In *Congressus Numerantium* **79** (1990), pp. 187-203.

PERFECT DOMINATING SETS

Marilynn Livingston*
Department of Computer Science
Southern Illinois University
Edwardsville, IL 62026-1653

Quentin F. Stout†
Elec. Eng. and Computer Science
University of Michigan
Ann Arbor, MI 48109-2122

Abstract

A dominating set S of a graph G is *perfect* if each vertex of G is dominated by exactly one vertex in S. We study the existence and construction of PDSs in families of graphs arising from the interconnection networks of parallel computers. These include trees, dags, series-parallel graphs, meshes, tori, hypercubes, cube-connected cycles, cube-connected paths, and de Bruijn graphs. For trees, dags, and series-parallel graphs we give linear time algorithms that determine if a PDS exists, and generate a PDS when one does. For 2- and 3-dimensional meshes, 2-dimensional tori, hypercubes, and cube-connected paths we completely characterize which graphs have a PDS, and the structure of all PDSs. For higher dimensional meshes and tori, cube-connected cycles, and de Bruijn graphs, we show the existence of a PDS in infinitely many cases, but our characterization is not complete. Our results include distance d-domination for arbitrary d.

1 Introduction

Suppose $G = (V, E)$ is a graph with vertex set V and edge set E. A vertex i is said to *dominate* a vertex j if E contains an edge from i to j or if $i = j$. A set of vertices $S \subseteq V$ is called a *dominating set* of G if every vertex of G is dominated by at least one member of S. When each vertex of G is dominated by exactly one element of S, the set S is called a *perfect dominating set* (PDS) of G.

The size of a set of least cardinality among all dominating sets for G is called the *domination number* of G and any dominating set of this cardinality is called a *minimum dominating set* for G. It is clear that a perfect dominating set for a graph is necessarily a minimum dominating set for it as well. These notions can be extended to d-domination, where a vertex i is said to *d-dominate* a vertex j if there is a path from i to j in G of length at most d. When $d > 1$, we will use the terminology *distance d PDS* or *perfect d-dominating set* to describe a subset S of vertices of a graph G such that every vertex of G is d-dominated by a unique vertex in S.

The concept of a perfect d-dominating set seems to have appeared first in a paper by Biggs [5], who introduced the term *perfect d-code* to denote what we call a perfect d-dominating set. Biggs [5, 6] was concerned with characterizing all perfect d-error correcting codes, and claimed that the proper setting to study such issues is in the class of distance-transitive graphs. He defined a *distance-transitive* graph G as a connected graph with distance function δ such that whenever u, v, x, y, are vertices of G for which $\delta(u, v) = \delta(x, y)$, there is an automorphism σ of G such that $\sigma(u) = x$ and $\sigma(v) = y$. Using algebraic techniques, Biggs derived an important necessary condition for the existence of a perfect d-code in a distance transitive graph and applied his condition to several distance-transitive graphs including the classical hypercube graph.

*Partially supported by National Science Foundation grant CCR-8808839

†Partially supported by National Science Foundation grant DCR-8507851 and an Incentives for Excellence Award from Digital Equipment Corporation

Later Bange, Barkauskas, and Slater [1], apparently unaware of Bigg's work, defined the class of *efficient* dominating sets, which is exactly the same as the class of perfect 1-dominating sets. They concentrated on finding perfect dominating sets in trees, showing that there are linear-time algorithms that decide if a tree has a PDS, and if so then produce one [2].

Our motivation for studying the notion of perfect domination in graphs arose from our work involving resource allocation and placement in parallel computers [15]. To see how PDSs arise in this context, suppose we have a parallel computer whose processors (pes) and interconnection network are modelled by the graph $G = (V, E)$, where each pe is associated with a vertex of G and a direct communication link between two pes is indicated by the existence of an edge between the associated vertices. Suppose, further, that we have a limited resource such as disks, I/O connections, or software modules, and we want to place a minimum number of these resource units at the pes, with at most one per pe, and so that every pe is within a distance d of at least one resource unit. Finding such a placement involves constructing a minimum d-dominating set for the graph G. If G has a perfect d-dominating set, this represents an optimal situation in which there is neither duplication nor overlap. Even when a distance d PDS does not exist for a given graph, information about perfect dominating sets for related graphs can be useful to help construct near optimum d-dominating sets. This is particularly true in graphs with quite regular structures, which are the graphs that arise from parallel computers.

Determining if an arbitrary graph has a dominating set of a given size is a well-known NP-complete problem [8, 13]. Straightforward proofs can be used to show that it is also NP-complete to decide if a graph has a PDS, and the problem remains NP-complete even if the graphs are restricted to 3-regular planar graphs. Thus the general problem of determining if a graph has a PDS is quite hard, but we show that for many significant classes of graphs it is manageable.

In Section 2 we consider whether perfect dominating sets exist for several classes of graphs which arise in the context of networks for parallel computers. These families include meshes, tori, trees, dags, series-parallel graphs, hypercubes, cube-connected cycles, cube-connected paths, and de Bruijn graphs. Except for hypercube graphs, none of these is distance transitive, and therefore the Biggs condition does not apply.

The existence of perfect d-dominating sets for trees, dags, and series-parallel graphs is studied in Section 2.1. For fixed $d \geq 1$, we give linear time algorithms that determine if a perfect d-dominating set exists, and generate them when they do exist. In Section 2.2, we completely characterize all 2- and 3-dimensional meshes and 2-dimensional tori that possess a PDS and also characterize the structure of the existing PDSs. For 2-dimensional meshes and tori we extend this to distance d PDSs for arbitrary d.

Distance d perfect dominating sets for hypercubes and hypercube related networks such as cube-connected cycles and cube-connected paths are considered in Section 2.3. For completeness we include the complete characterization of distance d PDSs for hypercubes, which follows from the results on perfect d-error correcting codes. Our characterization of the cube-connected cycles which have a PDS and the structure of the existing PDSs is not complete, however. For, while we have shown that there are infinitely many dimensions k for which cube-connected cycles of dimension k have a PDS, and that no PDS exists for dimensions 2 and 5, we do not know if there are infinitely many dimensions for which a PDS does not exist. The situation for cube-connected paths is much simpler. In Section 2.3.3, we show no PDS can exist for cube-connected paths of even dimension, while for cube-connected paths of odd dimension, we construct the PDSs, which are unique up to isomorphism.

Graphs which are constructed from binary shift register sequences, called de Bruijn graphs, are considered in Section 2.4. For directed de Bruijn graphs, which we denote by B_k, we construct distance 1 PDSs for all k and show the existence of distance d PDSs for infinitely many values of k. Undirected de Bruijn graphs, denoted by B_k^*, have a PDS for $k = 1, 2$, but we show that no PDS exists for $k = 3, 4, 5$. Whether distance d PDSs exist for B_k^* when $k > 5$ and $k-1 > d \geq 1$ is not known.

