Utilization of Cut Vertices in Finding the Domination Number of a Graph

Eleanor Hare
Department of Computer Science
Clemson University
Clemson, SC 29634-1906

1 Introduction

In this paper it is assumed that $G = (V, E)$ is a connected graph without self-loops or multiple edges. Let $v \in V$. The *closed neighborhood of v*, denoted $N[v]$, is the set $\{v\} \cup \{u \in V \mid uv \in E\}$. Let S be a subset of V. The *closed neighborhood of S*, denoted $N[S]$, is the set $\{N[v] \mid v \in S\}$. Let S be a subset of V. It is a *dominating set* for the graph G if and only if for every $u \in V$, there exists $v \in S$ such that $u \in N[v]$. A dominating set is *minimal* if for every $u \in S$, there exists $w \in N[u]$ such that $|N[w] \cap S| = 1$. The *domination number* of a graph G, denoted $\gamma(G)$, is the size of a smallest minimal dominating set.

Let S be a set of vertices in a graph G. Consider $N[S]$. If there exists a set of vertices, S', such that either

(1) $|S'| < |S|$ and $N[S'] \supseteq N[S]$ or

(2) $|S'| = |S|$ and $N[S'] \supset N[S]$,

then we say S is a *beatable dominating set*; otherwise, S is an *unbeatable dominating set.*

Examples are given in Figures 1 and 2. Observe that the relation "is beatable by" is irreflexive, asymmetric and transitive.

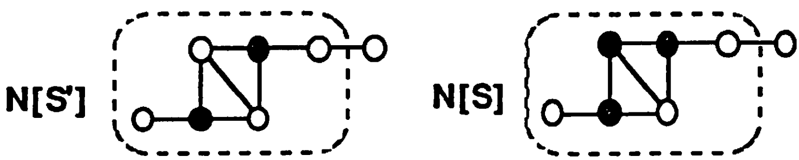

Figure 1. $|S'| < |S|$ and $N[S'] \supseteq N[S]$.
S is a beatable dominating set.

JCMCC 30 (1999), pp. 91-97

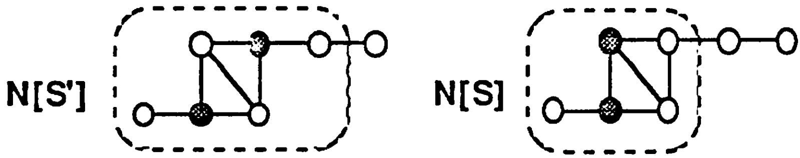

Figure 2. $|S'| = |S|$ and $N[S'] \supset N[S]$.
S is a beatable dominating set.

Theorem 1. *If S is beatable, then every set dominated by S is also dominated by a set S' having no more vertices than S.*

Proof:

Case 1. $N[S] = V$. Since S is beatable, there exists S' such that $|S'| < |S|$ and $N[S] = N[S'] = V$. Thus, $|S|$ is not the domination number of G, and G is dominated by a set, S', having fewer vertices.

Case 2. $V \supset N[S]$ and $|S'| < |S|$ and $N[S'] \supseteq N[S]$ for some set S'. By the definition, every vertex in $N[S]$ is dominated by S', using fewer vertices

Case 3. $V \supset N[S]$ and $|S'| = |S|$ and $N[S'] \supset N[S]$ for some set S'. By the definition, every vertex in $N[S]$ is dominated by S', using the same number of vertices. □

Beatable dominating sets were introduced in 1989 by Hare and Hedetniemi [4] as a tool for finding the domination number of arbitrary graphs. Hare and Fisher [3] have used beatable dominating set to decrease the time required to compute $\gamma(P_m \times P_n)$. Slater [6] has shown the problem of determining whether a set is beatable to be NP-complete.

Let v be a cut vertex, the removal of which partitions V into disjoint subgraphs, $C_1, C_2, \ldots, C_m$, called components. Let $X = C_i$, for $1 \le i \le m$.

