Perfect Domination

Michael R. Fellows
Department of Computer Science
University of Victoria
Victoria, B.C. V8W 2Y2, Canada

Mark N. Hoover
Educational Testing Service
Princeton, N.J. 08541, U.S.A.

Abstract

The complexity of decision problems concerning the existence of subsets $V' \subseteq V$ of graphs $G = (V, E)$ with domination properties that involve conditions on the number of neighbors of a vertex that belong to V' is studied. Several such problems are shown to be NP-complete, even for input restricted to planar graphs of maximum degree 3. In particular, an independent set V' with the property that each vertex in $V - V'$ has a unique neighbor in V' is termed a *perfect dominating set*. It is shown that determining whether a tree has a perfect dominating set can be solved in $O(\log |V|)$ time with $O(|V|)$ processors in the CREW PRAM model of parallel computation and in $O(|V|)$ time sequentially.

1. Introduction

A *dominating set* in a graph $G = (V, E)$ is a subset V' of the vertex set V having the property that every vertex of G either belongs to V' or has a neighbor that belongs to V'. Applications of concepts of domination arise in situations where every vertex in the graph which models the problem must have "at least one" neighbor in the distinguished subset V' of the vertex set V of the graph. We study here variations which arise where at least one neighbor in V' is necessary, but more than one is too much! All graphs are simple, without loops or multiple edges. The following describes one such variation.

Perfect Domination
Instance: A graph $G = (V, E)$.
Question: Is there a subset $V' \subseteq V$ which satisfies the conditions: (i) V' is a dominating set, and (ii) $N[u] \cap N[v] = \emptyset$ for all $u, v \in V', u \neq v$, where $N(x) = \{y | y$ is adjacent to $x\}$ and $N[x] = \{x\} \cup N(x)$.

A perfect dominating set in the graph Q_q (the q-dimensional binary cube) is precisely a perfect single error-correcting code. The graph Q_q has a perfect dominating set if and only if $q = 2^s - 1$.

Perfect dominating sets, also called *perfect codes* in graphs, have been previously studied by a number of authors [Bi,DR,HS,KK,KMM,Kr1,Kr2,Kr3,Kr4,Sm]. Our

Australasian Journal of Combinatorics 3(1991) pp 141-150

focus in this paper is on issues of computational complexity. Note that replacing the requirement "at least one neighbor in V'" by "exactly one neighbor in V'" allows the interesting question to be asked (for a variety of properties of V') whether such a set V' exists at all in the graph. Our main result in this paper is that many such decision problems for domination properties are *NP*-complete, even for planar graphs, but can be answered in linear sequential time and in $O(\log |V|)$ time using $O(|V|)$ CREW processors for trees.

2. Perfect Domination

Theorem 1 shows that *Perfect Domination* is *NP*-complete for planar graphs of maximum degree three. A stronger result has been obtained independently by [KK]. We include a proof here as it forms the basis of some of our subsequent theorems. We give only the main points of the argument.

Theorem 1. *Perfect Domination* is *NP*-complete for planar graphs of maximum degree three.

Proof. The problem is clearly in NP. We show that it is NP-hard by reducing from the *NP*-complete problem *Planar Three-Dimensional Matching* [GJ,DF]. An instance of this problem consists of three disjoint sets R, B, Y, of equal cardinality q and a set T of triples, $T \subseteq R \times B \times Y$. We assume without loss of generality that each element of $R \cup B \cup Y$ belongs to at least one triple in T. To each instance I we may associate a bipartite graph G_I. The vertex classes of G_I are $R \cup B \cup Y$ and T. An edge connects a triple $t \in T$ to an element $x \in R \cup B \cup Y$ if and only if x is a member of t. The question is to decide if there is a subset of q triples of T that contains all the elements of R, B and Y. In [DF] this problem is shown to be NP-complete even when restricted to instances I for which the associated graph G_I is planar.

