Information Processing Letters 23 (1986) 131–134
North-Holland

22 October 1986

TOTAL DOMINATION IN INTERVAL GRAPHS

Alan A. BERTOSSI

Dipartimento di Informatica, Università di Pisa, 56100 Pisa, Italy

Communicated by K. Mehlhorn
Received February 1985
Revised 24 September 1985

A total dominating set of a graph G is a subset S of nodes such that each node of G is adjacent to some node of S. We present an $O(n^2)$ time algorithm for finding a minimum cardinality total dominating set in an interval graph (one which represents intersecting intervals on the line) by reducing it to a shortest path problem on an appropriate acyclic directed network.

Keywords: Total dominating set, interval graph, shortest path problem, acyclic directed network, polynomial-time algorithm

1. Introduction

In an undirected graph G(N, E), each node in N *dominates* all nodes joined to it by an edge. A *dominating set* is a subset D of nodes which dominate all nodes in N − D [4]. In this way, each node in N − D is joined by an edge to at least one node in D. A *total dominating set* is a dominating set D which also dominates itself. This means that also each node in D is joined by an edge to at least another node in D, or, equivalently, that the subgraph induced by D has no isolated node. A smallest such set is a *minimum total dominating set* [5,9].

The problem of determining a minimum total dominating set along with several closely related variants has been investigated remarkably (see [7,8] for extensive bibliographies). Indeed, this problem has important applications in the optimum location of facilities in a network (see, e.g., [4,11]).

For general graphs, the total dominating set problem is a computationally intractable one. In fact, it is an NP-*complete* problem [8]. In other words, it is very unlikely that this problem can be solved by an efficient algorithm whose running time is bounded by a polynomial in the size of the input. The total dominating set problem remains NP-complete even for special classes of graphs, such as *bipartite* graphs [1,3], *split* graphs [1,3], and *undirected path* graphs [9], these two latter classes of graphs being subclasses of the well-known *chordal* graphs. Thusfar, the only class of graphs for which a polynomial-time algorithm has been devised is that of *trees* [9], which in turn is another subclass of chordal graphs. In [9], the question of determining whether a polynomial-time algorithm exists to solve the total dominating set problem for yet another subclass of chordal graphs known as *interval* graphs was mentioned as being an interesting open question. Interval graphs are exactly those graphs which represent intersecting intervals on the line, and they have found application in several disciplines (see, e.g., [6,11]).

In this paper, we exhibit an $O(n^2)$ time algorithm for finding a minimum total dominating set in an interval graph by reducing it to a shortest path problem on an appropriate acyclic directed network.

2. Preliminaries

An *interval family* is a set $I = \{I_1, \ldots, I_n\}$ of intervals on the real line, where $I_i = [a_i, b_i]$ for $i = 1, 2, \ldots, n$. A graph G is an *interval graph* if there exist an interval family and a one-to-one correspondence between the nodes of the graph and the intervals of the family such that two nodes are joined by an edge if and only if their corresponding intervals overlap.

A linear time algorithm exists for deciding whether a given graph G is an interval graph and constructing, in the affirmative case, the required interval family I [2,6]. Therefore, we shall denote an interval graph as G(I) and directly deal only with the intervals instead of the nodes. Without loss of generality, we assume that each interval contains both its endpoints and that no two intervals share a common endpoint [6].

In this way, a total dominating set for an interval graph G(I) corresponds to a subset S of intervals in I such that every interval in I overlaps with some interval in S. Note that this property must hold for intervals in S, too. Thus, the subgraph induced by S (obtained by deleting all intervals appearing in I but not in S) cannot have any isolated interval.

Lastly, here we consider *connected* interval graphs only, that is, graphs such that between any two intervals in I there exists a sequence of distinct pairwise overlapping intervals joining them. If G(I) is unconnected, we can apply the reasonings that will follow in the next section to each *connected component.* (Given I, such components can be trivially detected in O(n log n) time.) We finally assume that there are at least two intervals in I, since otherwise G(I) has no total dominating set at all.

3. The algorithm

Theorem 3.1. *If* G(I) *has no interval properly containing all the others, then there exists a minimum total dominating set* S *having no interval properly contained within another interval in* I.