2 Perfect Dominating Sets for Graphs

Let $N_d(G, v)$ denote the set of vertices in the graph $G = (V, E)$ within a distance d of vertex v and let $n_d(G, v) = |N_d(G, v)|$. If S is a perfect d-dominating set for G then $\{N_d(G, v) : v \in V\}$ forms a partition of V and

$$\sum_{v \in S} n_d(G, v) = |V|. \tag{1}$$

When G is regular or nearly regular, Equation (1) can be simplified, which provides a useful tool in combinatorial arguments for the existence of a PDS of G. We note also that if d is at least the size of the radius of G, then G has a distance d PDS.

As we investigate the existence of perfect dominating sets in the families of graphs mentioned, we use many different techniques, depending on the particular graphs under consideration. For example, we introduce linear time algorithms to determine perfect dominating sets in trees, dags, and series-parallel graphs. For tori, hypercubes, cube-connected paths, and directed de Bruijn graphs, a mixture of algebraic and combinatorial methods are used. Several ad hoc methods are required for meshes, cube-connected cycles, and undirected de Bruijn graphs.

2.1 Trees

Let T be a tree with two vertices u_1 and u_2 that have a common parent, and suppose T has the property that any minimum dominating set for T must contain u_1 and u_2. Then T cannot have a perfect dominating set. This means, for example, that a complete m-ary tree of height greater than 1 fails to have a PDS, for any $m > 1$. On the other hand, as we shall see in the following theorem, there are no "forbidden" subgraphs that prevent perfect dominating sets in arbitrary trees or arbitrary graphs.

Theorem 2.1 *Given any graph G and any positive integer d, there is a graph G' containing G as an induced subgraph, such that G' has a distance d PDS. Given any tree T and any positive integer d, there exists a tree T' containing T as a subtree and which has a distance d PDS.*

Proof: Given a graph $G = (V, E)$, let u be a new vertex not in G, and let G' have vertices $V \cup \{u\}$, and edges $E \cup \{\{u, v\} : v \in V\}$. Then $\{u\}$ is a distance d PDS for G' for any d.

For the tree result we will give the proof for $d = 1$ as the proof for $d > 1$ is similar. Let the tree T be given, and suppose r is its root. We will proceed recursively, simultaneously building the tree T' and a perfect dominating set, S, as we go. Initially, $T' = T$ and $S = \phi$. If r is a leaf, add a child q to it and place q in S, otherwise, choose a child of r, say p, add it to S. Recursively apply this procedure to the subtrees rooted at r's other children and to the subtrees rooted at p's grandchildren. □

Note that in the above construction of T', the only vertices added to T were added to leaves of T.

The question of whether an arbitrary tree T has a perfect dominating set can be answered in time that is linear in the number of nodes of T. To see how this can be done, consider the following algorithm.

Let V denote the set of vertices of T and let $l(v)$ denote the label of a vertex $v \in V$, where $l(v)$ is a subset of $\{C, D, N\}$ determined by the rules described below. Conceptually, the label of vertex v holds the information of the possible assignment of v as an element in some PDS that, at least up to that stage of the construction, is possible. Thus, if $C \in l(v)$ then v is already dominated (covered) by one of its children in some PDS construction to that stage, if $D \in l(v)$ then v is not dominated by any of its children and v could be a dominator. Finally, if $N \in l(v)$ then all of v's children are dominated

Algorithm 2.1 (PDS Finder for Trees)

Let V denote the set of vertices of T and let $l(v)$ denote the label of a vertex $v \in V$.

1. *If v is a leaf of T, initialize $l(v) = \{D, N\}$.*
2. *Initialize $node$ = root of T.*
3. *Traverse T in postorder, computing the label of $node$ as soon as the labels of all its children have been computed. Computation stops if the computed label is empty, for no PDS exists for T.*

---

but none of them is a dominator (i.e., v needs to be covered but cannot be a dominator itself). More specifically, if all of v's children have labels, we compute v's label $l(v)$ as follows: $C \in l(v)$ if v has a child whose label contains D while the labels of the remaining children of v all contain C; $D \in l(v)$ provided that N is in the label of each child of v; $N \in l(v)$ if C is in the label of each child of v; if none of these hold, $l(v) = \phi$.

The algorithm proceeds by computing the labels of the vertices of T in postorder and stops either when the label of a node is determined to be empty, in which case T has no PDS, or when the label of the root is determined to be nonempty. All PDSs of T can be generated by using the algorithm, proceeding from the root in a depth-first search traversal, selecting a compatible element from the label of each node. The computation of the labels is outlined in Algorithm 2.1.

We thus have shown the following result, which was also proven earlier in [2].

Theorem 2.2 *Let T be a tree. Algorithm 2.1 determines whether T has a perfect dominating set in time proportional to the number of vertices of T. Moreover, all perfect dominating sets for T are found by this algorithm.* □

A straightforward modification of Algorithm 2.1 and of the definition of $l(v)$ provides an algorithm to determine the existence of a distance d PDS.

Corollary 2.3 *Let T be a tree. For fixed d, the question of whether T has a perfect d-dominating set can be answered in time proportional to the number of vertices of T. Further, if a perfect d-dominating set exists then one can be determined in time proportional to the number of vertices of T.* □

The above methods can also be used to give a linear time decision algorithm to determine whether a directed acyclic graph (dag) has a distance d PDS [16]. Adapting work on series-parallel graphs [18, 23, 24], these methods extend to yield a PDS decision algorithm for them in time proportional to the size of the graphs [16].

2.2 Meshes and Tori

Let $M_k(m_1, m_2, \ldots, m_k)$ denote a k-dimensional mesh of size $m_1 \times m_2 \times \ldots \times m_k$, where we will assume the labelling is such that $m_1 \leq m_2 \leq \ldots \leq m_k$. The vertices of this k-dimensional mesh are identified as k-tuples of integers $(i_1, i_2, \ldots, i_k)$, where $1 \leq i_j \leq m_j$ for $1 \leq j \leq k$. An edge is present between two vertices if and only if their labels differ in only one component and that difference is one.

The k-dimensional torus of size $m_1 \times m_2 \times \ldots \times m_k$, denoted by $T_k(m_1, m_2, \ldots, m_k)$ is the k-dimensional mesh $M_k(m_1, m_2, \ldots, m_k)$ enhanced with *wrap-around* connections. Two vertices $u =$

$(u_1, u_2, \ldots, u_k)$ and $v = (v_1, v_2, \ldots, v_k)$ of the torus have a wrap-around connection in component t provided $\{u_t, v_t\} = \{1, m_t\}$ and, for all $j \neq t$, $u_j = v_j$.

It is easy to characterize those 1-dimensional meshes and tori that have a distance d PDS. We state these in the following.

Theorem 2.4 *The 1-dimensional mesh $M_1(m)$ always has a distance d PDS for any d. The 1-dimensional torus $T_1(m)$ has a distance d PDS if and only if $m \equiv 0 \bmod (2d+1)$.* □

In the next theorem, we give a characterization of the 2-dimensional tori $T_2(m,n)$ that have a distance d PDS. Our proof of the characterization, which appears in [16], shows also that for each d there is only one distance d PDS, up to isomorphism.