To reduce an instance I of *Planar Three-Dimensional Matching* to *Perfect Domination* we modify the associated graph G_I by introducing a third set of vertices T' in one–to–one correspondence with the set of vertices T together with edges joining each vertex $t \in T$ to the corresponding vertex $t' \in T'$. Thus each vertex of T' of the modified graph G'_I has degree one.

If $P \subseteq T$ is a solution set of q triples for the instance I of *Three-Dimensional Matching* then $P \cup \{t' | t \in T - P\}$ is a perfect dominating set in G'_I. Conversely, if G'_I has a perfect dominating set D then no vertex x of $R \cup B \cup Y$ belongs to D. To see this, note that there is at least one triple t to which x belongs and $x \in D$ implies $t \notin D$. If $t' \notin D$ then t' has no neighbor in D, while if $t' \in D$ then t has two neighbors in D. In either case this contradicts that D is perfect.

A perfect dominating set D in G'_I that consists entirely of vertices in $T \cup T'$ corresponds to a solution set for I, since each element of $R \cup B \cup Y$ has exactly one neighbor (triple in which it occurs) in $D \cap T$. The graphs G_I and G'_I can clearly be computed from I in polynomial time.

Figure 1 shows an example (for a vertex of degree 5) of a transformation of a planar graph H to a planar graph H' which has maximum degree three with the property that H' has a perfect dominating set if and only if H has one. This transformation is easily accomplished in polynomial time. □

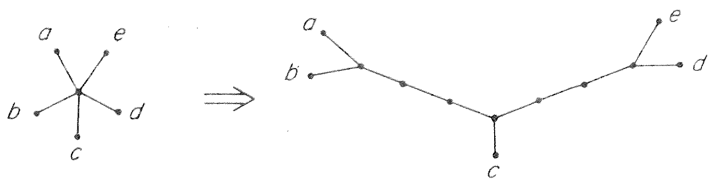

Figure 1. Transformation to maximum degree three.

3. Semiperfect Domination

The conditions which define a perfect dominating set are rather stringent; not only do we require that vertices in $V - V'$ have exactly one neighbor in V', but we also require that no two vertices of V' be adjacent. Relaxing this second requirement motivates the following definition, which would be applicable to problems of "interference-free" servicing or broadcasting, where the service centers or transmitters can be adjacent.

Definition. A *semiperfect dominating set* of a graph $G = (V, E)$ is subset V' of the vertex set $V' \subseteq V$ such that

(*i*) V' is a dominating set.

(*ii*) For $v \in V - V', |N(v) \cap V'| = 1$.

Since every graph $G = (V, E)$ has a semiperfect dominating set (by taking $V' = V$) it is natural to ask whether it has one of size less than or equal to k.

Minimal Semiperfect Domination
Instance: A graph $G = (V, E)$ and an integer k.
Question: Does G have a semiperfect dominating set $V' \subseteq V$, such that $|V'| \leq k$?

As a special case, we may ask whether G has a *nontrivial* semiperfect dominating set, i.e. one with $V' \neq V$. *Minimal Semiperfect Domination* is *NP*-complete as a corollary to the *NP*-completeness of the following problem, which is the special case of $k = |V| - 1$.

Semiperfect Domination
Instance: A graph G.
Question: Does G have a nontrivial semiperfect dominating set?

Theorem 2. *Semiperfect Domination* is *NP*-complete for planar graphs.

Proof. The problem is clearly in *NP*. We reduce from *Perfect Domination* for connected, planar graphs of maximum degree three. We may assume without loss of generality that no vertex of degree 2 in G has two distinct neighbors of degree 2, since otherwise G can be simplified by replacing a path of length 4 with all internal vertices of degree 2 by a single edge, with the resulting graph having a perfect dominating set if and only if G has one.

Under this assumption, let G_0 be obtained from G as follows:
1. Replace each edge x_0x_1 of G where $deg(x_0) = deg(x_1) = 3$ by a path of length 7.
2. Replace each path $x_0x_1x_2$ where $deg(x_0) = deg(x_2) = 3$ and $deg(x_1) = 2$ by a path of length 5.
3. Replace each path $x_0x_1x_2x_3$ where $deg(x_0) = deg(x_3) = 3$ and $deg(x_1) = deg(x_2) = 2$ by a path of length 9.

