Permutation Graphs: Connected Domination and Steiner Trees

Charles J. Colbourn
Department of Computer Science
University of Waterloo
Waterloo, Ontario, Canada N2L 3G1

Lorna K. Stewart
Department of Computer Science
University of Toronto
Toronto, Ontario, Canada M5S 1A1

Research Report CS-85-02

February 1985

Permutation Graphs: Connected Domination and Steiner Trees

Charles J. Colbourn

Department of Computer Science,
University of Waterloo,
Waterloo, Ontario, N2L 3G1,
CANADA.

Lorna K. Stewart

Department of Computer Science,
University of Toronto,
Toronto, Ontario, M5S 1A1,
CANADA.

ABSTRACT

Efficient algorithms are developed for finding a minimum cardinality connected dominating set and a minimum cardinality Steiner tree in permutation graphs. This contrasts with the known NP-completeness of both problems on comparability graphs in general.

1. Introduction

A *dominating set* of vertices in a graph G=(V,E) is a set S for which every vertex of V-S has a neighbour in S. Dominating sets are *independent* if the subgraph induced on S is void, *total* if the subgraph induced on S has no isolated vertices, and *connected* if the subgraph induced on S is connected. The size of a minimum cardinality dominating set in a graph is called the *domination number;* analogously, one defines the graph's *independent domination number, total domination number,* and *connected domination number*.

Finding minimum cardinality dominating sets of these various types has attracted much attention. In the following table, we summarize the current status of these four domination problems. Note that many of the results are implied by the inclusions:

(1) {bipartite graphs} $\subseteq$ {comparability graphs}

(2) {split graphs} $\subseteq$ {chordal graphs}

(3) {interval graphs} $\subseteq$ {directed path graphs} $\subseteq$ {undirected path graphs} $\subseteq$ {chordal graphs}

(4) {interval graphs} $\subseteq$ {strongly chordal graphs} $\subseteq$ {chordal graphs}

(5) {cographs} $\subseteq$ {permutation graphs} $\subseteq$ {comparability graphs}

Complexity of Domination Problems				
graphs	domination	independent	connected	total
bipartite	NP-c [7]	NP-c [5]	NP-c [13]	NP-c [13]
comparability	NP-c [7]	NP-c [5]	NP-c [13]	NP-c [13]
split	NP-c [1]	P [8]	NP-c [24]	NP-c [24]
chordal	NP-c [1]	P [8]	NP-c [24]	NP-c [23]
strongly chordal	P [9]	P [8]	P [21]	
interval	P [9]	P [8]	P [21]	
directed path	P [2]	P [8]		
undirected path	NP-c [2]	P [8]		NP-c [23]
series-parallel	P [12]	P [14]	P [21]	P [14]
cographs	P [10]	P [10]	P [5]	P [19]
permutation	P [10]	P [10]	this paper	P [19]

Connected domination involves finding a smallest connected subgraph which dominates the remainder of the vertices. This bears some similarity to the Steiner tree problem, a central problem in network analysis and design. Given a graph G=(V,E), we identify a set T $\subseteq$ V of *target vertices*. Then a *Steiner tree* for T in G is a set S $\subseteq$ V-T of vertices, such that S $\cup$ T induces a connected subgraph. A minimum cardinality Steiner tree is a set S of smallest cardinality; we call the problem of finding such a set CARDST to distinguish it from the usual edge-weighted Steiner tree problem widely studied in the networks and graph theory literature. White, Farber, and Pulleyblank [21] observe that whenever the complexities of CARDST and connected domination are currently known, they are the same. In fact, CARDST is NP-complete for bipartite and comparability graphs [11], split and chordal graphs [21], but can be solved efficiently for strongly chordal graphs [21] and series-parallel graphs [6,16,20].

In this paper, we examine the connected domination and CARDST problems on permutation graphs, and develop efficient algorithms for each. The solutions are remarkably similar; however, we develop different methods in the two cases, exploiting the structure of minimum cardinality connected dominating sets. It is perhaps important to note that the solution for connected domination given here differs considerably from the Farber-Keil approach used in other domination problems on permutation graphs.

The definition of the various families referred to are standard, and can be found in many of the references; we repeat the necessary ones here. A *permutation graph* is a graph for which there is a labeling $\{v_1, ..., v_n\}$ of the vertices and a permutation π of $\{1, ..., n\}$ for which $(i-j)(\pi(i)-\pi(j)) < 0$ if and only if (v_i, v_j) is an edge. All permutation graphs have a transitive orientation, and hence are comparability graphs; in fact, a graph is a permutation graph if and only if both the graph and its complement are comparability graphs [15]. A subclass of permutation graphs which has been studied extensively is the class of cographs. A graph is a *cograph* if and only if it has no induced subgraph which is a 4-vertex path. Many equivalent characterizations are known [4,18].