Theorem 2.5 *The 2-dimensional torus $T_2(m,n)$ has a perfect d-dominating set if and only if $\{m,n\}$ is a member of*

$$\{\{2, 4dp\}, \{4, (4d-2)p\}, \{6, (4d-4)p\}, \ldots, \{2d, (2d+2)p\} : p \geq 1\} \cup$$
$$\{\{(2d^2+2d+1)p, (2d^2+2d+1)q\} : p, q \geq 1\}.$$

□

Thus we see that the only 2-dimensional tori for which distance 1 perfect dominating sets exist are $T_2(2, 4p)$, $T_2(4, 2p)$, and $T_2(5p, 5q)$, where p and q are positive integers.

While we have not completed the PDS characterization for all 3-dimensional tori, we have found several instances for which a PDS exists. For example, $T_3(2, 3p, 6q)$ has a PDS for all positive integers p and q. In addition, for arbitrary positive integers $p_1, p_2, \ldots, p_k$, the torus $T_k((2k+1)p_1, (2k+1)p_2, \ldots, (2k+1)p_k)$ has a PDS. Perfect dominating sets for these tori appear in [16].

Any distance d PDS for the 2-dimensional torus $T_2(m,n)$, with $m, n \geq 2d+1$, can be used to construct a distance d PDS for the infinite 2-dimensional mesh $M_2(\infty, \infty)$, where we think of the torus as being unrolled and copies of it placed, non-overlapping, to cover the mesh. In [3], Bange et al. noticed a periodic distance 1 PDS for the infinite 2-dimensional mesh and used it to construct dominating sets for 2-dimensional meshes of finite size. The periodic distance 1 PDS observed by Bange et al. is also the PDS constructed in [16] for $T_2(5,5)$.

In the following two theorems we characterize all 2- and 3-dimensional meshes of finite size that possess a PDS.

Theorem 2.6 *The 2-dimensional mesh $M_2(m,n)$ has a PDS if and only if either $m = n = 4$, or $\{m,n\}$ is a member of the set $\{\{2, 2p+1\} : p \text{ a positive integer}\}$. Furthermore, the PDS for $M_2(4,4)$, unique up to automorphism, is $\{(1,2), (2,4), (3,1), (4,3)\}$. The PDS for $M_2(2, 2p+1)$, S_p, is also unique up to automorphism, where*

$$S_p = \{(1 + (i \bmod 2), 1+2i) : 0 \leq i \leq p\}.$$

Proof: First, it is straightforward to show that the given sets are indeed perfect dominating sets for the specified meshes.

Now, suppose S is a PDS for $M_2(2,n)$. If $(1,1)$ is not in S, then exactly one of $(1,2)$ and $(2,1)$ must be in S. If $(1,2)$ is in S then $(2,1)$ cannot be dominated without overlapping, and if $(2,1)$ is in S then the result will be a dominating set automorphic to one including $(1,1)$. We conclude that, without loss of generality, $(1,1) \in S$. From this point, the remaining elements of S are completely determined and we find that $n \equiv 1 \pmod 2$ and that S is as described in the statement of the theorem.

If T denotes a PDS for $M_2(m,n)$, where $m, n > 2$, then the assumption that $(1,1)$ is in T leads to a contradiction. It follows that exactly one of $(1,2)$ and $(2,1)$ must be in T, and without loss of generality we take $(1,2)$ to be in T. This forces $(3,1), (2,4)$, and $(4,3)$ to be in T. Thus, if $n = 4$

then T is a PDS for $M_2(4,4)$. If $n > 4$, however, then $(1,6) \in T$, leaving $(3,5)$ undominated. $\Box$

A characterization of perfect dominating sets for 3-dimensional meshes can be found by similar methods, as the proof of the following theorem illustrates.

Theorem 2.7 *Let $2 \le m_1 \le m_2 \le m_3$. The 3-dimensional mesh $M_3(m_1, m_2, m_3)$ has a PDS if and only if $m_1 = m_2 = m_3 = 2$.*

Proof: Suppose S is a PDS for $M_3(m_1, m_2, m_3)$. It is easy to check that $\{(1,1,1), (2,2,2)\}$ is a PDS for $M_3(2,2,2)$, and that it is unique up to automorphism. Thus, it remains to show that $M_3(m_1, m_2, m_3)$ has no PDS for any other values of m_1, m_2, m_3.

Case 1: suppose $m_1 = m_2 = 2$ and $m_3 > 2$. If $(1,1,1) \in S$, then $(2,2,2)$ must be in S which leaves $(2,1,3)$ and $(1,2,3)$ undominated. Consider, then, the possibility that $(1,1,1)$ is not in S. We must have $(2,1,2)$ and therefore $(1,2,1)$ in S. This is equivalent to the case $(1,1,1) \in S$ under automorphism, however.

Case 2: suppose $m_1 = 2$ and $m_2 > 2$. If $(1,1,1) \in S$, then we consider which element of S can dominate $(1,2,2)$. The assumption that $(1,2,3) \in S$ leads to the fact that $(2,1,2)$ is not dominated; the assumption that $(2,2,2) \in S$ leads to the conclusion that $(2,1,4)$ is in S, leaving $(1,1,3)$ undominated.

Case 3: suppose $m_1 \ge 3$. A similar but more lengthy case-by-case analysis establishes that no mesh $M_3(m_1, m_2, m_3)$ has a PDS. $\Box$

For distance $d > 1$, we have completed the characterization of all 2-dimensional meshes that have a distance d PDS and the characterization of all perfect d-dominating sets for them. We include, below, the 2-dimensional meshes that have a distance d PDS but the complete statements and proofs of these results are given in [16].

Theorem 2.8 *The 2-dimensional mesh $M_2(n,m)$, $2 \le n \le m$, has a distance d PDS if and only if one of the following holds:*

(*i*) $n + m \le 2d + 1$

(*ii*) $n = m = 2d + 2$

(*iii*) $n \le d{+}1$ *and* $(m \bmod 2d{+}2{-}n) \in \{1, \ldots, 2d{+}3{-}2n\}$

$\Box$

We have shown that weaker results of a similar form hold for k-dimensional meshes, in that $M_k(m_1, \ldots, m_{k-1}, n)$ has a perfect d-dominating set for infinitely many n whenever $d \ge m_1 + \cdots + m_{k-1} - (k-1)$, and there are straightforward formulas giving such n and the perfect d-dominating sets [16]. However, it may be that $M_k(m_1, \ldots, m_k)$ does not have a distance d PDS when $m_1, \ldots, m_k \ge 2d + 1$ for $d > 1$ and $k > 2$.