The result of the above modifications is to produce a graph G_0 that has a perfect dominating set if and only if G has one, and which is isomorphic to a subdivision of a cubic graph C with each edge e of C subdivided s_e time where $s_e \geq 4$ and $s_e \equiv 0 \pmod 2$.

Next, for all vertices u of degree 2 in G_0 create a new vertex u' and attach it to u and to both neighbors of u in G_0. The construction of G_0 insures that the u'_i of figure 2(c) can have a consistent orientation around the trivalent vertices v, thus assuring the planarity of G', the corresponding subgraph of which is shown in figure 2(d).

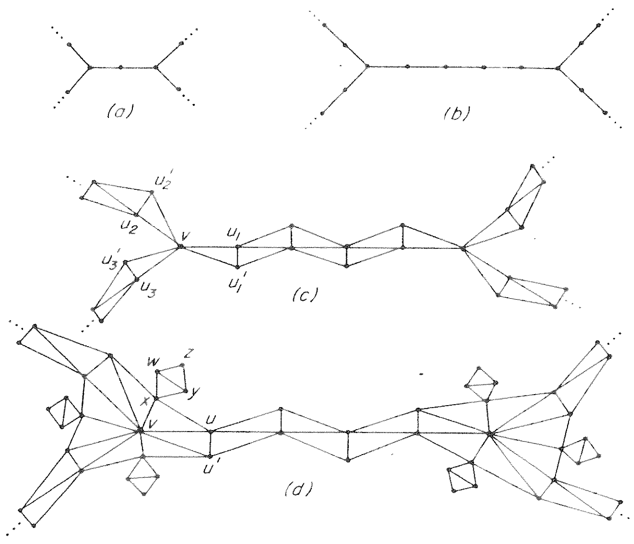

Figure 2. (a) G. (b) G_0. (c) Degree two modification. (d) G'.

Claim 1. If G_0 has a perfect dominating set P then G' has a nontrivial semiperfect dominating set S.

Clearly P dominates all of G' except the vertices labeled y, w, z and possibly x in figure 2(d). Let S be the union of P with appropriate vertices from the set $\{w, x, y, z\}$. If x has a neighbor in P then include z in S, otherwise include y. S is in fact a perfect dominating set of G'.

Claim 2. Any nontrivial semiperfect dominating set S of G' does not contain adjacent vertices.

Note that if two vertices of a 3-cycle are in S then the third vertex is in S also. If $x \in S$ then $\{y, w, z\} \subseteq S$. If S is nontrivial and contains an adjacent pair of vertices, these must belong to a set $\{x, y, w, z\}$ as in figure 3(d). If two of these are in S then all must be. In particular, x must be in S. But then no vertex of $N[u']$ can be in S, a contradiction.

Since any semiperfect dominating set with no adjacent vertices is a perfect dominating set, Claim 1 and Claim 2 together conclude the proof. □

4. Weakly Perfect Domination

In *Semiperfect Domination* we require that each vertex not in V' have exactly one neighbor in V', with no conditions on adjacencies between vertices in V'. If we require in addition that each vertex in V' be adjacent to at most one other vertex in V', then we have the following problem, equivalently obtained by replacing closed neighborhoods by open neighborhoods in the definition of a perfect dominating set.

Weakly Perfect Domination
Instance: A graph $G = (V, E)$.
Question: Is there a subset $V' \subseteq V$ such that: (i) V' is a dominating set, and (ii) for all $u, v \in V'$, $u \neq v, N(u) \cap N(v) = \emptyset$?

Theorem 3. *Weakly Perfect Domination* is *NP*-complete for planar graphs.

