Redundance of Complete Grid Graphs

David R. Guichard

August 26, 2002

Abstract

The redundance $R(G)$ of a graph G is the minimum, over all dominating sets S, of $\sum_{v\in S} 1 + \deg(v)$, where $\deg(v)$ is the degree of vertex v. We use some dynamic programming algorithms to compute the redundance of complete grid graphs $G_{m,n}$ for $1 \le m \le 21$ and all n, and to establish good upper and lower bounds on the redundance for larger m. We conjecture that the upper bound is the redundance when $m > 21$.

1 Preliminaries

A dominating set S for a graph G is a subset of the vertices of G that either contains or is adjacent to every vertex of G. The domination number of G, $\gamma(G)$, is the minimum size of a dominating set, or equivalently the minimum, over all dominating sets S, of $\sum_{v\in S} 1$. The redundance of a graph G, $R(G)$, is the minimum, over all dominating sets S, of $I(S) = \sum_{v\in S} 1+\deg(v)$, where $\deg(v)$ is the degree of vertex v; $I(S)$ is the *influence* of S. For an introduction to the domination number and redundance, see [3].

Let P_n denote the path on n vertices; the complete grid graph $G_{m,n}$ is the product $P_m \times P_n$. "Grid graph" usually denotes a subgraph of a complete grid graph; here we use "grid graph" to mean complete grid graph for simplicity. The domination number of grid graphs has been the subject of previous work; here we adapt the most successful algorithms for the computation of the domination number of grid graphs to compute the redundance of all such graphs $G_{m,n}$ with $m \le 21$, and to compute upper and lower bounds for the redundance of all grid graphs. Figure 1 shows a

JCMCC 47 (2003), pp. 201-212

dominating set of the 14×14 grid graph that realizes both the domination number and the redundance.

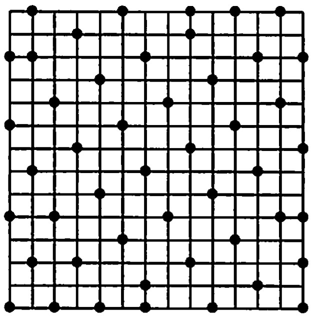

Figure 1: The 14×14 grid graph has $\gamma = 47$ and $R = 215$.

Fisher [1, 2] developed a fast dynamic programming algorithm to compute the domination number of grid graphs; we describe a modification of the algorithm that will allow us to compute the redundance. (The two papers by Fisher cover essentially the same ground in somewhat different ways. Both papers are unpublished; our treatment here is closer to [1] than the other.)

2 Computing Exact Values of R

Imagine a grid graph with a designated subset S of the vertices, as in Figure 1. We describe a column (a copy of P_m) in such a diagram by a state vector $\mathbf{s}$, in which $\mathbf{s}_j$ is 0 if vertex number j on the path is in S, 1 if vertex j is adjacent to a member of S in $G_{m,n}$, and 2 otherwise. The vertices coded 0 or 1 are those that are dominated by S. For example, the rightmost column in Figure 1 has state vector $(0,1,0,1,0,1,1,0,1,1,1,0,1,1)$; since S is a dominating set, there are no entries of 2 in any state vectors for this diagram. Note that no state vector can ever have a 0 adjacent to a 2.

We partially order the state vectors by defining $\mathbf{t} \geq \mathbf{s}$ if for all j, $\mathbf{t}_j \leq \mathbf{s}_j$; think of the column with state $\mathbf{t}$ as being "more dominated" than that with state $\mathbf{s}$.

An *exact* $\mathbf{s}$*-domination* of $G_{m,n}$ is a subset S of the vertices that dominates the first $n-1$ columns and for which the state vector of the final column is $\mathbf{s}$. An $\mathbf{s}$*-domination* is any exact $\mathbf{t}$-domination with $\mathbf{t} \geq \mathbf{s}$.

Let $\overline{R}_{m,n}(\mathbf{s})$ be the minimum influence of any exact $\mathbf{s}$-domination of

$G_{m,n}$, and $R_{m,n}(\mathbf{s}) = \min_{\mathbf{t} \geq \mathbf{s}} \overline{R}_{m,n}(\mathbf{t})$, that is, the minimum influence of any s-domination. The algorithm computes $R_{m,n}(\mathbf{s})$ for fixed m and all $n \leq N$ and all s incrementally, computing all values $R_{m,n}(\mathbf{s})$ after the values $R_{m,n-1}(\mathbf{s})$. The redundance of $G_{m,n}$ is then $R_{m,n-1}(\mathbf{1})$, where $\mathbf{1}$ is the vector all of whose entries are 1.