2.3 Hypercubes and Related Graphs

Here we investigate the existence of distance d perfect dominating sets in the hypercube graph in Section 2.3.1, and in two variations of the hypercube graph: the cube-connected cycles in Section 2.3.2, and the cube-connected paths in Section 2.3.3.

2.3.1 Hypercubes

The hypercube graph of dimension k, which we denote by Q_k, consists of 2^k vertices labeled as binary k-tuples, and $k2^{k-1}$ edges, where each edge joins two vertices whose corresponding k-tuples differ in exactly one component. These graphs have a long and rich history [11] and have recently become an important architectural model for several commercial parallel computers such as the Intel iPSC and the NCUBE machines.

For a hypercube graph of dimension k, a distance d perfect dominating set is precisely a perfect binary d-error-correcting code over an alphabet of 2^k symbols. As illustrated in the survey article of van Lint [25], these codes have been the object of great interest since their inception by Hamming [10] and Golay [9] in the late 1940's. All perfect binary error-correcting codes are known, in fact, all perfect codes over alphabets which are finite fields are known [25]. Perfect binary single-error-correcting codes of length k exist if and only if $k+1$ is a power of 2, and when this is the case, can be constructed directly using algebraic methods such as those described in [4] or can be constructed recursively, as shown by Golay [9]. The only perfect, binary, multiple-error-correcting code is the 3-error correcting Golay code of dimension 23. We state these results in terms of distance d PDS in the following.

Theorem 2.9 *The k-dimensional hypercube, Q_k, has a distance d perfect dominating set if and only if one of the following conditions is satisfied.*

(*i*) $k \leq d$

(*ii*) $d = 1$ *and* $k + 1$ *is a power of two*

(*iii*) $d = 3$ *and* $k = 23$.

□

2.3.2 Cube-Connected Cycles

The cube-connected cycles network of dimension k, which we denote by CCC_k, consists of a k-dimensional hypercube, each of whose 2^k "vertices" consists of a cycle of k nodes [20]. For each dimension, every cycle has a node connected to a corresponding node in the neighboring cycle in that dimension. In order to be more precise, suppose that we label the nodes of CCC_k as ordered pairs (i, α), where $1 \leq i \leq k$ and α is a binary k-tuple. At position α in the k-dimensional hypercube, the k-cycle of nodes are labelled in, say, clock-wise order around the cycle as $(1, \alpha), (2, \alpha), \ldots, (k, \alpha)$. Moreover, two nodes (i, α) and (j, β) are joined by an edge in dimension t if and only if $i = j = t$ and the k-tuples α and β differ only in the t-th component. Figure 1 illustrates CCC_3.

From the description we can see that CCC_k has $k2^k$ nodes, $3k2^{k-1}$ edges, a diameter of $2k$, and each node has degree 3 for $k \geq 3$. These are some of the properties which make cube-connected cycles an interesting alternative architecture for multicomputers.

Characterizing which cube-connected cycles have distance d perfect dominating sets appears to be difficult. Our first result shows that if a distance d PDS exists for a k-dimensional cube connected cycle graph, then a distance d PDS will exist for all dimensions that are a multiple of k.

Theorem 2.10 *If CCC_k has a perfect d-dominating set, then so does CCC_{mk} for all $m \geq 1$ and* $k \geq 2d + 1$.

Proof: We form a projection π of CCC_{mk} onto CCC_k by $\pi : (w, \beta) \mapsto (w \bmod k, \alpha)$, where $\alpha_i = \beta_i \oplus \beta_{i+k} \oplus \cdots \oplus \beta_{i+(m-1)k}$ for all $1 \leq i \leq k$, and where $\oplus$ denotes the *exclusive or* operation. It is straightforward to verify that with this projection, CCC_{mk} is a covering space of CCC_k such that given any point (i, α) in CCC_k and any point (w, β) mapping onto (i, α), the distance $\lfloor (k-1)/2 \rfloor$-neighborhood of (w, β) is mapped isomorphically onto the distance $\lfloor (k-1)/2 \rfloor$-neighborhood of

(i, α). To construct a distance d PDS of CCC_{mk}, let S_k be a distance d PDS of CCC_k, and let $S_{mk} = \pi^{-1}(S_k)$. A straightforward argument establishes that S_{mk} is a distance d PDS for CCC_{mk}, since $d \leq \lfloor (k-1)/2 \rfloor$. $\square$

We summarize in the table below what we have learned about the existence of a PDS for CCC_k for $1 \leq k \leq 9$.

The Table 1 entry for $k = 1$ clearly holds. The case $k = 2$ is easy, too, for equation (1) becomes $3 \cdot |S| = 2 \cdot 2^2$, which is impossible.

We verify the cases $k = 3, 4$ by exhibiting perfect dominating sets S_3 and S_4 for CCC_3 and CCC_4, respectively. A PDS for CCC_3 is

$$S_3 = \{(1, 000), (2, 110), (3, 101), (1, 111), (2, 001), (3, 010)\}$$

and is unique up to isomorphism. It was also shown in [21] that a perfect dominating set exists for CCC_3. Now, any perfect dominating set for CCC_4 contains 16 elements, with one element of the PDS in each 4-cycle at the hypercube nodes. The set

$$\{(2, \alpha) : \alpha \text{ is of odd parity}\} \cup \{(4, \alpha) : \alpha \text{ is of even parity}\}$$

is a PDS for CCC_4, as can be easily verified.

A simple counting argument establishes the fact that CCC_5 has no PDS. By equation (1), if a PDS, say S, did exist, it must contain 40 elements. Thus, some hypercube node must have at least 2 elements of S in its 5-cycle, but that is impossible.

Table entries for $k = 6, 8$, and 9 follow from Theorem 2.10.

