Discrete Applied Mathematics 13 (1986) 287-292
North-Holland

A LINEAR ALGORITHM FOR FINDING A MINIMUM DOMINATING SET IN A CACTUS

S.T. HEDETNIEMI*
Department of Computer Science, Clemson University, Clemson, SC 29631, USA

Renu LASKAR* and John PFAFF
Department of Mathematical Sciences, Clemson University, Clemson, SC 29631, USA

Received 1 February 1985

A dominating set in a graph $G=(V,E)$ is a set of vertices D such that every vertex in $V-D$ is adjacent to at least one vertex in D. A cutvertex in a connected graph G is a vertex whose removal results in a disconnected graph. A block in a graph G is a maximal connected subgraph of G having no cutvertices. A cactus is a graph in which each block is either an edge or a cycle. In this paper we present a linear time algorithm for finding a minimum order dominating set in a cactus.

1. Introduction

We consider undirected graphs $G=(V,E)$ without loops or multiple edges. A dominating set in G is a set D of vertices such that every vertex in $V-D$ is adjacent to at least one vertex in D. The domination number of a graph G, denoted $\gamma(G)$, is the minimum number of vertices in a dominating set.

Although the notion of dominating sets of queens on chessboards dates back to the 1800's [1], the modern study of domination can be attributed initially to Ore [11] and Berge [2]. For a survey of results on domination see Cockayne and Hedetniemi [6], Cockayne [4] or Laskar and Walikar [9].

For arbitrary graphs the problem of finding a minimum dominating set is NP-complete [8]. For the special case of trees, Cockayne, Goodman and Hedetniemi [5] presented a linear time algorithm, which was generalized by Natarajan and White [10] for weighted trees.

A graph G is chordal if every cycle of length greater than 3 has a chord (i.e. an edge joining two non-consecutive vertices of the cycle). Booth and Johnson [3] established that the problem of finding minimum dominating set in chordal graphs is NP-complete. However, Farber [7] has shown that for strongly chordal graphs, a subclass of chordal graphs, the domination problem is linear. A block of a graph

* This research was partly supported by National Science Foundation Grants ISP-8011451 (EPSCOR) and MCS-8307832.

is a maximal nonseparable (i.e. connected, nontrivial, and has no cut vertices) subgraph. A graph G is chordal if and only if each block of G is chordal. On the other hand, if each block of G is complete, then G is a block graph. Block graphs are strongly chordal, and hence the domination problem for block graphs is linear. A graph G is a cactus if each block is either an edge or a cycle. Thus, a tree is a cactus in which block is an edge. An endblock of a cactus is a block containing at most one cut vertex.

This paper presents a linear time algorithm, called MCACTUS, for finding a minimum dominating set (γ-set) in a cactus, which in turn uses an algorithm called MCYCLE for finding a γ-set in a cycle.

2. A linear algorithm for finding a γ-set in a cycle

Let $V(G)$ be partitioned into three subsets F, B, and R, where F consists of free vertices, B consists of bound vertices, and R consists of required vertices. Following Cockayne, Goodman and Hedetniemi [5], a mixed dominating set D in G is defined as follows:

(1) $R \subset D$.

(2) If $v \in B$, then v is either in D or adjacent to some vertex in D.

Free vertices need not be dominated by D, but may be included in D in order to dominate bound vertices. We define $\gamma_m(G)$, the mixed domination number of G, to be the order of a smallest mixed dominating set in G; such a set is also called a γ_m-set of vertices of G.

Let C be a cycle. The construction and correctness of Algorithm MCYCLE is based on the following theorem. The proof of this theorem is straightforward and is omitted.

Theorem 1. *Let C be a cycle having free, bound and required vertex sets F, B and R, respectively.*

(i) *Let x be a vertex of C which is labeled with R. Let P be the path formed by labeling each B-neighbor of x (if there are any) with F, and deleting x. Then $\gamma_m(C) = \gamma_m(P) + 1$.*

(ii) *Let x and y be adjacent vertices of C which are labeled with F. Let P be the path formed by deleting the edge (x, y) from C. Then $\gamma_m(C) = \gamma_m(P)$.*

(iii) *If all vertices of C are labeled with B, and if x is any vertex of C, and C' is the cycle obtained from C by relabeling x with R, then $\gamma_m(C) = \gamma_m(C')$.*

(iv) *Let the vertices of C be alternately labeled with $B, F, B, \ldots$, and let x be labeled with F. Let C' be the cycle obtained from C by relabeling x with R. Then $\gamma_m(C) = \gamma_m(C')$.*

(v) *Let x, y and z be three consecutive vertices of C labeled with B, B, and F, respectively. If D is a γ_m-set with $y \in D$, then $D - \{y\} \cup \{x\}$ is also a γ_m-set.*

Algorithm MCYCLE finds a γ_m-set D of a cycle or an isolated edge C, whose vertices are labeled with F, B, and R. This algorithm calls two subroutines; CREATEPATH(C, x, P) takes a cycle C and a vertex x (labeled with R) and relabels all B-neighbors of x with F, deletes x, and returns the resulting path P; DOMSET(T) is the Cockayne, Goodman, Hedetniemi [5] algorithm for finding a γ_m-set of a tree T.