The computation of $R_{m,n}(\mathbf{s})$ from $R_{m,n-1}(\mathbf{s})$ is complicated by the fact that the degree of a vertex in column $n-1$ changes when column n is added. To account for this, we compute $R^+_{m,n}(\mathbf{s})$, in which the contribution of a vertex in column n includes the degree of the vertex in $G_{m,n+1}$. From $R^+_{m,n-1}(\mathbf{s})$ we then compute both $R^+_{m,n}(\mathbf{s})$ and $R_{m,n}(\mathbf{s})$. More precisely, for $S \subseteq V(G_{m,n})$, denote by $|S_n|$ the number of 0 entries in the state vector for column n of S. Then $R^+_{m,n}(\mathbf{s})$ is the minimum of $I(S) + |S_n|$ taken over all s-dominations S, and $\overline{R}^+_{m,n}(\mathbf{s})$ is the minimum over all exact s-dominations S.

When $n = 1$ and $n = 2$, there exist s for which no exact s-domination exists, for example, $\mathbf{1}$ and $\mathbf{2}$, respectively. In such cases, we set $\overline{R}$ and $\overline{R}^+$ to ∞. Note that for every s and every $n \geq 1$ there is an s-domination, because $\mathbf{0} \geq \mathbf{s}$.

For a particular state s, some states cannot occur in column $n-1$ if column n is to have state at least s. Fisher [1] defines

$$p(\mathbf{s})_j = \begin{cases} 2 & \text{if } \mathbf{s}_j = 0, \\ 0 & \text{if } \mathbf{s}_j = 1,\ \mathbf{s}_{j-1} > 0, \text{ and } \mathbf{s}_{j+1} > 0, \\ 1 & \text{otherwise.} \end{cases}$$

(Let $\mathbf{s}_0 = \mathbf{s}_{m+1} = 1$ to avoid special cases.) Suppose $\mathbf{t}$ is the state of column $n-1$ in $G_{m,n-1}$ for subset S. Given a state vector s, suppose we add a column n to the grid, and add elements to S corresponding to the 0 entries in s. Then $\mathbf{t}$ and s will not necessarily be the state vectors for columns $n-1$ and n—new state vectors for these columns will be induced. (Note, however, that the new state vectors can differ from the old only by containing entries equal to 1 in place of entries equal to 2.) Fisher [1] defines $u(\mathbf{t}, \mathbf{s})$ to be the state vector induced on column n provided the state vector induced on column $n-1$ is at least $\mathbf{1}$; $u(\mathbf{t}, \mathbf{s})$ is undefined otherwise. Fisher [1] proves the following lemma and its corollary.

Lemma 1 $u(\mathbf{t}, \mathbf{s}) \geq \mathbf{s}$ *if and only if* $\mathbf{t} \geq p(\mathbf{s})$.

Corollary 2 $\{ \mathbf{t} \mid \exists \mathbf{s}'\, |\mathbf{s}'| = |\mathbf{s}| \wedge \mathbf{s}' \geq \mathbf{s} \wedge u(\mathbf{t}, \mathbf{s}') = \mathbf{s}' \} = \{ \mathbf{t} \mid \mathbf{t} \geq p(\mathbf{s}) \}$.

Suppose s is the state of column n, and let e be the number of endpoints of the column in S, that is, the number of 0 entries in positions 1 and m of s. The "contribution" that column n makes to $I(S) + |S_n|$ is

$$\|\mathbf{s}\|_n = \begin{cases} 4|\mathbf{s}| - e & \text{if } n = 1, \\ 5|\mathbf{s}| - e & \text{if } n > 1. \end{cases}$$

Finally, as in Fisher [1], define $\mathbf{s}' \succ \mathbf{s}$ if for some j, $\mathbf{s}'_j = \mathbf{s}_j - 1$, and $\mathbf{s}'_i = \mathbf{s}_i$ otherwise. The next theorem is the basis of the algorithm.