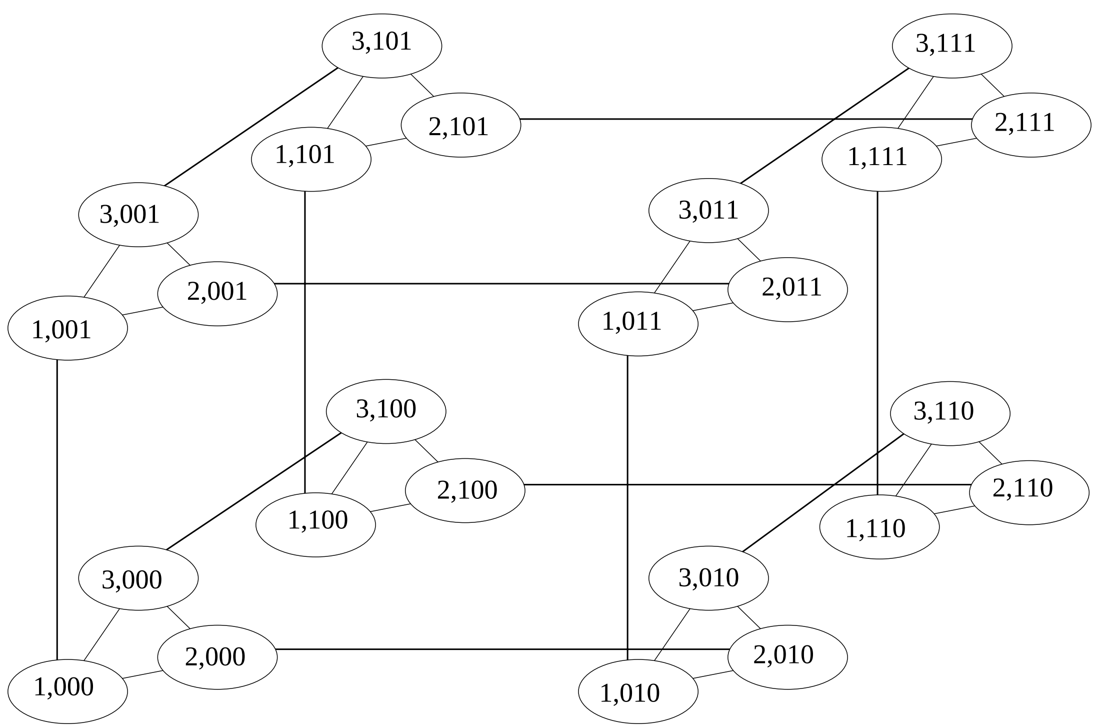

Figure 1: Cube-Connected Cycles of Dimension 3.

k	PDS for CCC_k
1	yes
2	no
3	yes
4	yes
5	no
6	yes
7	?
8	yes
9	yes

Table 1: Existence of a PDS for CCC_k

2.3.3 Cube-Connected Paths

The cube-connected paths network of dimension k, which we denote by CCP_k, is very similar to the cube-connected cycles except that the cycles at the hypercube nodes are replaced by paths of length k. For each dimension, the path at a hypercube node is connected to the corresponding node in the path located at the neighboring hypercube node in that dimension. As in the cube-connected cycles, we label the nodes of CCP_k as ordered pairs (i, α), where $1 \le i \le k$ and α is a binary k-tuple. At position α in the k-dimensional hypercube, the nodes of the path of length k of nodes are labelled $(1, \alpha), (2, \alpha), \ldots, (k, \alpha)$. Notice that two nodes (i, α) and (j, β) are joined by an edge in dimension t if and only if $i = j = t$ and the k-tuples α and β differ only in the t-th component. An example of CCP_3 is pictured in Figure 2.

From the description of cube-connected paths, we see that CCP_k has $k2^k$ nodes, $(3k-2)2^{k-1}$ edges, a diameter of $3k - 2$, and node degrees 2 and 3, where nodes with labels $(1, \alpha)$ and (k, α) have degree 2, and other nodes have degree 3. One advantage of the cube-connected paths over cube-connected cycles is that a CCP_{k+1} has two disjoint copies of CCP_k in it, while the corresponding statement for CCC_k is false. This permits some recursive algorithms to be developed more cleanly on the CCP than on the CCC.

Theorem 2.11 *If k is an odd positive integer, then CCP_k has a perfect dominating set which is unique up to isomorphism and is given by*

$$\begin{aligned} T_k &= \{(i, \alpha) : i \equiv 1 \pmod 4 \ \text{ and } \alpha \text{ is of even parity}\} \quad \cup \\ &\quad \{(i, \alpha) : i \equiv 3 \pmod 4 \ \text{ and } \alpha \text{ is of odd parity}\}. \end{aligned}$$

If k is even then CCP_k has no perfect dominating set.

Proof: In the case when k is odd, the demonstration that T_k is a PDS for CCP_k is straightforward, so we do not include it here.

Suppose S is a perfect dominating set for CCP_k. For each $1 \le i \le k$, let x_i denote the number of elements of S with label (i, α), where α is arbitrary. Since any node of the form (i, α) must be adjacent to a unique element of S of the form $(i-1, \alpha)$, $(i+1, \alpha)$, or (i, β), where the binary k-tuples α and β differ only in their i^{th} component, the x_i must satisfy the following system of equations:

$$\begin{array}{lllllllllll} 2x_1 & + & x_2 & & & & & & & & = 2^k \\ x_1 & + & 2x_2 & + & x_3 & & & & & & = 2^k \\ & & x_2 & + & 2x_3 & + & x_4 & & & & = 2^k \\ & & & & & & \cdots & & & & \\ & & & & & & x_{k-2} & + & 2x_{k-1} & + \; x_k & = 2^k \\ & & & & & & & & x_{k-1} & + \; 2x_k & = 2^k \end{array}$$

Let d_k denote the determinant of the matrix of coefficients of the above system of equations. Since $d_2 = 3$, $d_3 = 4$ and $d_k = 2d_{k-1} - d_{k-2}$ for $k \geq 3$, it follows that $d_k = k + 1$ for all $k \geq 2$. Thus, the above system has a unique solution. The uniqueness of the PDS, up to isomorphism, now follows easily for k odd.

For k even, say $k = 2m$, and, for $1 \leq i \leq m$, a solution to this system is obtained by taking $x_{2i-1} = (m-i+1)2^k/(2m+1)$ and $x_{2i} = i2^k/(2m+1)$. Since these are not integer values, we may conclude that CCP_k has no PDS when k is even. □

2.4 Binary de Bruijn Graphs

Let B_k denote the graph whose vertices are binary k-tuples, and for which an edge is directed from the vertex $x_1x_2 \ldots x_k$ to vertex $y_1y_2 \ldots y_k$ provided that $x_2x_3 \ldots x_k = y_1y_2 \ldots y_{k-1}$. This graph, known as a (directed) *binary de Bruijn graph* or *shift register graph*, is illustrated in Figure 3 for $k = 3$. Parallel computers based on de Bruijn graphs have recently been suggested as alternatives to hypercubes [19].

When it is notationally convenient, we will denote the vertices of B_k as $0, 1, \ldots 2^k - 1$, using the decimal equivalent of their binary k-tuple. With this notation, vertex j has an edge directed from it to vertex $2j$ and to vertex $2j + 1$, where these numbers are taken modulo 2^k.

The following theorem shows that the graphs B_k have perfect dominating sets for all $k \geq 1$, and that for any d, distance d perfect dominating sets exist for infinitely many k. The theorem leaves open the possibility there are distance d PDS for other values of k, a possibility partially discussed in Proposition 2.13.

