A Linear Time Algorithm to Compute a Dominating Path in an AT-free Graph

Derek G. Corneil* Stephan Olariu† Lorna Stewart‡

January 31, 1998

Abstract

An independent set $\{x, y, z\}$ is called an asteroidal triple if between any pair in the triple there exists a path that avoids the neighborhood of the third. A graph is referred to as AT-free if it does not contain an asteroidal triple. We present a simple linear-time algorithm to compute a dominating path in a connected AT-free graph.

Keywords. asteroidal triple-free graphs, domination, algorithms

1 Introduction

A number of families of graphs including interval graphs [10], permutation graphs [6], trapezoid graphs [3, 5], and cocomparability graphs [8] feature a type of linear ordering of their vertex sets. It is precisely this linear ordering that is exploited in a search for efficient algorithms on these classes of graphs [2, 5, 7, 8, 9, 11, 12]. 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, for short). An independent triple $\{x, y, z\}$ is called an *asteroidal triple* if between any pair in the triple there exists a path that avoids the neighborhood of the third. In [4] it has been argued that the property of being asteroidal triple-free is what is enforcing the linear ordering of the vertex sets.

One interesting "certificate" of linearity is the existence of a path such that every vertex outside the path is adjacent to some vertex on the path. Such a path is called dominating. Recently the authors have proved [4] that in every connected AT-free graph some path is dominating. The purpose of this work is to show that such a path can be computed in linear time. It is interesting to note that a linear-time algorithm for this problem is not known even for cocomparability graphs, a strict subclass of AT-free graphs.

The remainder of the paper is organized as follows. Section 2 establishes notation and terminology that will be used throughout the paper and presents background information about AT-free graphs. Section 3 presents the details of our linear time algorithm. Finally Section 4 summarizes the results and poses some open problems.

*Department of Computer Science, University of Toronto, Toronto, Canada
†Department of Computer Science, Old Dominion University, Norfolk, U.S.A.
‡Department of Computing Science, University of Alberta, Edmonton, Canada

2 Background

All the graphs are undirected with no loops nor multiple edges. In general, we use graph theoretical terminology compatible with [7]. To present our results, we shall also define a number of new terms. All the paths in this note are chordless. A vertex u *misses* a path P if u is adjacent to no vertex on P; otherwise, u *intercepts* P. A path joining vertices x and y is termed an x, y-path. Throughout, an underlying connected AT-free graph $G = (V, E)$ with n vertices and m edges is assumed. A set D of vertices of G is *dominating* if every vertex in $V \setminus D$ is adjacent to some vertex in D. A dominating set that induces a path is referred to as a *dominating path*. For a vertex x in G, $N(x)$ denotes the set of all the vertices adjacent to x: as usual, we assume that $x \notin N(x)$; $N'(x)$ stands for the set of all the vertices adjacent to x in the complement $\overline{G}$ of G. Occasionally, in order to simplify the notation, we shall blur the distinction between a set of vertices and the subgraph of G it induces, using the same notation for both.

Let v be an arbitrary vertex in an AT-free graph. We say that vertices u and w are *unrelated* (with respect to v) if u misses some w, v-path and if w misses some u, v-path. We now state a result concerning unrelated vertices that will be useful in our arguments.

Proposition 2.1. [4] Let v be an arbitrary vertex of a connected AT-free graph. No connected component of the subgraph induced by $N'(v)$ contains unrelated vertices with respect to v. □

The well-known Breadth First Search (BFS, for short), starting from an arbitrary vertex v of G, partitions the vertices of G into *layers*, where all the vertices of a layer are at the same distance from v. In this paper we let $L_t(v)$ stand for the layer consisting of vertices at distance t from v. Let w be an arbitrary vertex in some layer $L_t(v)$. Every shortest w, v-path will be termed *direct*. Note that distinct vertices on a direct path belong to distinct layers.