Theorem 3 *Define $R_{m,0}(\mathbf{s}) = R^+_{m,0}(\mathbf{s}) = 0$ if $\mathbf{s} \leq 1$, and ∞ otherwise. Then for $n \geq 1$*

$$R^+_{m,n}(\mathbf{s}) = \min(\|\mathbf{s}\|_n + R^+_{m,n-1}(p(\mathbf{s})), \min_{\mathbf{s}' \succ \mathbf{s}} R^+_{m,n}(\mathbf{s}'))$$

$$R_{m,n}(\mathbf{s}) = \min(\|\mathbf{s}\|_n - |\mathbf{s}| + R^+_{m,n-1}(p(\mathbf{s})), \min_{\mathbf{s}' \succ \mathbf{s}} R_{m,n}(\mathbf{s}')).$$

Proof. This proof is nearly identical to Fisher's corresponding proof for domination number. Also, the proofs of the two parts of this theorem are essentially identical, so we present only the first. The proof uses a very limited amount of arithmetic involving the values of $\overline{R}$ and $\overline{R}^+$, which can be ∞; to make sense of these statements we adopt the convention that $x + \infty = \infty$.

We first prove "$\leq$". If $\mathbf{s}' \succ \mathbf{s}$ then $R^+_{m,n}(\mathbf{s}) \leq R^+_{m,n}(\mathbf{s}')$ because each is a minimum over a set, and the set corresponding to $R^+_{m,n}(\mathbf{s}')$ is a subset of the one corresponding to $R^+_{m,n}(\mathbf{s})$. So $R^+_{m,n}(\mathbf{s}) \leq \min_{\mathbf{s}' \succ \mathbf{s}} R^+_{m,n}(\mathbf{s}')$.

Next note that

$$R^+_{m,n}(\mathbf{s}) = \min_{\mathbf{s}' \geq \mathbf{s}} \overline{R}^+_{m,n}(\mathbf{s}') \leq \min_{\substack{|\mathbf{s}'| = |\mathbf{s}| \\ \mathbf{s}' \geq \mathbf{s}}} \overline{R}^+_{m,n}(\mathbf{s}'),$$

because the second minimum is taken over a subset of the set used in the first minimum. Then

$$\begin{aligned} R^+_{m,n}(\mathbf{s}) \leq \min_{\substack{|\mathbf{s}'| = |\mathbf{s}| \\ \mathbf{s}' \geq \mathbf{s}}} \overline{R}^+_{m,n}(\mathbf{s}') &= \min_{\substack{|\mathbf{s}'| = |\mathbf{s}| \\ \mathbf{s}' \geq \mathbf{s}}} \left(\|\mathbf{s}'\|_n + \min_{u(\mathbf{t},\mathbf{s}') = \mathbf{s}'} \overline{R}^+_{m,n-1}(\mathbf{t}) \right) \\ &= \|\mathbf{s}\|_n + \min_{\substack{|\mathbf{s}'| = |\mathbf{s}| \\ \mathbf{s}' \geq \mathbf{s}}} \min_{u(\mathbf{t},\mathbf{s}') = \mathbf{s}'} \overline{R}^+_{m,n-1}(\mathbf{t}) \\ &= \|\mathbf{s}\|_n + R^+_{m,n-1}(p(\mathbf{s})). \end{aligned}$$

The last equality relies on Lemma 1 and its corollary.

Now we prove "$\geq$". Suppose that $R^+_{m,n}(\mathbf{s}) = \overline{R}^+_{m,n}(\mathbf{s}'')$ for some $\mathbf{s}'' > \mathbf{s}$. Then for some $\mathbf{s}' \succ \mathbf{s}$, $\mathbf{s}'' \geq \mathbf{s}'$, and so $R^+_{m,n}(\mathbf{s}) = \overline{R}^+_{m,n}(\mathbf{s}'') \geq R^+_{m,n}(\mathbf{s}')$, so $R^+_{m,n}(\mathbf{s}) \geq \min_{\mathbf{s}' \succ \mathbf{s}} R^+_{m,n}(\mathbf{s}')$.