Proof. Assume G(I) to be an interval graph having the above stated property and let T be a minimum total dominating set for it. Without loss of generality, we can assume that for each $I_i \in T$ there is no $I_m \in I - T$ with $I_i \subset I_m$, since otherwise I_i and I_m could be swapped to obtain a minimum total dominating set having this property. If no interval in T is properly contained within any other interval in T, we are done. Otherwise, assume there are two intervals in T, say I_i and I_j, such that $I_i \subset I_j$. Obviously, for any interval $I_k \in T$ distinct from both I_i and I_j we have that $I_k \cap I_j = \emptyset$, since otherwise $T - \{I_i\}$ would be a smaller total dominating set. Let us now suppose that an interval $I_h \in I - T$ exists with $a_h < a_j < b_h < b_j$ or $a_j < a_h < b_j < b_h$. If more than one such I_h exists, we choose one which is not properly contained within any other interval. We can swap I_h and I_i in T and thus obtain another minimum total dominating set T′. This change does not introduce any additional proper inclusion. Indeed, no interval $I_k \in T'$ can exist with $I_k \subset I_h$, since otherwise $T' - \{I_k\}$ would be a smaller total dominating set. Let us now see that such an I_h always exists. Suppose not. Since G(I) is connected, for every $I_k \in I - T$, $I_k \subset I_j$ must result, and thus I_j properly contains any other interval in I, contrary to the assumption.

Repeating this reasoning for each interval in T properly contained within another interval leads to a total dominating set S having the same cardinality as T, which has not interval properly contained within another interval of I. □

The anomalous case in which G(I) has an interval properly containing all the others can be detected in O(n) time. One only needs to compute

$$a_i = \min_{1 \leqslant k \leqslant n} \{a_k\} \quad \text{and} \quad b_j = \max_{1 \leqslant k \leqslant n} \{b_k\}.$$

If $i = j$, we are done. A minimum total dominating set is then given by only two intervals: I_i along with any other interval.

Let us now see how we can find a minimum total dominating set which satisfies the property stated in Theorem 3.1. First of all, we augment I with two dummy intervals, say I_0 and I_{n+1}, such that

$$b_0 < \min_{1 \leqslant k \leqslant n} \{a_k\} \quad \text{and} \quad a_{n+1} > \max_{1 \leqslant k \leqslant n} \{b_k\}.$$

Let $I' = I \cup \{I_0, I_{n+1}\}$. We assume that the intervals in I' are indexed by increasing values of their leftmost endpoints, namely, $a_0 < a_1 < \cdots < a_n < a_{n+1}$.

We define a *directed network* D(N, A) as follows. The nodes in N correspond to the intervals in I' which are not properly contained within other intervals. The arcs in A are partitioned into two kinds, namely, $A = B \cup C$, with $B \cap C = \emptyset$. Arcs in B join some overlapping intervals, namely, $(i, j) \in B$ iff $a_i < a_j < b_i < b_j$, while arcs in C join some nonoverlapping intervals, namely $(i, j) \in C$ iff $b_i < a_j$ and there is no interval I_h with $b_i < a_h < b_h < a_j$. All arcs in A have unit *length,* except the arcs of type (0, i), which have zero length. Observe that D is *acyclic,* since the intervals in I' are ordered by increasing a_i's, and for each arc $(i, j) \in A$ we have that $a_i < a_j$.

Theorem 3.2. *Any total dominating set having no interval properly contained within any interval in* I *corresponds to a path between nodes* 0 *and* n + 1 *in* D *which does not include any two consecutive arcs of* C.

Proof. Let p be a path from node 0 to node n + 1 in D with no two consecutive arcs of C. Define

$$S = \{I_i : \text{node i appears in p}, i \neq 0, i \neq n+1\}.$$

For each arc (i, j) in p, I_i and I_j dominate all the intermediate intervals $I_{i+1}, I_{i+2}, \ldots, I_{j-1}$ (note that intervals I_0 and I_{n+1} do not dominate any interval of I). Since no two consecutive arcs in p, say (i, j) and (j, k), can belong to C, either $I_i \cap I_j \neq \emptyset$ or $I_j \cap I_k \neq \emptyset$. Thus, S is a total dominating set for G(I).