A connected component C of $N'(v)$ is called *deep* if $C \cap L_t(v) \neq \emptyset$ for some $t \geq 3$. Since deep components play an important role in our algorithm we investigate a few of their properties. Our first result shows that the number of deep components cannot be too large. Specifically, we have the following

Lemma 2.2. Let v be an arbitrary vertex of a connected AT-free graph. The number of deep components of $N'(v)$ is at most 2.

Proof. Suppose not, and let C_1, C_2, C_3 be arbitrary mutually disjoint deep components of $N'(v)$. Further, let c_1, c_2, c_3, be vertices in the farthest layers of C_1, C_2, and C_3, respectively. It is now easy to confirm that $\{ c_1, c_2, c_3\}$ is an asteroidal triple. The conclusion follows. □

Let C be a deep component in G and let k be the largest subscript for which $L_k(v)$ has a nonempty intersection with C. Define the *anchor set A with respect to C* to contain the set of all vertices w in $C \cap L_k(v)$ for which

$$| N(w) \cap L_{k-1}(v) | \text{ is a minimum.} \tag{1}$$

The elements of the set A are referred to as *anchors*. For later reference we state the following result.

Lemma 2.3. Let C be a deep component with respect to v and let a be an arbitrary anchor in C. Then, every direct a, v-path dominates all vertices in C.

Proof. Assume, without loss of generality, that $a \in L_k(v)$. If the statement is false, we find a vertex u in C that misses some direct a, v-path P. Let $u \in L_i(v)$.

Note that in case $i < k$, a misses a direct u, v-path implying that a and u are unrelated, a contradiction to Proposition 2.1. Thus, we must have $i = k$. Since u misses P, u must be non-adjacent to some vertex in $N(a) \cap L_{k-1}$. Consequently, our choice of a specified in (1) guarantees the existence of a vertex in $N(u) \cap L_{k-1}$ missed by a. However, now a and u are unrelated, contradicting Proposition 2.1. □

3 The Algorithm

We are now in a position to spell out the details of our simple algorithm to compute a dominating path in a connected AT-free graph G.

Procedure Dominating_Path(G);
Input: a connected AT-free graph $G = (V, E)$;
Output: a dominating path P in G;
1. choose an arbitrary vertex v in G;
2. perform a BFS layering on G, starting from v;
3. **if** $N'(v) = \emptyset$ **then** return v and **stop**;
4. identify the deep components of $N'(v)$;
5. **if** the number of deep components is 2 **then**
    let C_1 and C_2 be the two deep components;
    select an anchor a in C_1 and an anchor b in C_2
6. **else if** the number of deep components is 1 **then**
    let C be the unique deep component;
    select an anchor a in C;
    let S stand for the set of vertices in $N(v)$ that belong to at least one direct a, v-path;
    let M be the set of vertices in $L_2(v) \setminus C$ that are adjacent to every vertex in S;
    if $L_2(v) \setminus (C \cup M)$ is empty **then**
        set $b \leftarrow v$
    else
        select a vertex b in $L_2(v) \setminus (C \cup M)$ that minimizes $| N(b) \cap N(v) |$
7. **else** {there is no deep component in $N'(v)$}
    select a vertex a in $L_2(v)$ that minimizes $| N(a) \cap N(v) |$;
    assume that a belongs to a component C of $N'(v)$;
    let M be the set of vertices in $L_2(v) \setminus C$ that are adjacent to every vertex in $N(a) \cap N(v)$;
    if $L_2(v) \setminus (C \cup M)$ is empty **then**
        set $b \leftarrow v$
    else
        select a vertex b in $L_2(v) \setminus (C \cup M)$ that minimizes $| N(b) \cap N(v) |$
8. choose an arbitrary direct a, v-path P_1;
9. choose an arbitrary direct b, v-path P_2;

10. return a chordless a, b-path P contained in $P_1 \cup P_2$.

