w be the vertex being processed when $\mathrm{span}(v)$ was first set to its final value. (It could be that $w = v$.) Then u, v, and w were tied at w; that is, $N(u) \cap \{t|\sigma(t) > \sigma(w)\} = N(v) \cap \{t|\sigma(t) > \sigma(w)\} = N(w) \cap \{t|\sigma(t) > \sigma(w)\}$. Since $u \notin D(v,x)$ there exists a v,x-path $\pi : v = v_1, v_2, \ldots, v_k = x$ missed by u. By Theorem 4.2 and Lemma 5.1, all vertices of π are larger than u. Let i be the greatest index such that $\sigma(v_i) < \sigma(w)$. Clearly, $i < k$, since $\sigma(w) < \sigma(x)$. Now $\sigma(u) < \sigma(v_i) < \sigma(w)$ and thus, by the monotonicity property (Proposition 3.2), v_i was tied with u, v, and w at w. But v_i is adjacent to v_{i+1} with $\sigma(w) < \sigma(v_{i+1})$; thus, v_{i+1} belongs to $N(w) \cap \{t|\sigma(t) > \sigma(w)\}$ and is therefore adjacent to u, contradicting that π is missed by u. Thus, $D(v) \subseteq D(v,x)$.

We now prove that $D(v,x) \subseteq D(v)$, thereby completing the proof of the theorem. Suppose that $D(v,x) \not\subseteq D(v)$. Let u be a vertex in $D(v,x)$ but not in $D(v)$. Clearly, $uv \notin E$ and $\sigma(u) < \mathrm{span}(v) \leq \sigma(v)$.

If at any stage u and v are tied with the vertex being processed, then $\mathrm{span}(v)$ is set to a value less than or equal to $\sigma(u)$, and $\mathrm{span}(v)$ is never increased. Thus, since $\mathrm{span}(v) > \sigma(u)$, we know that u and v are never tied with the vertex being processed. Let z be the largest neighbor of v. Since u cannot be adjacent to any vertex greater than z (else $\sigma(u) < \sigma(v)$ is contradicted), and since there is a z,x-path consisting entirely of z and vertices larger than z (by the breadth-first nature of the search), u must be adjacent to z. (Otherwise we contradict the fact that $u \in D(v,x)$.) Now let Z be the vertices of $N(v) \cap \{t|\sigma(t) > \sigma(v)\}$ which have a neighbor greater than z, and let Z' be the remaining vertices of $N(v) \cap \{t|\sigma(t) > \sigma(v)\}$. Clearly, $z \in Z$. Note that u is adjacent to all vertices of Z; otherwise there is a v,x-path missed by u, which is a contradiction. We observe that for every $z \in Z$ and every $z' \in Z'$, it must be that $z' \prec z$. This follows from the monotonicity property (Proposition 3.2) and by the fact that, at z, all vertices of Z' have labels lexicographically less than all vertices of Z. Finally, all vertices of Z' are adjacent to all vertices of Z, since any vertex of Z' nonadjacent to a vertex of Z would have been processed after v. But when the first vertex of Z' is processed, all vertices of Z', u, and v are tied, contradicting our earlier statement that u and v are not tied at any stage. This completes the proof. $\square$

Theorem 5.2, along with the discussion preceding Procedure DSETS, implies the following result.

THEOREM 5.3. *Let $G = (V,E)$ be a connected AT-free graph and let x be a pokable dominating pair vertex of G. Procedure DSETS computes implicit representations for the sets $D(v,x)$ for every vertex v of G in $O(|V| + |E|)$ time.*

6. Computing all dominating pairs. We now describe how to use the span values computed by Procedure DSETS to compute all dominating pairs in a connected AT-free graph with sufficiently large diameter. Our algorithm relies on the following result, the proof of which appears in [9].

THEOREM 6.1 (see [9]). *Let G be a connected AT-free graph with diam(G) > 3. There exist nonempty, disjoint sets X and Y of vertices of G such that (x,y) is a dominating pair if and only if $x \in X$ and $y \in Y$.*

We note that Theorem 6.1 is best possible in the sense that, for AT-free graphs of diameter less than 4, the sets X and Y are not guaranteed to exist. To wit, C_5 and the graph of Figure 2 provide counterexamples of diameters 2 and 3, respectively.

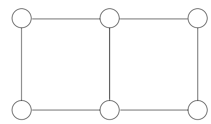

FIG. 2. *An AT-free graph of diameter* 3 *for which the sets* X *and* Y *do not exist.*

Procedure ALL-DPs takes as input a connected AT-free graph $G = (V, E)$ with diameter greater than 3, and returns X and Y, subsets of V such that (x, y) is a dominating pair if and only if $x \in X$ and $y \in Y$. Procedure DSETS is an integral part of Procedure ALL-DPs.