Proof. The problem is clearly in *NP*. The reduction is from *Perfect Domination* for connected, planar graphs of maximum degree three. For such a graph G let G_0 be constructed as in the previous proof. For each $u \in G_0$ of degree two create a vertex u' as before and attach it to u and to both neighbors of u in G_0. Denote this new graph G'.

Any perfect dominating set of G_0 is a perfect dominating set of G' and thus a weakly perfect dominating set of G'. To show that a weakly perfect dominating set of G' is a perfect dominating set of G_0 we argue the following claims.

Claim 1. Any weakly perfect dominating set W of G' does not contain any vertex u' created in the construction of G'.

Assume $u' \in W$, u adjacent to v, v adjacent to $w \neq u$, $w \neq u'$ and u, v, w of degree 2 in G_0. That such a v and w exist for every u of detree 2 is assured, since in the construction of G_0 the vertices of degree 2 induce a subgraph that consists of a disjoint collection of paths, each of length at least 5. Since $N(u') \cap N(u) \cap N(v') \cap N(w) = \{v\}$ none of u, v' or w can be in W. Since $N(u') \cap N(v) = \{u\}$, v cannot be in W. Then v' is not dominated.

Claim 2. Any weakly perfect dominating set W of G' does not contain adjacent vertices.

W contains no pair of adjacent vertices that includes a vertex u' created in the construction of G', by Claim 1. Suppose $u, v \in V(G_0)$ are adjacent. Without loss of generality we may asume u has degree 2 in G_0. Since $u' \in N(u) \cap N(v)$, either $u \notin W$ or $v \notin V$.

Any weakly perfect dominating set in G' with no adjacent vertices is a perfect dominating set in G' and by Claim 1, a perfect dominating set in G_0. Since G_0 has a perfect dominating set if and only if G has a perfect dominating set, the proof is concluded. □

5. Perfect Domination in Trees

We now show that we can determine if a tree has a perfect dominating set in time linear in the number of vertices of the tree. Our result holds for unrooted trees, although the algorithm assumes that the tree has a root. For an unrooted tree, a root may be chosen arbitrarily.

Consider a tree $T = (V, E)$ with root vertex r. With respect to a perfect dominating P, each vertex v of T must satisfy exactly one of the following statements:

(*above*)	The parent of v is in P.
(*in*)	The vertex v is in P.
(*below*)	Some child of v is in P.

Note that a leaf must satisfy statement *in* or statement *above* and that if two leaves are siblings in T then their parent must be in P. By evaluating the tree beginning with the leaves, we can determine if T has a perfect dominating set. The algorithm given may be viewed as a special case of the elegant and general methods of [BLW,Wi] for linear algorithms in trees and other families of graphs. For a constructive characterization of trees which have a perfect dominating set see [BBS]. The linear time algorithm presented there is essentially the same, but to clarify our parallel algorithm for the problem we provide the following sketch.

To simplify notation and details of the algorithm, consider the following transformation (as indicated in figure 3) of T to a tree T' in which every vertex has at most two children. The tree T' has a perfect dominating set if and only if T has a perfect dominating set. The number of vertices in T' is bounded by $4\ |V|$, and the transformation can be made in sequential time linear in the size of T. The transformation can also be accomplished in time $O(\log |V|)$ with a linear number of processors by repeatedly "splitting" the task at each vertex with more than 2 children.

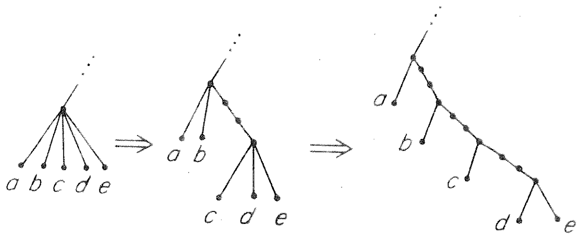

Figure 3. Transformation to at most two children.