Before proving Theorem 2, let us establish the existence of two cases. First, it is possible that $\gamma(X) \ge \gamma(\langle X \cup N[v] \rangle)$. In Figure 3 observe that removal of v separates G into components, C_i, $i \le 6$. In particular, C_1 is a path on four vertices, and $\gamma(P_4) = 2$, but $\gamma(\langle C_1 \cup N[v] \rangle) = 1$. Next, show that it is possible for $\gamma(X) = \gamma(\langle X \cup N[v] \rangle)$. Again in Figure 3, observe that $\gamma(C_4) = 2 = \gamma(\langle C_4 \cup N[v] \rangle)$. (Here $\langle X \rangle$ denotes the subgraph of G induced by X.)

Theorem 2. *If $\gamma(X) < \gamma(\langle X \cup N[v] \rangle)$, then v is in any smallest dominating set of G.*

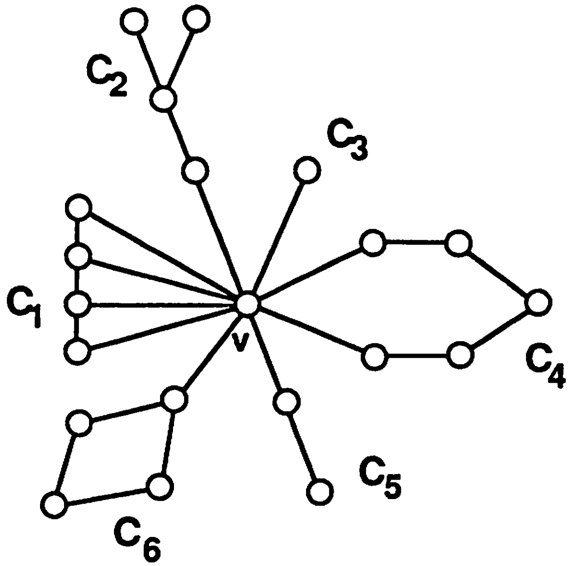

Figure 3. Example graph.

Proof: If $|S| = \gamma(X) > \gamma(\langle X \cup N[v] \rangle) = |S'|$, then S is beatable by S' and $v \in S'$. On the other hand, if $|S| = \gamma(X) = \gamma(\langle X \cup N[v] \rangle) = |S'|$, then S is beatable by S' and $v \in S'$. □

Theorem 3. *If $\gamma(X) < \gamma(\langle X \cup N[v] \rangle)$, then v is not in any minimum dominating set of G unless needed to dominate $G-X$. Also, v is dominated by $u \in X$ if and only if $\gamma(X) = \gamma(\langle X \cup \{v\} \rangle)$.*

Proof: The proof is subdivided into

Case 1. $|S| = \gamma(X) = \gamma(\langle X \cup \{v\} \rangle)$,

Case 2. $|S| = \gamma(X) < \gamma(\langle X \cup \{v\} \rangle)$, and

Case 3. $|S| = \gamma(X) > \gamma(\langle X \cup \{v\} \rangle)$.

Case 1. If $|S| = \gamma(X) = \gamma(\langle X \cup \{v\} \rangle) = |S'|$, then S is beatable by S', $v \notin S'$, and $v \in N[S']$.
Case 2. If $|S| = \gamma(X) < \gamma(\langle X \cup \{v\} \rangle) = |S'|$, then $v \notin S$ and $v \notin N[S]$. Observe that v is not dominated by S.
Case 3. If $|S| = \gamma(X) > \gamma(\langle X \cup \{v\} \rangle) = |S'|$, then $v \in S'$ and $\gamma(X) \geq \gamma(\langle X \cup N[v] \rangle)$, which contradicts the assumption that $\gamma(X) < \gamma(\langle X \cup N[v] \rangle)$. Thus, Case 3 is not possible. □

Algorithm 1 finds the domination number of a graph containing v.