We begin with an informal description of Procedure ALL-DPs. The first step is to find a pokable dominating pair vertex, which is done by LBFS in linear time. Then, Procedure DSETS(G, x) computes span(v) for all vertices v in linear time. From the resulting span values, and by Theorem 6.1, it is easy to see how to proceed.

Now, Y is the set of all vertices y with $D(y, x) = V$ (whether or not $D(y, x) = V$ can be computed in $O(d(y))$ time by scanning the adjacency list of y and checking whether all vertices w with $\sigma(w) < \text{span}(y)$ are adjacent to v). Finally, we call DSETS(G, y), where y is the vertex that was numbered last by DSETS(G, x). The new set of span values can be used to compute the set X in a manner identical to the above method for computing Y. We now state the procedure more precisely.

PROCEDURE ALL-DPs(G).
{Input: connected AT-free graph $G = (V, E)$ with diam$(G) > 3$;
Output: $X \subseteq V$ and $Y \subseteq V$ such that (x, y) is a dominating pair of G if and only if $x \in X$ and $y \in Y$}
begin
  choose an arbitrary vertex w of G;
  LBFS(G, w);
  let x be the vertex numbered last by LBFS(G, w);
  DSETS(G, x);
  $Y = \emptyset$;
  for every $y \in V$ **do begin**
    count $\leftarrow |V| - \text{span}(y)$; {number of vertices not in $\{u | \sigma(u) \geq \text{span}(y)\}$}
    for each $u \in N(y)$ **do**
      if $\sigma(u) < \text{span}(y)$ **then** *count* $\leftarrow$ *count* $- 1$;
    if *count* $= 0$ **then** $Y \leftarrow Y \cup \{y\}$
    end;
  let y be the vertex numbered last by LBFS(G, x);
  DSETS(G, y);
  $X = \emptyset$;
  for every $x \in V$ **do begin**
    count $\leftarrow |V| - \text{span}(x)$; {number of vertices not in $\{u | \sigma(u) \geq \text{span}(x)\}$}
    for each $u \in N(x)$ **do**
      if $\sigma(u) < \text{span}(x)$ **then** *count* $\leftarrow$ *count* $- 1$;
    if *count* $= 0$ **then** $X \leftarrow X \cup \{x\}$
    end;
  return(X, Y)
  end; {ALL-DPs}

As an illustration, when run with the graph G of Figure 1 as input, Procedure ALL-DPs returns $X = \{a, q\}$, $Y = \{e, f, g, h, i, j, k\}$ or $X = \{e, f, g, h, i, j, k\}$, $Y = \{a, q\}$ (depending upon the initial choice of w).

THEOREM 6.2. *For a connected AT-free graph $G = (V, E)$ with diam(G) > 3, Procedure ALL-DPs computes sets X and Y such that (x, y) is a dominating pair of G if and only if $x \in X$ and $y \in Y$ in $O(|V| + |E|)$ time.*

Proof. We observe that, by Theorem 4.1 and Lemma 5.1, the vertex x in ALL-DPs is guaranteed to be a pokable dominating pair vertex. Similarly, by Theorem 4.1 and Lemma 5.1, the vertex y is a pokable dominating pair vertex. (This follows from the observation that the set of possible numberings produced by DSETS(G, x) is exactly the set of possible numberings of LBFS(G, x), since DSETS is simply LBFS with some additional computations.) Thus, the correctness of Procedure ALL-DPs follows from Theorem 5.3 and Theorem 6.1. Similarly, the complexity of ALL-DPs is the sum of the complexities of LBFS and DSETS plus an $O(|V| + |E|)$ term. $\square$

Let $G = (V, E)$ be a connected AT-free graph with diam$(G) > 3$. Notice that, even though there may be $O(|V|^2)$ dominating pairs in G, Procedure ALL-DPs can compute and represent them in linear time, by virtue of Theorem 6.1. A similar comment applies to the sets $D(v, x)$ for all $v \in V$; even though the sum of the cardinalities of the sets may be $O(|V|^2)$, Procedure DSETS can compute an implicit representation of them in linear time.

We conclude with a corollary, which follows from the fact that some minimum cardinality connected dominating set must be a shortest path between the vertices of a dominating pair (proved in [9]). Once X and Y have been found, a minimum distance dominating pair can be found in linear time by performing a breadth-first search starting at X until a vertex of Y is encountered. In [7], we presented a linear time algorithm to compute a dominating path in an arbitrary connected AT-free graph, but that algorithm does not guarantee a minimum cardinality dominating path. The method of the present paper does guarantee a minimum cardinality dominating path for connected AT-free graphs with diameter greater than 3.