We may assume henceforth that T is a tree in which every vertex has at most two children. Associate with each vertex v of T a boolean vector, $value(v) = (a, b, c)$, where a, b or c is true if and only if, based only on examination of the subtree rooted at v, some perfect dominating set of T exists in which v satisfies statement *above, in* or *below* respectively. Thus *value* (*leaf*) is initialized to (1,1,0), reflecting the fact that a leaf can not be dominated from below. For an internal vertex v, *value* (v) is to be computed in terms of the values of its children. The function to be computed at each internal vertex v of T can be expressed as follows:

If v has one child then

(1) $\quad value(v) = (z, x, y)$

where (x, y, z) is the value of the child of v.

If v has two children then

(2) $\quad value(v) = (z_1 z_2, x_1 x_2, z_1 y_2 + y_1 z_2)$

where (x_1, y_1, z_1) and (x_2, y_2, z_2) are the values of the children of v.

For example, v satisfies statement *above* if all of its children satisfy statement *below*; v satisfies statement *in* if all of its children satisfy statement *above*; v satisfies statement *below* if some child satisfies statement *in* and all other children (if any) satisfy statement *below*.

Since the root r cannot be dominated from above, $value(r)(0, 1, 1)^T \neq (0, 0, 0)$ if and only if T has a perfect dominating set. That the above algorithm correctly determines whether T has a perfect dominating set is easily shown by induction on the number of vertices of T. That it is linear is obvious.

We next show that *Perfect Domination* can be solved in $O(\log |V|)$ time with $O(|V|)$ CREW processors. The basic approach is due to the work of [MR] on evaluation of expression trees while the observations needed to carry out the approach on a problem of determining a graph property such as *Perfect Domination* are due to [MP]. A naive parallelization of the previous algorithm by associating with each vertex v of T a processor to compute $value(v)$ is $O(height)$. The obvious shortcoming is that it fails to deal adequately with long paths in the tree. By composing the functions associated with certain parent-child pairs we can sufficiently collapse long paths of partially evaluated vertices to obtain an $O(\log |V|)$ time algorithm.

In place of (1) and (2) consider the following computation rules :

If v has one child then

$$(3) \qquad value(v) = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} (x, y, z)^T$$

If v has two children then

$$(4) \qquad value(v) = \begin{pmatrix} 0 & 0 & z_1 \\ x_1 & 0 & 0 \\ 0 & z_1 & y_1 \end{pmatrix} (x_2, y_2, z_2)^T$$

(Since the function is commutative, which child is $child_1$ and which is $child_2$ can be decided arbitrarily.) The significant difference is in (4). Notice that (2) is in no apparent way associative. The rule (4) divides the function evaluation into two tasks; firstly, the creation of the three by three matrix, and secondly, matrix multiplication. We exploit the fact that matrix multiplication is associative. If we have a parent-child pair for which some other child of the parent is evaluated then the three by three matrix for the parent can be created, leaving only the matrix multipication to be performed. For example, consider the tree shown in figure 4, where (a, b, c) and (d, e, f) are known and (x, y, z) is unknown.

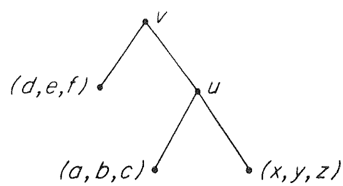

Figure 4.

Here

$$value(u) = \begin{pmatrix} 0 & 0 & c \\ a & 0 & 0 \\ 0 & c & b \end{pmatrix} (x, y, z)^T$$

and

$$value(v) = \begin{pmatrix} 0 & 0 & f \\ d & 0 & 0 \\ 0 & f & e \end{pmatrix} (value(v))^T$$

so,

$$value(v) = \begin{pmatrix} 0 & 0 & f \\ d & 0 & 0 \\ 0 & f & e \end{pmatrix} \begin{pmatrix} 0 & 0 & c \\ a & 0 & 0 \\ 0 & c & b \end{pmatrix} (x,y,z)^T = \begin{pmatrix} 0 & fc & fb \\ 0 & 0 & dc \\ fa & ec & eb \end{pmatrix} (x,y,z)^T$$

We can associate with v a known matrix, $\begin{pmatrix} 0 & fc & fb \\ 0 & 0 & dc \\ fa & ec & eb \end{pmatrix}$, and delete u from the computation, thus "shortening" the tree. By the techniques of [MR] the entire tree can be evaluated in time $O(\log |V|)$.