Theorem 3.1. With a connected AT-free graph G with n vertices and m edges as input, procedure Dominating_Path correctly returns a dominating path in G in $O(n+m)$ time.
Proof. Let v be an arbitrary vertex of G and let P be the path returned by the procedure. In case $N'(v)$ is empty (i.e. v is adjacent to the remaining vertices of the graph) the correctness is clear. If $N'(v)$ contains a single component (i.e. $N'(v)$ is connected), the correctness follows from Lemma 2.3 in case the component is deep and by an easy ad-hoc argument otherwise.

For the proof of correctness in the general case we need the following intermediate results.
Lemma 3.2. If $N'(v)$ has two deep components, then P is dominating.
Proof. Suppose not, and let u be a vertex that misses P. By Lemma 2.3, u must lie in $N(v) \cup (N'(v) \setminus (C_1 \cup C_2))$. We claim that $\{u, a, b\}$ is an asteroidal triple.

To see that this is the case, note that trivially u misses an a, b-path. Since both C_1 and C_2 are deep, $N(a) \cap N(v) = N(b) \cap N(v) = \emptyset$. Thus there exists an obvious u, a-path missed by b and an obvious u, b-path missed by a. The conclusion follows. □
Lemma 3.3. If $N'(v)$ has a single deep component, then P is dominating.
Proof. Suppose that a and b have been selected in Step 6 of the procedure. If $b = v$ then P is dominating by Lemma 2.3 and the fact that $L_2(v) \setminus (C \cup M) = \emptyset$.

Now, suppose that $b \neq v$, and consider a vertex u that misses P. Lemma 2.3 and the definition of the set M guarantees that u belongs to $N(v) \cup (N'(v) \setminus (C \cup M))$. We claim that $\{u, a, b\}$ is an asteroidal triple. First, u misses an a, b-path. Since C is deep, $N(a) \cap N(v) = \emptyset$, and so a misses some u, b-path.

To see that b misses a u, a-path, note that since b is not in M, it is non-adjacent to some vertex s in S. In case u belongs to $N(v)$ the u, a-path missed by b is easily seen. If, however, u belongs to $N'(v) \setminus (C \cup M)$, then since u misses P, the choice of b implies that u has a neighbor u' in $N(v)$ missed by b. Now the desired u, a-path is contained in a path that proceeds from u to u', to v, to s, and to a in C. This completes the proof of Lemma 3.3. □
Lemma 3.4. If $N'(v)$ has no deep component, then P is dominating.
Proof. Let a and b be vertices selected in Step 7 of the procedure. Let u be a vertex that misses P. If u is in C, then the choice of a and the fact that u misses some vertex of $N(a) \cap N(v)$ implies the existence of a vertex in $N(v)$ adjacent to u and missed by a, contradicting Proposition 2.1. Thus, P dominates all vertices of C. Therefore, if $b = v$ then P is dominating since $L_2(v) \setminus (C \cup M) = \emptyset$.

If $b \neq v$ then $\{u, a, b\}$ is an asteroidal triple. Trivially, u misses an a, b-path.

To see that a misses a u, b-path, note that since b is not in M, b misses some vertex in $N(a) \cap N(v)$. The choice of a in Step 7, guarantees the existence of a vertex b' in $N(v)$ adjacent to b and missed by a. In case u belongs to $N(v)$ the u, b-path missed by a is contained in $\{b, b', v, u\}$. In case u belongs to $L_2(v)$, then an argument similar to that asserting the existence of b' guarantees the existence of a vertex u' in $N(v)$ adjacent to u but not to a. Now the desired u, b-path is contained in $\{b, b', v, u', u\}$.

Finally, to see that b misses a u, a-path, note that since b is not in M it misses a vertex a' in $N(a) \cap N(v)$. If u belongs to $N(v)$ the u, a-path missed by b is contained in $\{a, a', v, u\}$.

If, on the other hand, u belongs to $N'(v) \setminus M$, then the choice of b together with the fact that u misses P guarantees the existence of a vertex u' in $N(v)$ adjacent to u and missed by b. Now the desired path is contained in $\{u, u', v, a', a\}$, and the proof of Lemma 3.4 is complete. □

By virtue of Lemma 2.2, $N'(v)$ contains at most two deep components. Therefore, Lemmas 3.2–3.4 prove the correctness of the procedure.