2. Connected Domination

In this section, we develop a simple algorithm for finding a minimum cardinality connected dominating set (MCCDS) in a permutation graph. Connected domination is NP-complete for comparability graphs. However, it is not hard to see that connected domination has a trivial solution for cographs [5]. In fact, a cograph which does not have a single dominating vertex must have a pair of adjacent vertices which forms a dominating set; otherwise, a four vertex path would be induced. This is of interest here, since {cographs} $\subseteq$ {permutation graphs} $\subseteq$ {comparability graphs}.

The algorithm for finding a MCCDS in a permutation graph employs a geometric representation. Consider two columns, each consisting of the integers $\{1, ..., n\}$ in order, and a permutation π. A *line* connects i in the left column with $\pi(i)$ in the right. A permutation graph is obtained by taking the n lines as vertices; edges are denoted by crossings of the lines.

Each line e has endpoints in the left and right columns; $left(e)$ and $right(e)$ denote the indices of these endpoints. Then the *left-span* of a set $L = \{e_1, ..., e_k\}$ of lines is a set $\text{LEFT}(L) = \{i \mid i \geq min(left(e) | e \in L) \text{ and } i \leq max(left(e) | e \in L)\}$. The *right-span* of L, $\text{RIGHT}(L)$ is defined analogously. The pair $(\text{LEFT}(L), \text{RIGHT}(L))$ is the *span* of L. Two spans (L,R) and (L',R') are said to *intersect* if one of the following holds:

(1) $L \cap L' \neq \phi$,

(2) $R \cap R' \neq \phi$, or

(3) $\max(L') > \max(L)$ and $\min(R') < \min(R)$.

Two sets of lines L_1 and L_2 are said to *intersect* if their spans intersect. A set of lines is *connected* if there is no nontrivial partition into two sets of non-intersecting lines.

Given any permutation graph, Spinrad's algorithm will produce this geometric representation in $O(n^2)$ time [17], an improvement on the earlier $O(n^3)$ algorithm [15]. Hence, it suffices to determine dominating sets in this geometric setting.

Lemma 2.1: A connected dominating set is a connected set of lines which each other line intersects.

Proof: A connected dominating set induces a connected subgraph, and therefore the set of lines in the dominating set must be connected. Moreover, any line not in the dominating set which does not intersect a line in the dominating set is not dominated in the corresponding permutation graph. ●

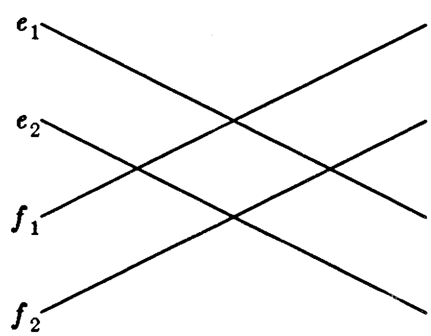
The structure of MCCDS is somewhat special; we explore this in a sequence of preliminary lemmas.

Lemma 2.2: Let M be an MCCDS. Then M contains no three lines all intersecting.

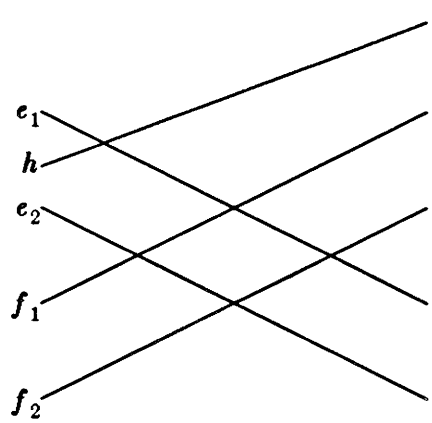
Proof: Let e_1, e_2, and e_3 be the three lines, with $left(e_1) < left(e_2) < left(e_3)$. Since all three cross, $right(e_1) > right(e_2) > right(e_3)$. But then any line crossing e_2 must cross either e_1 or e_3 as well. Then M - e_2 is a connected dominating set, which is a contradiction. ●

Lemma 2.3: Let M be an MCCDS. Then there is an MCCDS of the same cardinality in which there are no four lines which induce a four-vertex cycle.