Algorithm MCYCLE

```
if C is an edge
    then D←DOMSET(C)
    else case [C]:
        [there is a vertex x in C labeled R];
            call CREATEPATH(C, x, P)
            D←DOMSET(P)∪{x}
        [there are adjacent vertices x and y labeled F]:
            P←C−(x, y)
        D←DOMSET(P)
        [all vertices are labeled B]:
            label any vertex x with R
            call CREATEPATH(C, x, P)
            D←DOMSET(P)∪{x}
        [vertices are labeled alternately F, B, F, ...]:
            label any F vertex x with R
            call CREATEPATH(C, x, P)
            D←DOMSET(P)∪{x}
        [there are consecutive vertices x, y, z labeled B, B, F, resp.]:
            label vertex x with R
            call CREATEPATH(C, x, P_x)
            relabel vertex x with B, label vertex z with R
            call CREATEPATH(C, z, P_z)
            if |DOMSET(P_x)| ≤ |DOMSET(P_z)|
                then D←DOMSET(P_x)∪{x} [Note that x∈P_x]
                else D←DOMSET(P_z)∪{z} [Note that z∈P_z]
                fi
    fi;
stop
```

3. A linear algorithm for finding a γ-set in a cactus

Let K be a cactus. Let C be an endblock of K and let x be the unique cutvertex of K in C. Let C_F and C_R denote the block C with x relabeled with F and R, respectively. Note that $\gamma_m(C_F) \le \gamma_m(C) \le \gamma_m(C_R) < \gamma_m(C_F) + 1$. Let K'_F, K'_B, and K'_R

denote the cactus obtained from K by deleting all vertices only in C, and relabeling x with F, B, or R, respectively. With this notation we state the following theorem, the proof of which guarantees the correctness of Algorithm MCACTUS. Again, the proof of this theorem is straightforward. We include only the proof of case (v); the other cases have similar proofs.

Theorem 2. *Let K be a cactus having free, bound and required vertices F, B, and R, respectively, and let x be the unique cutvertex of the endblock C.*

(i) *If x is labeled with R, then $\gamma_m(K)=\gamma_m(K'_R)+\gamma_m(C)-1$.*

(ii) *If x is labeled with F and $\gamma_m(C)<\gamma_m(C_R)$, then $\gamma_m(K)=\gamma_m(C)+\gamma_m(K'_F)$.*

(iii) *If x is labeled with F and $\gamma_m(C)=\gamma_m(C_R)$, then $\gamma_m(K)=\gamma_m(K'_R)+\gamma_m(C_R)-1$.*

(iv) *If x is labeled with B and $\gamma_m(C_F)<\gamma_m(C)$, then $\gamma_m(K)=\gamma_m(K'_B)+\gamma_m(C_F)$.*

(v) *If x is labeled with B and $\gamma_m(C_F)=\gamma_m(C)<\gamma_m(C_R)$, then $\gamma_m(K)=\gamma_m(K'_F)+\gamma_m(C)$.*

(vi) *If x is labeled with B and $\gamma_m(C_F)=\gamma_m(C_R)$, then $\gamma_m(K)=\gamma_m(K'_R)+\gamma_m(C_R)-1$.*

Proof. (v) Let D be a γ_m-set of K.

Case 1: *x is not in D.* Then x is adjacent to some vertex y in D.

Subcase 1a: *y is in C.* Then $D\cap C$ is a mixed dominating set of C, and $D-C$ is a mixed dominating set of K'_F. Thus we have

$$\begin{aligned}\gamma_m(K)&=|D|=|D-C|+|D\cap C|\\&\geq\gamma_m(K'_F)+\gamma_m(C).\end{aligned}$$

Subcase 1b: *y is not in C.* Then since $D\cap C$ must contain a mixed dominating set of C_F, we let A be a γ_m-set of C_F, which is also a γ_m-set of C. Let $D'=(D-C)\cup A$. Since $|D'|=|D|$, we now have a γ_m-set D' of K such that x is dominated by a vertex in $D'\cup C$, as in the previous subcase. Thus, $\gamma_m(K)\geq\gamma_m(K'_F)+\gamma_m(C)$.