Theorem 2.12 *For any $d \geq 1$ and for k a positive integer of the form $(d+1)m$ or $(d+1)m - 1$ or $k < d$, let T_k denote a subset of the vertices of B_k defined as follows.*

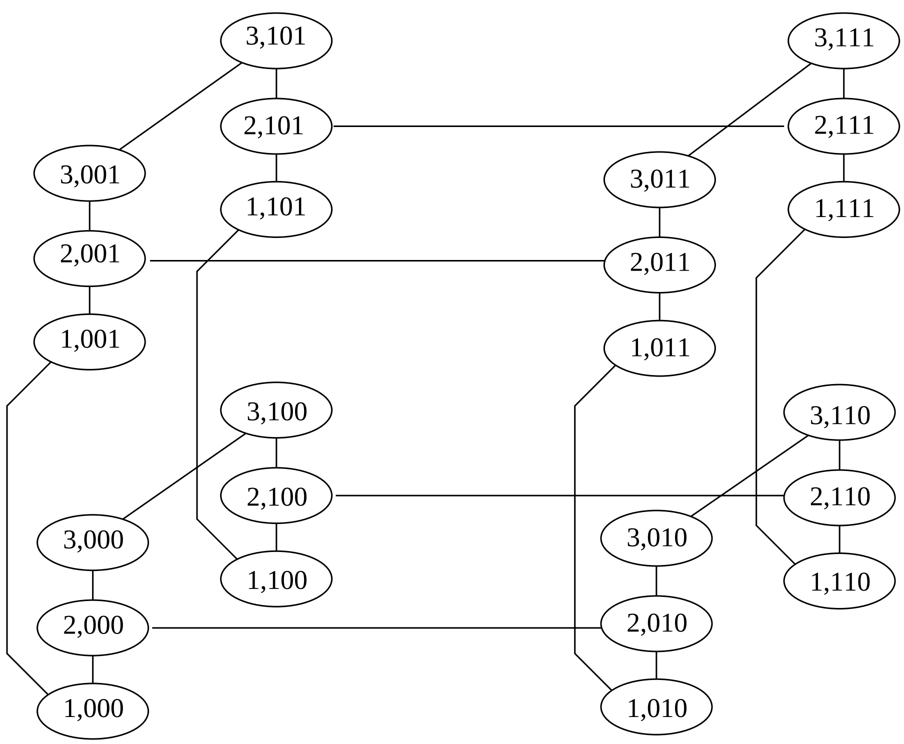

Figure 2: Cube-Connected Paths of Dimension 3.

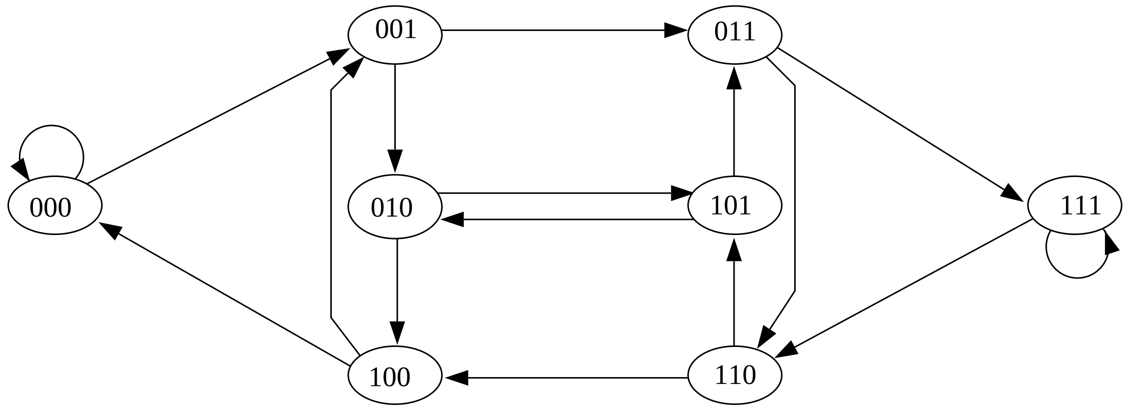

Figure 3: de Bruijn Graph for $k = 3$.

(*i*) $T_1 = T_2 = \ldots = T_d = \{0\}$,

(*ii*) $T_{(d+1)(m+1)-1} = T_{(d+1)m-1} \cup \{j : 2^{(d+1)m-1} \leq j \leq 2^{(d+1)m} - 1\}$,

(*iii*) $T_{(d+1)m} = T_{(d+1)m-1} \cup \{2^{(d+1)m} - 1 - s : s \in T_{(d+1)m-1}\}$.

Then the set T_k is a perfect dominating set for B_k.

Proof: It is easy to check that the set $\{0\}$ is a distance d PDS for B_k when $k \leq d$. For k of the form $(d+1)m - 1$, the fact that T_k is a perfect d-dominating set for B_k follows by induction and the fact that all vertices of the form $2^\alpha j + \beta$ are within distance d of vertex j for all $0 \leq j < 2^{k-d}$ and $1 \leq \alpha \leq d$, $0 \leq \beta < 2^\alpha$. For k of the form $(d+1)m$, the result follows by virtue of the fact that T_k is the union of T_{k-1}, which is a distance d PDS for B_{k-1}, and the "reflection" of T_{k-1} found by taking the exclusive or of its elements with the k-tuple binary representation of $2^k - 1$. This reflection provides a unique dominator for every vertex j such that $j \geq 2^{k-1}$. □

It may be the case that the only distance d dominating sets are of the form given above, and in particular that no perfect 2-dominating sets are possible for B_k when $k-1$ is a multiple of 3. However, we have been unable to prove or disprove that statement. To see that none can exist for B_4, we offer the following argument. First, $N_2(B_4, v)$ has: 4 elements if $v = 0, 15$; 5 elements if $v = 7, 8$; 6 elements if $v = 5, 10$; and 7 elements otherwise. The criterion supplied by equation (1) serves to reduce the possibilities to just three, corresponding to the partitions $4+6+6$, $4+5+7$, and $5+5+6$. The case corresponding to the partition $4 + 6 + 6$ cannot occur because $N_2(B_4, 5)$ and $N_2(B_4, 10)$ overlap. The remaining two cases are settled similarly. We have thus proved the following.

Proposition 2.13 *The graph B_4 has no perfect 2-dominating set.* □

Let us consider the undirected analog of B_k, which we denote by B_k^*. Vertex j in B_k^* is adjacent to vertices $2j, 2j+1, \lfloor j/2 \rfloor, 2^{k-1} + \lfloor j/2 \rfloor$, where these numbers are taken modulo 2^k. To illustrate, vertex 0 is adjacent to vertices $0, 1$ and 2^{k-1}; vertex $2^k - 1$ is adjacent to $2^{k-1} - 1$, $2^k - 2$ and $2^k - 1$. Since the radius of B_k^* is $k - 1$ when $k > 1$, it follows that a distance $k - 1$ PDS exists for all $k > 1$. On the other hand, although $\{1\}$ is a PDS for B_1^* and B_2^*, we have found no other values of k for which a PDS exists. In the proof of the following proposition, we give a case-by-case analysis which shows that no PDS exists for $k = 3, 4, 5$.