1.0 Set v-dominated to FALSE.
    Set v-in-dom-set to FALSE.
    Set Count to zero.
2.0 For $i \leftarrow 1$ to m do
        If $\gamma(C_i \cup N[v]) \leq \gamma(C_i)$ then
            add $(\gamma(C_i \cup N[v]) - 1)$ to Count
            set v-in-dom-set to TRUE
        else
            Add $\gamma(C_i)$ to Count
            If $\gamma(C_i) = \gamma(C_i \cup \{v\})$ then
                set v-dominated to TRUE.
3.0 If v-in-dom-set then add 1 to Count
        else if NOT v-dominated add 1 to Count.
4.0 $\gamma(G) \leftarrow$ Count.

Algorithm 1.
Computes the domination number of a graph G containing a cut vertex v.

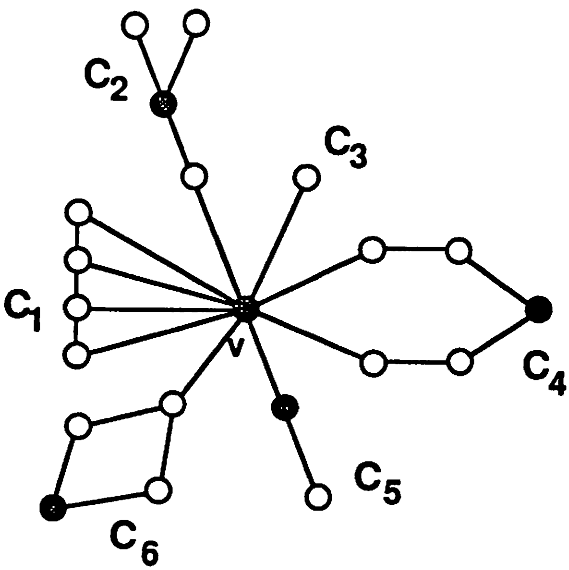

Figure 4. Domination of example graph.

Algorithm 1 may be extended to utilize more than one cut vertex. Suppose $U = \{u_1, u_2, \ldots, u_n)$ is a collection of cut vertices of G.

Algorithm 2 orders the vertices of U in time at most $O(n^2)$.

Let u_n be a distinguished vertex of $G - U$.
For $i \leftarrow m$ down to 2 do
    Find u_j, $j \le i$, such that removal of u_j partitions V into
        components, $C_{j_1}, C_{j_2}, \ldots, C_{j_x}$, with at most one
        component containing any vertices of $\{u_r\} \cup \{u_1, u_2, \ldots, u_i\}$
    Interchange u_i and u_j.
od

Algorithm 2. Ordering the set $U = \{u_1, u_2, \ldots, u_m\}$.

Without loss of generality, let C_{i_1}, $1 \le i \le m$, denote the component containing the distinguished vertex, u_r. Algorithm 3 assumes that the vertices of U have been so ordered.

A block is a non-trivial connected graph with no cut vertices. If a graph has any cut vertices, there is an underlying tree structure of the blocks. Algorithm 2 orders the cut vertices of a graph so that Algorithm 3 can process the blocks in a manner analogous to processing the leaves of a tree and pruning each leaf until the root is processed.

1.0 Set Count to zero.
1.1 Copy G to G_m.
2.0 for $i \leftarrow m$ downto 1 do
2.1   Set u-in-dom-set to FALSE.
2.2   If $u_i \in N[w]$ where w has been marked "dominating"
            then set u-dominated to TRUE
            else set u-dominated to FALSE.
2.3   For $k \leftarrow i_2$ to i_x do
            If $\gamma(\langle C_k \cup N[u_i]\rangle) \le \gamma(C_k)$ then
                Add $(\gamma(\langle C_k \cup N[u_i]\rangle) - 1)$ to Count
                Set u-in-dom-set to TRUE
            else
                Add $\gamma(C_k)$ to Count
                If $\gamma(C_k) = \gamma(\langle C_k \cup \{u_i\}\rangle)$ then
                    set u-dominated to TRUE
        end for
2.4   If u-in-dom-set then mark u_i "in-dom-set"
        else if u-dominated then mark u_i "dominated"
2.5   Set G_{i-1} to $\langle C_{i_1} \cup \{u_i\}\rangle$
3.0 $\gamma(G) \leftarrow$ Count $+\gamma(G_0)$.