Proof: Suppose there are four lines which induce a four-vertex cycle. Geometrically, they appear as two nonintersecting pairs e_1 and e_2, f_1 and f_2, with each e_i crossing each f_j. This is illustrated here:



Suppose there is a line $h \in$ M crossing e_1. Without loss of generality, using lemma 2.2 and symmetries, we obtain



Then any line crossing f_1 also crosses another line in M, and hence M-f_1 is a connected dominating set contradicting minimality of M. Thus the only possibility is that M=$\{e_1,e_2,f_1,f_2\}$. Then there must be a line which crosses e_1 but not f_1, for otherwise e_1 is redundant in M, contradicting the assumption that M is minimal. Consider such a line h and replace f_1 by h in M; this produces a connected dominating set of the same cardinality which does not induce a four-vertex cycle. ●

Since induced subgraphs of permutation graphs are themselves permutation graphs, and since permutation graphs have no induced cycles of lengths five and greater [22], lemmas 2.2 and 2.3 establish that a MCCDS induces a *tree*. In fact, we can establish an even stronger result:

Theorem 2.4: An MCCDS in a permutation graph induces a path.

Proof: In view of lemmas 2.2 and 2.3 we need only exclude stars on three edges. Such a star, geometrically, is a triple of three pairwise nonintersecting lines e_1, e_2, and e_3 and a fourth line f crossing all three. Supposing that $left(e_1) < left(e_2) < left(e_3)$, it is immediate that e_2 can be removed from the set. ●

The algorithmic importance of theorem 2.4 is that, in order to find a MCCDS in a permutation graph, we need only find the minimum cardinality dominating induced path. An algorithm to do this for a connected permutation graph is quite straightforward, and is outlined here. A line e for which there is no line f having both $left(f)<left(e)$ and $right(f)<right(e)$ is termed *initial*.

```
{locate set I of initial lines}
let l be the line with left(l) = 1.
let r be the line with right(r) = 1.
I = {l,r}
minr = right(l)
for i = 1 to left(r) do
   let q be the line with left(q) = i
   if right(q) < minr then
      I = I ∪ {q}
      minr = right(q)

{I now contains all initial lines}

minsize = n  {all n lines form a connected dominating set}
for each line e in turn do
   {try e as the first line of the path, i.e. the line with lowest left() in the path}
   if e ∈ I
   then
      L_0 = {1, ..., left(e)}
      R_1 = {1, ..., right(e)}
      L_1 = R_0 = φ
      i = 1
   else
      let f be an initial line not crossing e for which
         right(f) is minimal
      let g be a line crossing both e and f for which
         left(g) is maximum; if no such line exists or left(g)<left(e),
         abandon e as a possible first line
      L_1 = L_2 = {left(e)+1, ..., left(g)}
      R_1 = R_2 = {right(g)+1, ..., right(e)}
      i = 2
      {now the first two lines are e and g, in that order}
   endif

   done = false
   while not done
      if either  every line l has left(l)≤max(L_{i-1}) or right(l)≤max(R_i)
         or      every line l has left(l)≤max(L_i) or right(l)≤max(R_{i-1})
      then
         done = true
         minsize = min(i,minsize)
      else
         i = i+1
         maxleft = max{j | some l has left(l)=j and right(l)∈R_{i-1}}
                  (undefined when R_{i-1}=φ)
         L_i  = {max(L_{i-2})+1, ..., maxleft}
                  (empty when L_{i-2}=φ or maxleft undefined)
         maxright = max{j | some l has right(l)=j and left(l)∈L_{i-1}}
         R_i  = {max(R_{i-2})+1, ..., maxright}
      endif
   endwhile
endfor

{result is minsize}
```

Minsize gives the connected domination number; it is a simple matter to retain the lines themselves and produce the MCCDS.

This algorithm demonstrates that

Theorem 2.5: A MCCDS in an n-vertex permutation graph can be found in $O(n^2)$ time.

Proof:

The geometric representation can be produced in $O(n^2)$ time [17]. Once done, initial lines can be classified in $O(n)$ time. Each of the n lines is selected as the first line; we must show that $O(n)$ time is spent per selection. When the first line is initial, the sets are constructed in $O(1)$ time by retaining only the smallest and largest members of each set. The update operation requires time linear in the size of the set; $O(n)$ operations are required in total, since each line is considered in at most four sets (two left, two right). When the first line is not initial, the only important note is that there is a unique second line which can be located in $O(n)$ time.

To verify correctness, observe that any MCCDS $\{e_1, ..., e_k\}$ induces sets L'_i and R'_i; $L'_i = \{1, ..., \max\{left(e_j) \mid j>i\}$. But at each step of the algorithm, these sets are maximized in size, and hence $k \geq$ minsize. ●

3. Steiner trees

We again employ the geometric representation of a permutation graph in solving the CARDST problem. Initially, we have a classification of lines into two types: *target lines* and *non-target lines*. The CARDST problem can be formulated as requiring the selection of the minimum number of non-target lines, which when included with all of the target lines induces a connected set.