For the other direction, let S be a total dominating set. Without loss of generality, let the intervals in S be ordered by increasing indices. Consider two consecutive intervals in S, say I_i and I_j. Since the intervals in I are ordered by increasing a_i's and no interval in S is properly contained within any interval in I, $a_i < a_j$ and $b_i < b_j$ must result. If $I_i \cap I_j \neq \emptyset$, then surely arc (i, j) appears in B. Otherwise, if $I_i \cap I_j = \emptyset$, then arc (i, j) must appear in C, since otherwise an intermediate interval would exist, say I_h, with $b_i < a_h < b_h < a_j$, which is not dominated by S. Thus, let p be the path of D which, starting at node 0, passes through all the nodes corresponding to intervals in S, crossed in the above-mentioned order, and terminates at node n + 1. Since S is a total dominating set, the resulting induced subgraph has no isolated interval. Therefore, the path p does not include any two consecutive arcs of C. □

By Theorems 3.1 and 3.2, the problem of finding a minimum total dominating set in an interval graph has been reduced to that of finding a shortest path in an appropriate network with some sequencing constraints on certain arcs. However, the latter problem can in turn be reduced to an ordinary shortest path problem. To see this, we define another directed network D′ by splitting each node i, $1 \leqslant i \leqslant n$, into two nodes, say i_{in} and i_{out}. These two nodes are joined by the arc (i_{in}, i_{out}), whose length is zero. For each arc $(i, j) \in B$, we introduce in D′ the arc (i_{out}, j_{in}) with unit length. For each arc $(i, j) \in C$, instead, we introduce in D′ the arc (i_{in}, j_{out}), also with unit length. Lastly, for each arc $(0, i) \in A$ we add to D′ the arc $(0, i_{out})$ with zero length, while for each arc $(i, n+1) \in A$ we add to D′ the arc $(i_{in}, n+1)$ with unit length.

One can readily check that a shortest path in D′ between nodes 0 and n + 1 corresponds to a minimum total dominating set in which interval $I_i \in I$ is included if and only if either i_{in} or i_{out} or both are crossed by the path.

It is easy to see that D′ can be constructed directly from I in $O(n^2)$ time. Moreover, D′ is acyclic, too. Then, a shortest path in D′ can be found in $O(n^2)$ time [10]. Thus, the overall time complexity to find a minimum total dominating set in an interval graph is $O(n^2)$.

4. Further research

We believe that further research is needed in the following main directions. On the one hand, one can search for a faster algorithm for the total dominating set problem in an interval graph, thus improving our $O(n^2)$ time bound. On the other hand, one can investigate the weighted analogue of this problem, in which a *weight* is associated to each interval, and one is asked to find a total dominating set having minimum overall weight. However, we believe that the kind of reduction employed here can easily be modified so as to work in the weighted case, too.

References

[1] A.A. Bertossi, Dominating sets for split and bipartite graphs, Inform. Process. Lett. 19 (1984) 37–40.

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

[3] G.J. Chang and G.L. Nemhauser, The *k*-domination and *k*-stability problems on sun-free chordal graphs, SIAM J. Algebraic Discr. Meths. 5 (1984) 332–345.

[4] E.J. Cockayne, Domination in undirected graphs—a survey, in: Y. Alavi and D.R. Lick, eds., Theory and Applications of Graphs in America's Bicentennial Year (Springer, Berlin, 1978).

[5] E.J. Cockayne, R.M. Dawes and S.T. Hedetniemi, Total domination in graphs, Networks 10 (1980) 211–219.

[6] M.C. Golumbic, Algorithmic Graph Theory and Perfect Graphs (Academic Press, New York, 1980).

[7] M. Hujter, Bibliography on the minimum dominating set problem, Tech. Rept., RUTCOR, Rutgers Univ., 1984.

[8] D.S. Johnson, The NP-completeness column—an ongoing guide, J. Algorithms 5 (1984) 147–160.

[9] R. Laskar, J. Pfaff, S.M. Hedetniemi and S.T. Hedetniemi, On the algorithmic complexity of total domination, SIAM J. Algebraic Discr. Meths. 5 (1984) 420–425.

[10] E.L. Lawler, Combinatorial Optimization—Networks and Matroids (Holt, Rinehart & Winston, New York, 1976).

[11] F.S. Roberts, Graph Theory and Its Applications to Problems of Society (SIAM Press, Philadelphia, PA, 1978).