Algorithm 3.
Finding the domination number of an arbitrary graph containing cut vertices.

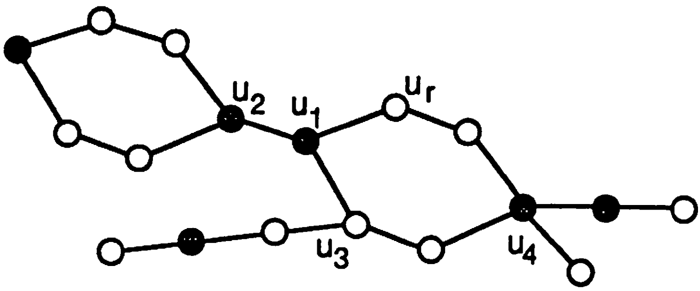

Figure 5.
Example of minimum dominating set selected by Algorithm 3

2 Discussion

Since the problem of finding a minimum dominating set for a graph G is known to be NP-complete [1], finding minimum dominating sets of subsets instead of the entire graph may be an improvement.

Algorithm 3 is especially useful for any graph with blocks for which the domination number can be easily calculated. For example, if the blocks are cycles, calculation of the domination number is straight-forward. In any case, Algorithm 3 allows the domination number of components to be calculated, rather than the domination number of the graph as a whole. The complexity of Algorithm 3 is $O(s \cdot t)$ where s is the number of cut vertices and $t = \max(O(f(c_i))$, where $f(C_i)$ is a function of the structure of the component C_i.

The table-driven algorithm of Goodman, Hetetniemi, and Cockayne [2] and a similiar algorithm used by Wimer [7] in developing a methodology for k-terminal families of graphs are among the linear algorithms for the domination of trees found in the literature. In the case of a tree, all interior vertices may be selected in the set of cut vertices, giving complexity $O(n)$. Thus, Algorithm 3 yields a linear algorithm for trees. (A conveniently ordered data structure, such as a Parent Array [5], should be used to avoid the complexity of Algorithm 2.)

It is expected that Algorithm 3 can be modified to calculate other parameters for which beatable sets can be defined, such as k-packing, total domination, distance-k-domination, and k-weight-domination.

References

[1] M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W.H. Freeman and Co., New York, 1979.

[2] S.E. Goodman, S.T. Hedetniemi and E.J. Cockayne, A linear algorithm for the domination number of a tree, *Information Processing Letters* 4 (1975), 41–44.

[3] E.O. Hare and D.C. Fisher, An application of beatable dominating sets to algorithms for complete grid graphs, *Graph Theory, Combinatorics, and Applications: Proc. Seventh Quadrennial International Conference on the Theory and Applications of Graphs*, Volume 1 (Edited by Y. Alavi and A. Schwenk), (1995) John Wiley and Sons, Inc., 497–506.

[4] E.O. Hare and S.T. Hedetniemi, Beatable sets, presented at the 20th Southeastern International Conference on Combinatorics, Graph Theory and Computing, Feb. 20-24, 1989, Boca Raton, Florida.

[5] D. Knuth, *The Art of Computer Programming*, Vol 1, Addison-Wesley, Reading, Mass., 1968, (p. 390).

[6] P.J. Slater, private communication, 1989.

[7] T. Wimer, Linear algorithms on k-terminal graphs, Ph.D. Thesis, Dept. Computer Sci., Clemson University, 1987.