Case 2: *x is in D.* In this case, we know $|D\cup C|>\gamma_m(C)$. Let A be any γ_m-set of C, and let $D'=(D-C)\cup A\cup\{x\}$. Now $|D'|=|D|$, A is a mixed dominating set of C, and $D-A$ is a mixed dominating set of K'_F, so $\gamma_m(K)\geq\gamma_m(K'_F)+\gamma_m(C)$. Conversely, let D be a γ_m-set of K'_F, and let D' be a γ_m-set of C. $D\cup D'$ is clearly a mixed dominating set of K, so

$$\begin{aligned}\gamma_m(K)&\leq|D\cup D'|\leq|D|+|D'|\\&=\gamma_m(K')+\gamma_m(C).\end{aligned}$$ □

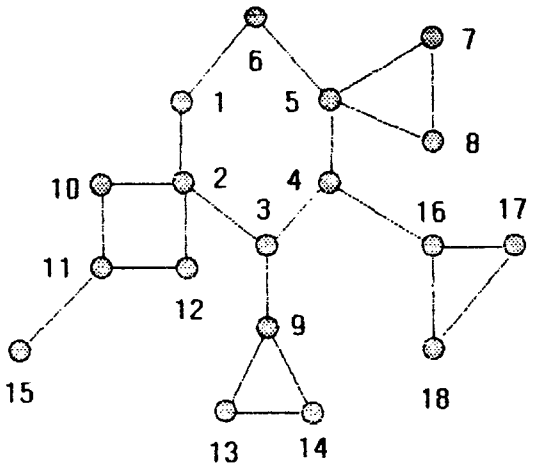
Algorithm MCACTUS finds a γ_m-set D of a cactus K with vertices labeled with F, B, or R. This algorithm calls MCYCLE as a subroutine.

Algorithm MCACTUS

```
D←∅
while K≠∅ do
    if K is a block
        then D←D∪MCYCLE(K); K←∅
        else let C be an endblock, x its unique cutvertex
            Case [x]:
            [labeled R]:
                D←D∪MCYCLE(C)
            [labeled F]:
                U←MCYCLE(C); V←MCYCLE(C_R)
                if |U|<|V|
                    then D←D∪U
                    else D←D∪V and label x with R
            [labeled B]:
                U←MCYCLE(C_F); V←MCYCLE(C); W←MCYCLE(C_R)
                if |U|<|V|
                    then D←D∪U
                    else [|U|=|V|]
                        if |V|<|W|
                            then D←D∪V
                                 label x with F
                            else D←D∪W
                                 label x with R
            end case
            K←(K−C)∪{x}
```

4. Complexity analysis

Let a cactus K have m blocks of order a_i, for i between 1 and m. K has p vertices, where $p=1+\sum_{i=1}^{m}(a_i-1)$. Since Algorithm DOMSET is linear, it is immediate that Algorithm MCYCLE is linear. Algorithm MCACTUS calls MCYCLE at most three times per block, so Algorithm MCACTUS is also linear. This algorithm has been implemented in FORTRAN in [12]. One of several different data structures that can be used to represent a cactus is illustrated in Fig. 1. By proceeding from right to left across the $2\times N$ array CACTUS we can easily recognize endblocks (e.g. 18–17–16; 16–4; 15–11; 14–13–9; 12–11–10–2; 9–3; 8–7–5; and 6–5–4–3–2–1). This representation can easily be achieved in linear time from a list of the edges of G using a standard $O(e)$ algorithm for finding the 2-connected components of a graph (cf. Algorithm 5.3, p. 185 in the textbook *The Design and Analysis of Computer Algorithms* by Aho, Hopcroft, and Ullman, Addison-Wesley, 1974). Since e is linear in p for cacti, Algorithm MCACTUS is $O(p)$.



1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
0	1	2	3	4	5	5	7	3	2	10	11	9	13	11	4	16	17
					1		5				2		9				16

Fig. 1. A cactus and its representation.

References

[1] W.W. Rouse Ball, Mathematical Recreations and Problems of Past and Present Times (MacMillan, London, 1892).
[2] C. Berge, Graphs and Hypergraphs (North-Holland, Amsterdam, 1973).
[3] K.S. Booth and J.H. Johnson, Dominating sets in chordal graphs, SIAM J. Comput. 11 (1) (1982) 191–199.
[4] E.J. Cockayne, Domination of undirected graphs: a survey, Theory and Applications of Graphs, Lecture Notes in Math. 642 (Springer, Berlin, 1978) 141–147.
[5] E.J. Cockayne, S. Goodman and S.T. Hedetniemi, A linear algorithm for the domination number of a tree, Inform. Process. Lett. 4 (1975) 41–44.
[6] E.J. Cockayne and S.T. Hedetniemi, Towards a theory of domination in graphs, Networks 7 (1977) 247–261.
[7] M. Farber, Domination, independent domination, and duality in strongly chordal graphs, Discrete Appl. Math. 7 (1984) 115–130.
[8] M.R. Garey and D.S. Johnson, Computers and Intractability – A Guide to the Theory of NP-completeness (W.H. Freeman, San Francisco, 1978).
[9] R. Laskar and H.B. Walikar, On domination related concepts in graph theory, Lecture Notes in Math. 885 (Springer, Berlin, 1980) 308–320.
[10] K.S. Natarajan and L.J. White, Optimum domination in weighted trees, Inform. Process. Lett. 7 (1978) 261–265.
[11] O. Ore, Theory of Graphs, Amer. Math. Soc. Colloq. Publ. 38 (AMS, Providence, RI, 1962).
[12] J.S. Pfaff, Algorithmic complexities of domination-related graph parameters, Ph.D. dissertation, Clemson University, May 1984.