We can recast this problem as follows. We are to determine the minimum number of non-target lines required to connect all target lines intersecting the span $(\{1,...,n\},\{1,...,n\})$. To do this, we determine the minimum number of non-target lines required to connect all target lines intersecting the span $(\{1,...,i\},\{1,...,j\})$, the *$i,j$-span*. Whenever there is a target line not intersecting the i,j-span, we further insist that the non-target lines chosen, together with the target lines intersecting the i,j-span, intersect all lines having exactly one end in the i,j-span. This minimum number is then denoted $\phi(i,j)$. We develop some simple constraints on $\phi(i,j)$.

Lemma 3.1: $\phi(i,j)$ satisfies the following inequalities:

(a) $\phi(1,1)=0$.

(b) $\phi(i,j) \leq \phi(i+1,j)$ and $\phi(i,j) \leq \phi(i,j+1)$.

(c) suppose there is a target line from k to l intersecting the i,j-span; then $\phi(max(i,k),max(j,l)) = \phi(i,j)$.

(d) suppose there is a non-target line from k to l intersecting the i,j-span; then $\phi(max(i,k),max(j,l)) \leq \phi(i,j)+1$.

(e) if every target line intersects the i,j-span then $\phi(n,n)=\phi(i,j)$.

(f) when there is a target line from i to j which is initial among the target lines, $\phi(i,j) = \phi(1,1)$.

Proof: All follow easily from the definition. ●

Theorem 3.2: ϕ is the maximum function satisfying inequalities (a)-(f).

Proof:

Any selection of non-target lines which produce a solution to CARDST appears as a sequence of inequalities in the list (a)-(f). ●

It is now a simple matter to compute $\phi(n,n)$ using dynamic programming techniques; we present a somewhat different technique here. Construct a directed graph with n^2 vertices $\{(i,j) \mid 1 \leq i \leq n,\ 1 \leq j \leq n\}$. Directed edges from $(i+1,j)$ to (i,j) and from $(i,j+1)$ to (i,j) appear with cost 0. A directed edge from (1,1) to (i,j) of cost 0 appears whenever there is an initial target line from i to j. A directed edge from (i,j) to (n,n) of cost 0 appears whenever every target line intersects the i,j-span. Whenever there is a target line from i to j, an edge of cost 0 from (i,k) to (l,j) is added for each $k \leq j$ and $l \geq i$; symmetrically, an edge of cost 0 from (k,j) to (i,l) is added for each $k \leq i$ and $l \geq j$. Finally, for any non-target line from i to j, the same edges are added, but each with cost 1. The Steiner tree is now easy to find; one simply finds a minimum cost path from (1,1) to (n,n) in this digraph. The cost of the path is the number of non-target lines chosen. Moreover, from the edges of cost 1 chosen, one can produce an actual selection of non-target lines.

Theorem 3.3: A minimum cardinality Steiner tree in an n-vertex permutation graph can be found in $O(n^3)$ time.

Proof: The required digraph can easily be constructed in $O(n^3)$ time. By theorem 3.2, any Steiner tree induces a directed path from (1,1) to (n,n) and vice versa. Minimum cost paths can be found using breadth-first search, for example, in time proportional to the number of edges in the digraph. ●

4. Weighted Connected Domination

In many practical applications, there is a cost, or weight, associated with the inclusion of a particular vertex in the dominating set. Thus much consideration has been given to the solution of weighted domination problems. Here each vertex has a weight, and the objective is to find a dominating set with minimum weight. The ideas of sections 2 and 3 combine nicely to yield an $O(n^3)$ algorithm for weighted connected domination in permutation graphs. We develop such an algorithm in this section. In (cardinality) connected domination, one could assume that the next line selected caused the largest increase in the span covered so far; in the weighted case, this need not be true. This consideration of all lines, rather than just those which maximize increase in covered span, is easily handled using ideas from the CARDST algorithm.

We define $\psi(i,j)$ to be the weight of a minimum weight connected dominating set covering all lines intersecting the i,j-span. Then we have the following inequalities for ψ:

Lemma 4.1:
(a) $\psi(0,0)=0$.
(b) if there is an initial line from i to j of weight k, $\psi(i,j)\leq k$.
(c) if there is a pair of lines, one from i to j of cost k, and another from i' to j' of cost k' which together intersect all initial lines, $\psi(i,j')\leq k+k'$.
(d) if there is a line from i to j of cost k which intersects the i',j'-span, $\psi(max(i,i'),max(j,j'))\leq\psi(i',j')+k$.
(e) if there is no line not intersecting the i,j-span, $\psi(n,n)=\psi(i,j)$.
Proof: All follow directly from the definition. ●

The details from this point on parallel the method for CARDST very closely and are omitted here. The same "shortest paths" approach leads to an $O(n^3)$ algorithm for weighted connected domination.