Suppose instead that for all $\mathbf{s}' > \mathbf{s}$, $R^+_{m,n}(\mathbf{s}) < \overline{R}^+_{m,n}(\mathbf{s}')$, so $R^+_{m,n}(\mathbf{s}) = \overline{R}^+_{m,n}(\mathbf{s})$. Then there is a $\mathbf{t} \geq p(\mathbf{s})$ with $u(\mathbf{t}, \mathbf{s}) = \mathbf{s}$ and

$$R^+_{m,n}(\mathbf{s}) = \overline{R}^+_{m,n}(\mathbf{s}) = \|\mathbf{s}\|_n + \overline{R}^+_{m,n-1}(\mathbf{t}) \geq \|\mathbf{s}\|_n + R^+_{m,n-1}(p(\mathbf{s})). \blacksquare$$

We are now prepared to describe the algorithm to compute $R^+_{m,n}$ and $R_{m,n}$ for fixed m and $1 \leq n \leq N$.

Let c_m be the number of state vectors of length m. Number the state vectors from 0 to $c_m - 1$, giving vector $\mathbf{s}$ number $i_{\mathbf{s}}$, so that $\mathbf{t} \geq \mathbf{s}$ implies $i_{\mathbf{t}} \leq i_{\mathbf{s}}$. (Thus, the largest vector $\mathbf{0}$ is numbered 0, and $\mathbf{2}$ is numbered $c_m - 1$.)

1. **Initialization.** Set $R_{m,0}(\mathbf{s}) = R^+_{m,0}(\mathbf{s}) = 0$ if $\mathbf{s} \leq \mathbf{1}$, and ∞ otherwise.

2. **Iteration.** Suppose that $R_{m,n-1}(\mathbf{s})$ and $R^+_{m,n-1}(\mathbf{s})$ have been computed for all $\mathbf{s}$, and $R_{m,n}(\mathbf{t})$ and $R^+_{m,n}(\mathbf{t})$ have been computed for all $\mathbf{t}$ with $i_{\mathbf{t}} < i_{\mathbf{s}}$. Then by Theorem 3 we may set

$$R^+_{m,n}(\mathbf{s}) = \min(\|\mathbf{s}\|_n + R^+_{m,n-1}(p(\mathbf{s})), \min_{\mathbf{s}' \succ \mathbf{s}} R^+_{m,n}(\mathbf{s}'))$$
$$R_{m,n}(\mathbf{s}) = \min(\|\mathbf{s}\|_n - |\mathbf{s}| + R^+_{m,n-1}(p(\mathbf{s})), \min_{\mathbf{s}' \succ \mathbf{s}} R_{m,n}(\mathbf{s}')).$$

   since all quantities in the right hand expressions have already been computed.

3. **Redundance.** The final values of interest, namely the redundance of the grid graphs $G_{m,n}$ for fixed m and $1 \leq n \leq N$, are the values $R_{m,n}(\mathbf{1})$.

3 Computation of $R_{m,n}$ for all n

Fisher discovered that in the case of the domination number of grid graphs, the vectors corresponding to our $R_{m,n}$ eventually form an arithmetic sequence, and it seems not unreasonable to hope that the same might happen here. That is, for fixed m, we say that the sequence of vectors $R_{m,n}$ is eventually arithmetic if there exist N, p and q so that for all $n \geq N$ and all s

$$R_{m,n}(\mathrm{s}) = R_{m,n-p}(\mathrm{s}) + q.$$

If $R_{m,n}$ is eventually arithmetic then we can determine $R(G_{m,n})$ for fixed m and all n. This indeed turns out to be the case.

Since the values $R_{m,n}$ and $R^+_{m,n}$ are determined by the values of $R^+_{m,n-1}$, it is not hard to prove that if for all s, $R^+_{m,N}(\mathrm{s}) = R^+_{m,N-p}(\mathrm{s}) + q$, then for all $n > N$ and for all s $R_{m,n}(\mathrm{s}) = R_{m,n-p}(\mathrm{s}) + q$.

For fixed m, define the vector J_n by $J_n(\mathrm{s}) = R^+_{m,N-p}(\mathbf{0}) - R^+_{m,N-p}(\mathrm{s})$. The values $J_n(\mathrm{s})$ are bounded above by $5m$, so by the pigeonhole principle, there are N and p with $J_N = J_{N-p}$. With $q = R^+_{m,N}(\mathbf{0}) - R^+_{m,N-p}(\mathbf{0})$, this implies $R^+_{m,N}(\mathrm{s}) = R^+_{m,N-p}(\mathrm{s}) + q$ for all s.