Proposition 2.14 *The undirected deBruijn graph B_k^* has a perfect dominating set for $k = 1, 2$, but has no perfect dominating set for $k = 3, 4, 5$.*

Proof: The fact that $\{1\}$ is a PDS for B_1^* and B_2^* was noted above. For the remaining cases, $n_1(B_k^*, 0) = n_1(B_k^*, 2^k - 1) = 3$, and it is easy to see that these are the only two vertices v for which $n_1(B_k^*, v) = 3$. Further, there are only two vertices for which $n_1(B_k^*, v) = 4$, namely $2 + 2^3 + \ldots + 2^{k-2}$ and $1 + 2^2 + \ldots + 2^{k-1}$ if k is odd and $2 + 2^3 + \ldots + 2^{k-1}$ and $1 + 2^2 + \ldots + 2^{k-2}$ if k is even. The remaining vertices have $n_1(B_k^*, v) = 5$. Now, suppose S is a PDS for B_k^* for some k. Let x_i be the number of elements of S for which $n_1(B_k^*, v) = i$. As we have just observed, x_i must be 0 unless $i = 3, 4, 5$. Moreover, $x_4 \leq 1$ because the two elements with $n_1(B_k^*, v) = 4$ have overlapping neighborhoods. It follows that

$$3 \cdot x_3 + 4 \cdot x_4 + 5 \cdot x_5 = 2^k. \tag{2}$$

Suppose $k = 3$. Equation (2) cannot hold unless $x_3 = x_5 = 1$. Without loss of generality we may assume $0 \in S$. This means that any vertex of B_3^* whose neighborhood overlaps $\{0, 1, 4\}$ cannot be in S. This leaves only the possibilities $S = \{0, 3\}$ and $= \{0, 7\}$, neither of which are a PDS.

If $k = 4$ then Equation (2) cannot hold unless $x_3 = x_5 = 2$. With both 0 and 15 in S, we find that the remaining 2 elements of S must be from the set $\{6, 9, 11\}$. In order that element 3 be dominated, we must have $6 \in S$, but neither of the neighborhoods of 9 and 11 are disjoint from the neighborhood of 6. Thus, B_4^* has no PDS.

Now suppose $k = 5$. Equation (2) cannot hold unless $x_3 = x_4 = 1$, which means that one of $\{0, 31\}$ and one of $\{10, 21\}$ must be in S. Without loss of generality we assume $0 \in S$. If $10 \in S$, then either 15 or 30 must be in S. If $15 \in S$, then the only possibilities for the remaining 4 elements of S come from the set $\{4, 6, 8, 9, 12, 13, 17, 22, 25\}$. A careful analysis of neighborhoods shows that it is impossible to complete the description of S in this case. In case $30 \in S$, the remaining 4 elements of S must come from the set $\{4, 6, 9, 12, 17, 19, 22\}$. Again, it is impossible to complete S. We may conclude that 10 is not in S and that 21 is in S. Again, one of 15 or 30 is in S. In the case $15 \in S$, the remaining 4 elements of S must be in $\{4, 6, 9, 12, 17, 18, 25\}$; in the case $30 \in S$, the remaining 4 elements of S must come from $\{4, 6, 9, 12, 17, 18, 19\}$. In both of these cases, we find that S cannot be completed. Therefore no PDS exists for B_5^*. □

3 Variations on PDS

Domination, in general, can be thought of as a binary relation of the form "x dominates i", where $x \in X$, $i \in I$, and X need not be the same as I. Perfect dominating sets can be defined for this general notion of domination as follows. We call a dominating set $S \in X$ *perfect* if each $i \in I$ is dominated by a unique $x \in S$. With this definition, a perfect dominating set is not necessarily of minimum size unless $X = I$ and the domination is symmetric. For example, in vertex-edge domination on undirected graphs, consider a path P of 3 vertices. The two end vertices form a PDS, but the center vertex forms a PDS also. In general, a perfect dominating set may not be of minimum size, although it is always a minimal dominating set.

Let us consider some of these variations on domination in relation to the families of graphs considered in Section 2. While these variations have extension to distance $d > 1$, we will restrict our discussion to $d = 1$. In vertex-to-edge domination, denoted *ve-domination*, each vertex v dominates all the edges of the form (v, q). There is a simple characterization for all undirected graphs that have a perfect ve-dominating set.

Theorem 3.1 *The undirected graph G has a perfect ve-dominating set if and only if G is bipartite. If G is bipartite, say $G = (U_1 \cup U_2, E)$ with each edge in E incident with one vertex in U_1 and one in U_2, then either of the sets U_1 or U_2 is a perfect ve-dominating set for G.* □

In edge-to-vertex domination, denoted by *ev-domination*, an edge (u, v) dominates vertex v. Undirected graphs which have perfect dominating sets in this sense are also easily characterized.

Theorem 3.2 *The undirected graph G has a perfect ev-dominating set if and only if G has a complete matching, in which case the set of edges in the complete matching form a PDS.* □

In Q_k, for example, we see that a perfect ev-dominating set is the set of edges with one vertex in one $(k-1)$-dimensional subcube and the other vertex in a vertex-disjoint $(k-1)$-dimensional subcube. Similar constructions yield perfect ev-dominating sets for CCC_k and CCP_k.

For edge-to-edge domination, denoted by ***ee*-*domination***, an edge (u, v) dominates all edges of the form (v, p). Although we do not have a complete characterization of graphs with perfect dominating sets of this type, we will list a few of the more immediate results. The techniques of Section 2.1 can be used to yield a linear time decision algorithm that decides whether a tree has a perfect ee-dominating set. As a consequence of the fact that an undirected graph with a 4-cycle can have no perfect ee-dominating set, we find that meshes, tori, and hypercubes have no perfect ee-dominating sets for dimension 2 or more. It follows similarly that CCC_3 and CCC_4 have no perfect dominating set of this type. On the other hand, it is easy to show that CCP_3 does have a perfect ee-dominating set.

In [14], Laskar and Peters study the domination numbers of arbitrary graphs for ve-, ev-, and ee-domination.

4 Related Questions

Throughout the paper we noted several families of graphs for which the characterization of those with perfect d-dominating sets is incomplete. It would be of particular interest, however, to complete the characterization of graphs with perfect dominating sets for higher dimensional meshes and tori, cube-connected-cycles, and the undirected de Bruijn graphs.

There is a closely related question which arises from the placement of several different types of resources in a parallel computer. In order to perform this placement optimally on a given interconnection network, one is interested in the existence of several disjoint perfect d-dominating sets for the graph of the network. Consider, for example, the torus $T_2(5, 5)$ with one PDS given by $\{(1, 1), (2, 4), (3, 2), (4, 5), (5, 3)\}$. Using this set and its translates, it is straightforward to check that $T_2(5, 5)$ has the maximum possible of 5 disjoint PDSs. For the hypercubes Q_k which have a PDS, namely those with $k = 2^t - 1$, a perfect code for Q_k, together with its cosets, yield a partition of the nodes of Q_k, where each cell of the partition is a PDS. Existence of multiple perfect d-dominating sets for other families of graphs is largely unexplored.

There are a number of interesting and quite general questions that involve constructing perfect dominating sets for a graph from a given subgraph, or involve determining subgraphs which prevent a graph from possessing a PDS. We mention just two of these here and refer the reader to [17] for further discussion of these.