6. Summary

We have addressed in this paper some formulations of "domination" where classes of vertices are permitted or required to have at most one or exactly one neighbor in the dominating set. In perfect domination every vertex must be uniquely dominated, in semiperfect domination every vertex not in the dominating set must be uniquely dominated, while a weakly perfect dominating set is both semiperfect and satisfies the condition that a vertex not in the dominating set has at most one neighbor in the dominating set.

We have shown that determining whether an arbitrary graph G has a perfect dominating set is NP-complete for planar graphs of maximum degree three. Determining whether G has a nontrivial semiperfect dominating set is also NP-complete for planar graphs, as is determining whether G has a weakly perfect dominating set. We have also shown how the problem *Perfect Domination* can be solved in $O(\log |V|)$ time with a linear number of processors for input restricted to trees.

Acknowledgements. We would like to thank Gary Miller and Andre Proskurowski for helpful comments regarding this work.

References

[BBS] D. W. Bange, A. E. Barkauskas and P. J. Slater, "Efficient Dominating Sets in Graphs," to appear.

[Bi] N. Biggs "Perfect Codes in Graphs," *J. Combin. Theory, Ser.* B 15 (1973), 289–296.

[BLW] M. W. Bern, E. L. Lawler and A. L. Wong, "Why Certain Subraph Computations Require Only Linear Time," *Proceedings of the 26th Annual IEEE Symposium on the Foundations of Computer Science,* 1985.

[DF] M. E. Dyer and A. M. Frieze. "Planar 3DM is NP-Complete," *Journal of Algorithms*, vol. 7, 1983, p. 174–184.

[DR] I. Dvořáková- Rulićová, "Perfect Codes in Regular Graphs," *Comment. Math. Univ. Carolin.* 29 (1988), 79–83.

[GJ] M. Garey and D. Johnson, Computers and Intractability, W. H. Freeman and Company, 1979.

[HS] P. Hammond and D. H. Smith, "Perfect Codes in the Graphs O_k," *Journal of combinatorial Theory, Ser.* B 19 (1975), 239–255.

[KK] J. Kratochvíl and M. Křivánek, "On the Computational Complexity of Codes in Graphs," Technical Report no. 87-70, Dept. of Applied Mathematics, Charles University, Prague, 1987.

[KMM] J. Kratochvíl, J. Malý and J. Matoušek, "On Perfect Codes in a Random Graph," Technical Report no. 87- 44, Dept. of Applied Mathematics, Charles University, Prague, 1987.

[Kr1] J. Kratochvíl, "Perfect Codes Over Graphs," *J. Combin. Theory, Ser.* B 40 (1986) 224–228.

[Kr2] J. Kratochvíl, "One-Perfect Codes Over Self- Complementary Graphs," *Comment. Math. Univ. Carolin.* 26 (1985), 589–595.

[Kr3] J. Kratochvíl, "Perfect Codes in Graphs and Their Powers," Ph.D. Dissertation (in Czech), Charles University, Prague, 1987.

[Kr4] J. Kratochvíl, "Perfect Codes in General Graphs," Technical Report no. 87-65, Dept. of Applied Mathematics, Charles University, Prague, 1987.

[MP] G. Miller and A. Proskurowski, private communication, 1988. [MR] G. Miller and J. Reif, "Parallel Tree Contraction and its Application," *Proceedings of the 26th Annual IEEE Symposium on the Foundations of Computer Science,* 1985.

[Sm] D.H. Smith, "Perfect Codes in the Graphs O_k and $L(O_k)$," *Glasgow Math. J.* 21 (1980), 169–172.

[Wi] T.V. Wimer, "Linear Algorithms on k-Terminal Graphs," Ph.D. Dissertation, Clemson University, 1987.