To argue for the complexity, note that Step 2 and Step 4 run in $O(n+m)$ time. Selecting an anchor takes at most $O(m)$ time. The set S in Step 6 can be identified on the fly while performing BFS. The selection of b takes $O(m)$ time. Finally, Step 10 amounts to running BFS again on the graph induced by $P_1 \cup P_2$. Therefore, the overall running time is bounded by $O(n + m)$, and the proof of the theorem is complete. □

4 Concluding Remarks

The class of asteroidal triple-free graphs (AT-free graphs) is a natural generalization of a number of classes of graphs including interval graphs, permutation graphs, trapezoid graphs, and cocomparability graphs. Recently the authors have argued [4] that the property of being asteroidal triple-free is responsible for the linearity featured by all these classes of graphs.

An interesting result in [4] asserts that every connected AT-free graph contains a dominating path, that is, a path with the property that every vertex outside the path is adjacent to some vertex on the path. In this note we have shown that a dominating path in a connected AT-free graph can be computed in linear time.

Yet another interesting concept put forth in [4] is the notion of a *dominating pair*. Specifically, a pair (x, y) of vertices of a connected AT-free graph is a dominating pair if every x, y-path is dominating. In [4] it has been shown that every connected AT-free graph contains a dominating pair. Quite recently, Balakrishnan *et al.* [1] have exhibited a straightforward algorithm that finds all dominating pairs in an arbitrary n-vertex graph in $O(n^3)$ time. In particular, their algorithm can be used to find a dominating pair in a connected AT-free graph in $O(n^3)$ time. Unfortunately, for large values of n this is prohibitively expensive. Second, their algorithm does not exploit the structure of AT-free graphs. Therefore, one expects that one can do better. In particular, it would be interesting to see whether the simple algorithm presented in this note can be extended to compute a dominating pair in a connected AT-free graph.

Acknowledgements:

D.G. Corneil and L. Stewart wish to thank the Natural Sciences and Engineering Research Council of Canada for financial assistance. S. Olariu was supported, in part, by the National Science Foundation under grant CCR-8909996.

References

[1] H. Balakrishnan, A. Rajaraman, and C. P. Rangan Connected domination and Steiner set on asteroidal triple-free graphs, a manuscript, 1993.

[2] F. Cheah, A recognition algorithm for II-graphs, Doctoral thesis, Department of Computer Science, University of Toronto, (available as a technical report), 1990.

[3] D.G. Corneil and P.A. Kamula Extensions of permutation and interval graphs, *Proceedings 18th Southeastern Conference on Combinatorics, Graph Theory and Computing* (1987), 267–276.

[4] D.G. Corneil, S. Olariu, L. Stewart, On the linear structure of graphs, *Proc. 19th International Workshop on Graph Theoretic Concepts in Computer Science*, WG'93, Arnhem, Holland, June 1993.

[5] I. Degan, M.C. Golumbic and R.Y. Pinter, Trapezoid graphs and their coloring, *Disc. Appl. Math.* 21 (1988), 35–46.

[6] S. Even, A. Pnueli and A. Lempel, Permutation graphs and transitive graphs, *Journal of the ACM* 19 (1972), 400-410.

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

[8] M.C. Golumbic, C.L. Monma and W.T. Trotter Jr., Tolerance graphs, *Disc. Appl. Math.* 9 (1984), 157-170.

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

[10] C.G. Lekkerkerker and J.C. Boland, Representation of a finite graph by a set of intervals on the real line, *Fundamenta Mathematicae* 51 (1962), 45-64.

[11] S. Olariu An optimal greedy heuristic to color interval graphs, *Information Processing Letters*, 37 (1991) 21–26.

[12] F.S. Roberts. *Graph Theory and Its Applications to Problems of Society.* SIAM Press, 1978.