COROLLARY 6.3. *Let $G = (V, E)$ be a connected AT-free graph with diameter greater than 3. A minimum cardinality connected dominating set of G can be computed in $O(|V| + |E|)$ time.* $\square$

7. Conclusions.

We have presented a linear time algorithm, based on the well-known lexicographic breadth-first search of [16], for finding a pokable dominating pair in a connected AT-free graph, $G = (V, E)$. The algorithm provides a constructive proof of the existence of pokable dominating pairs in connected AT-free graphs (an existential proof of this fact was given in [9]). It is an improvement over the previously known $O(|V|^3)$ algorithm of [2]. In addition, we extended the dominating pair algorithm to find all dominating pairs in a connected AT-free graph, $G = (V, E)$, with diameter greater than 3. Even though there may be $O(|V|^2)$ dominating pairs, the extended algorithm can compute and implicitly represent them in $O(|V| + |E|)$ time. We remark that the simpler maximum cardinality search (MCS) of Tarjan and Yannakakis [17] cannot take the place of LBFS in these algorithms.

In [8], we presented a different linear time algorithm for finding a dominating pair in a connected AT-free graph. This other algorithm is based on a recursive use of a maximum cardinality breadth-first search. The method does not seem to allow linear time calculation of $D(v, x)$ for all vertices v where x is a pokable dominating pair vertex, or of sets X and Y when the diameter of the graph is greater than 3.

REFERENCES

[1] K. A. Baker, P. C. Fishburn, and F. S. Roberts, *Partial orders of dimension two*, Networks, 2 (1971), pp. 11–28.

[2] H. Balakrishnan, A. Rajaraman, and C. Pandu Rangan, *Connected domination and Steiner set on asteroidal triple-free graphs*, in Workshop on Algorithms and Data Structures, WADS '93, F. Dehne, J.-R. Sack, N. Santoro, and S. Whitesides, eds., Lecture Notes in Comput. Sci., 709, Springer-Verlag, New York, 1993, pp. 131–141.

[3] J. A. Bondy and U. S. R. Murty, *Graph Theory with Applications*, North-Holland, Amsterdam, 1976.

[4] K. S. Booth and G. S. Lueker, *Testing for the consecutive ones property, interval graphs and graph planarity using PQ-tree algorithms*, J. Comput. System Sci., 13 (1976), pp. 335–379.

[5] K. S. Booth and G. S. Lueker, *A linear time algorithm for deciding interval graph isomorphism*, J. ACM, 26 (1979), pp. 183–195.

[6] D. G. Corneil and P. A. Kamula, *Extensions of permutation and interval graphs*, Congr. Numer., 58 (1987), pp. 267–276.

[7] D. G. Corneil, S. Olariu, and L. Stewart, *A linear time algorithm to compute a dominating path in an AT-free graph*, Inform. Process. Lett., 54 (1995), pp. 253–257.

[8] D. G. Corneil, S. Olariu, and L. Stewart, *Computing a dominating pair in an asteroidal triple-free graph in linear time*, in Workshop on Algorithms and Data Structures, WADS '95, S. G. Akl, F. Dehne, J.-R. Sack, and N. Santoro, eds., Lecture Notes in Comput. Sci., 955, Springer-Verlag, New York, 1995, pp. 358–368.

[9] D. G. Corneil, S. Olariu, and L. Stewart, *Asteroidal triple-free graphs*, SIAM J. Discrete Math., 10 (1997), pp. 399–430.

[10] I. Dagan, M. C. Golumbic, and R. Y. Pinter, *Trapezoid graphs and their coloring*, Discrete Appl. Math., 21 (1988), pp. 35–46.

[11] S. Even, A. Pnueli, and A. Lempel, *Permutation graphs and transitive graphs*, J. ACM, 19 (1972), pp. 400–410.

[12] M. C. Golumbic, *Algorithmic Graph Theory and Perfect Graphs*, Academic Press, New York, 1980.

[13] M. C. Golumbic, C. L. Monma, and W. T. Trotter, Jr., *Tolerance graphs*, Discrete Appl. Math., 9 (1984), pp. 157–170.

[14] D. Kratsch and L. Stewart, *Domination on cocomparability graphs*, SIAM J. Discrete Math., 6 (1993), pp. 400–417.

[15] C. G. Lekkerkerker and J. C. Boland, *Representation of a finite graph by a set of intervals on the real line*, Fund. Math., 51 (1962), pp. 45–64.

[16] D. J. Rose, R. E. Tarjan, and G. S. Lueker, *Algorithmic aspects of vertex elimination on graphs*, SIAM J. Comput., 5 (1976), pp. 266–283.

[17] R. E. Tarjan and M. Yannakakis, *Simple linear-time algorithms to test chordality of graphs, test acyclicity of hypergraphs, and selectively reduce acyclic hypergraphs*, SIAM J. Comput., 13 (1984), pp. 566–579.