5. Conclusions

The methods in this paper extend the research of Farber and Keil [10] to include a further domination problem which has been widely studied, and they also support the contention of White, Farber and Pulleyblank [21] that cardinality Steiner tree and connected domination are algorithmically closely related problems. The topic which we believe is of most significance for future research is to account for the remarkably similar behaviour of CARDST and connected domination algorithmically.

Acknowledgments

Research of the first author is supported by NSERC Canada under grant number A0579. Thanks are due to both referees for suggesting improvements in the presentation.

References

[1] K.S. Booth, "Dominating sets in chordal graphs", Research report CS-80-34, University of Waterloo, 1980.

[2] K.S. Booth and J.H. Johnson, "Dominating sets in chordal graphs", *SIAM Journal on Computing* 11 (1982) 191-199.

[3] D.G. Corneil and J.M. Keil, "A dynamic programming approach to the dominating set problem on k-trees", to appear.

[4] D.G. Corneil, H. Lerchs, and L. Stewart Burlingham, "Complement reducible graphs", *Discrete Applied Mathematics* 3 (1981) 163-174.

[5] D.G. Corneil and Y. Perl, "Clustering and domination in perfect graphs", *Discrete Applied Mathematics* 9 (1984) 27-40

[6] G. Cornuejols, J. Fonlupt, and D. Naddef, "The graphical travelling salesman problem and some related integer polyhedra", Research report 378, Laboratoire d'Informatique et de Mathematiques appliquees de Grenoble, 1983.

[7] A.K. Dewdney, "Fast Turing Reductions between Problems in NP 4", Report 71, University of Western Ontario, 1981.

[8] M. Farber, "Independent domination in chordal graphs", *Operations Research Letters* 1 (1982) 134-138.

[9] M. Farber, "Domination, independent domination and duality in strongly chordal graphs" *Discrete Applied Mathematics* 7 (1984) 115-130.

[10] M. Farber and J.M. Keil, "Domination in permutation graphs", *Journal of Algorithms,* to appear.

[11] M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness,* Freeman and Sons, San Francisco CA, 1979.

[12] T. Kikuno, N. Yoshida, and Y. Kokuda, "A linear algorithm for the domination number of a series-parallel graph", *Discrete Applied Mathematics* 5 (1983) 299-311.

[13] J. Pfaff, R. Laskar and S.T. Hedetniemi, "NP-completeness of connected and total domination and irredundance for bipartite graphs", Technical Report 428, Clemson University, 1983.

[14] J. Pfaff, R. Laskar and S.T. Hedetniemi, "Linear algorithms for independent domination and total domination in series-parallel graphs", Technical Report 441, Clemson University, 1984.

[15] A. Pnueli, A. Lempel, and S. Even, "Transitive orientation of graphs and identification of permutation graphs", *Canadian Journal of Mathematics* 23 (1971) 160-175.

[16] R.L. Rardin, R.G. Parker, and M.B. Richey, "A polynomial algorithm for a class of Steiner tree problems on graphs", ISE Report J-82-5, Georgia Institute of Technology, 1982.

[17] J. Spinrad, "Transitive orientation in $O(n^2)$ time", Proc. Fifteenth ACM Symposium on the Theory of Computing (1983) 457-466.

[18] L. Stewart, "Cographs: a class of tree representable graphs", M.Sc. Thesis, University of Toronto; also technical report 126/78, University of Toronto, 1978.

[19] L. Stewart, Ph.D. thesis, Department of Computer Science, University of Toronto, forthcoming.

[20] J.A. Wald and C.J. Colbourn, "Steiner trees, partial 2-trees, and minimum IFI networks", *Networks* 13 (1983) 159-167.

[21] K. White, M. Farber, and W.R. Pulleyblank, "Steiner trees, connected domination, and strongly chordal graphs", *Networks,* to appear.

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

[23] R. Laskar, J. Pfaff, S.M. Hedetniemi and S.T. Hedetniemi, "On the algorithmic complexity of total domination", *SIAM J. Alg. Disc. Meth.,* 5 (1984) 420-425.

[24] R. Laskar and J. Pfaff, "Domination and irredundance in split graphs", Technical Report 430, Clemson University, 1983.