Question 1. What are the *forbidden subgraphs* of perfect domination for trees (or some other large class of graphs) for various notions of domination?

In addition to the subgraph requirement, more specifications may be added, such as for trees, one may wish to specify that certain leaves of the subgraphs are also leaves of the tree.

For the vv-domination considered in this paper, Theorem 2.1 showed that there are no forbidden subgraphs. However, if one requires that certain nodes of the subgraph are leaves, then there are forbidden subgraphs. For example, the complete binary tree of height 2, where all the leaves of the binary tree must be leaves of the tree T, is a forbidden subgraph if T is to have a PDS.

In the case of ve-domination, for example, recall that a graph has a PDS of this type if and only if it is bipartite. Thus, we find that a graph has a perfect ve-dominating set if and only if it does not contain an odd cycle. The forbidden subgraphs for perfect ve-domination for the class of all undirected graphs is the set of odd cycles. In the case of edge-vertex domination, however, there are no forbidden subgraphs for the class of all graphs. For, given any graph $G = (V, E)$, consider the product graph $G \times Q_1$, in which corresponding to every vertex $v \in V$ there is a vertex v' (in the copy of G) with an edge between v and v'. The set of new edges $\{v, v'\}$ forms a perfect ev-dominating set.

Question 2. Given a graph $G = (V, E)$, and a subset S of V (or of E), is it possible to add edges to G to make S a perfect dominating set for some supergraph G' of G?

Admittedly, Question 2 is open-ended as stated. One can impose additional constraints on G' such as requiring that it be connected, say, or that it be planar. If we require that the graph G' be connected, and G and G' are undirected, then we have the following observations. To avoid trivialities, we assume that G has at least two vertices.

If vertex-to-vertex domination is used, we find that a connected G' can be created if and only if each of the following conditions holds.

- $|S| \geq 1$
- No vertex of G is dominated by more than one vertex of S
- The number of undominated vertices in G is at least as large as the number of isolated vertices of G that are in S.

If vertex-to-edge domination is used, then a connected G' can be created if and only if each of the following conditions holds.

- $|V| - 1 \geq |S| \geq 1$
- Every edge of G is dominated by exactly one vertex of S.

If edge-to-vertex domination is used, then a connected G' can be created if and only if S is a perfect matching for G.

If edge-to-edge domination is being used, then a connected G' can be created if and only if each of the following conditions holds.

- $|S| \geq 1$
- Every edge of G is dominated by exactly one edge of S
- G has only two vertices, or at least one vertex is not an endpoint of an edge of S.

Proofs of the above observations will appear in [17].

References

[1] D. Bange, A. Barkauskas, and P. Slater, “Disjoint dominating sets in trees”, Sandia Labs. tech. report SAND 78-1087J (1987).

[2] D. Bange, A. Barkauskas, and P. Slater, “Efficient dominating sets in graphs”, *Applications of Discrete Math.*, R.D. Ringeisen and F.S. Roberts, eds., SIAM, Philad. (1988) pp. 189-199.

[3] D. Bange, A. Barkauskas, L.H. Host, and P. Slater, “Efficient near-domination of grid graphs”, *Congr. Numer.* 58 (1987) pp. 83-92.

[4] E.R. Berlekamp, *Algebraic Coding Theory*, Aegean Park Press, CA. Revised 1984 edition.

[5] N. Biggs, “Perfect codes in graphs”, *J. Comb. Theory* (B) 15 (1973) pp. 289-296.

[6] N. Biggs, “Perfect codes and distance transitive graphs”, *Combinatorics, Proc. Third British Comb. Conf.* (1973) London Math. Soc. Lecture Notes 13 (F.P. McDonough and V.C. Mavron, eds.) pp. 1-8.

[7] E.J. Cockayne, E.O. Hare, S.T. Hedetniemi, and T.V. Wimer, “Bounds for the domination number of grid graphs”, *Congr. Numer.* 47 (1985) pp. 217-228.

[8] M.R. Garey and D.S. Johnson, *Computers and intractability: a guide to the theory of NP-completeness* (1979) W. H. Freeman & Co. New York.

[9] M.J.E. Golay, “Notes on digital coding”, *Proc. I.R.I. (IEEE)* 37 (1949) p. 657.

[10] R.W. Hamming, “Error detecting and error correcting codes”, *Bell System Tech. J.* 29 (1950) pp. 147-160.

[11] F. Harary, J.P. Hayes, and H. Wu, “A survey of the theory of hypercube graphs”, *Comput. Math. Appl* 15 (1988) pp. 277-289.

[12] O. Heden, “Perfect codes in antipodal distance transitive graphs”, Math. Institute of Univ. of Stockholm preprint (1974).

[13] D.S. Johnson, “The NP-completeness column: an ongoing guide”, *J. Algorithms* 6 (1985) pp. 434-451.

[14] R. Laskar and K. Peters, “Vertex and edge domination parameters in graphs”, *Congr. Numer.* 48 (1985), pp. 291-305.

[15] M.L. Livingston and Q.F. Stout, “Distributing resources in hypercube computers”, *Proc. 3rd Conf. on Hypercube Concurrent Computers and Appl.* (1988) pp. 222-231.

[16] M.L. Livingston and Q.F. Stout, “On problems of distributing resources in parallel computers”, in preparation.

[17] M.L. Livingston and Q.F. Stout, “Constructability and perfect dominating sets”, in preparation.

[18] J. Pfaff, R. Laskar, and S.T. Hedetniemi, “Linear algorithms for independent domination and total domination in series-parallel graphs”, *Congr. Numer.* 45 (1984) pp. 71-82.

[19] D.K. Pradhan and M.R. Samatham, “The de Bruijn multiprocessor network: a versatile parallel processing and sorting network for VLSI”, *IEEE Trans. Computers* 38 (1989) pp. 567-581.

[20] F.P. Preparata and J. Vuillemin, “The cube-connected cycles: a versatile network for parallel computation”, *Commun. Assoc. Comput. Mach.*, 24 (1981) pp. 300-309.

[21] A. Reddy, P. Banerjee and S. Abraham, “I/O embeddings in hypercubes”, *Proc. Int. Conf. on Parallel Processing* (1988) pp. 331-338.

[22] D.H. Smith and P. Hammond, “Perfect codes in the graphs O_k”, *J. Comb. Theory* (1975)

[23] K. Takamizawa, T. Nishezeki, and N. Sato, “Linear-time computability of combinatorial problems on series-parallel graphs”, *Jour. Assoc. Comput. Mach.*, 29 (1982) pp. 623-641.

[24] J. Valdes, R.E. Tarjan, and E.L. Lawler, “The recognition of series-parallel digraphs”, *SIAM Jour. Comp.* 11 (1982) pp. 298-313.

[25] J.H. van Lint, “A survey of perfect codes”, *Rocky Mountain Jour. of Math.*, 5 (1975) pp. 199-224.