Now if we run the algorithm until we find $0 < j < k$ such that for some q and all s, $R^+_{m,k}(\mathrm{s}) = R^+_{m,j}(\mathrm{s}) + q$, we will be able to write down a formula for $R(G_{m,n})$ for (at least) $n > j$. Namely, if $p = k - j$, $a = \lfloor (n-j-1)/p \rfloor$, and $b = (n-j-1) \bmod p$, then for $n > j$, $R(G_{m,n}) = aq + R(G_{m,j+1+b})$, taking $R(G_{m,j+1})$ through $R(G_{m,k})$ as constants. In practice, the formulas so obtained also work for some values of $n \leq j$.

The existence of N, after which the values of $R_{m,n}$ are arithmetic, depends on the pigeonhole principle applied to a very large number of pigeonholes. Fortunately, for $1 \leq m \leq 21$, N is quite small: always less than 128 and usually much less.

For $1 \leq m \leq 13$ we get 13 separate formulas for $R(G_{m,n})$, which are not very enlightening. For $14 \leq m \leq 21$, a single formula suffices, and we conjecture that this gives the value of $R(G_{m,n})$ for all larger m as well.

Theorem 4 *For $14 \leq m \leq 21$ and $n \geq m$, $R(G_{m,n}) = g(m,n)$ where*

$$g(m,n) = mn + \lfloor 6m/5 + 6n/5 - 1/5 \rfloor - 15 + C_{m \bmod 5, n \bmod 5}$$

and

$$C = \begin{bmatrix} 0 & 1 & 0 & 2 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 2 & 2 & 0 \\ 2 & 1 & 2 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 \end{bmatrix}.$$

4 An upper bound

For $m, n \geq 22$, we can show that $g(m,n)$ is an upper bound on the redundance. Figure 2 shows the same dominating set on a 14 by 14 grid as Figure 1, with two rectangles surrounding groups of five rows and five columns.

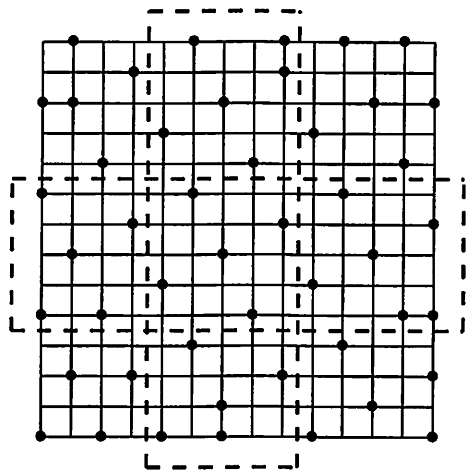

Figure 2: The 14×14 grid graph.

It is easy to check that if columns 5 through 9 are repeated as a block any number of times, then the resulting set of vertices marked by large dots is still a dominating set. If in this new grid, rows 5 through 9 are likewise repeated, the result is a dominating set in a larger grid. Suppose we first add q new copies of the columns, then r new copies of the rows; let us compute the influence.

The original set S shown in Figure 2, with $m = n = 14$, has influence $g(m,n)$. The contribution of the vertices in the vertical rectangle is $76 = 5m + 6$, so the contribution of the new copies of these columns is $5qm + 6q$. The new grid has m rows and $n' = n + 5q$ columns, with influence

$$g(m,n)+5qm+6q = m(n+5q)+\lfloor 6m/5+6(n+5q)/5-1/5\rfloor - 14 = g(m,n').$$

The contribution of the original five rows in the horizontal rectangle to the influence is $76 = 5n + 6$; the contribution of the same rows in the expanded grid is $5n + 6 + 25q = 5(n + 5q) + 6 = 5n' + 6$, since there are five points of S in the intersection of the rectangles. When these five rows are replicated r times, we get a grid with $m' = m + 5r$ rows. The new rows add $5n'r + 6r$

to the influence, giving a total of

$$g(m,n')+5n'r+6r = (m+5r)n'+\lfloor 6(m+5r)/5+6n'/5-1/5\rfloor-14 = g(m',n').$$

Thus, for any $m' > 14$, $n' > 14$, $m' \equiv 4 \pmod 5$, and $n' \equiv 4 \pmod 5$, $R(G_{m',n'}) \leq g(m',n')$. To verify this upper bound for all $m,n \geq 22$, we need to repeat the argument starting with small grid graphs representing all possible combinations of m mod 5 and n mod 5. By symmetry, it is enough to use grids with dimensions 14×15, 14×16, 14×17, 14×18, 15×15, 15×16, 15×17, 15×18, 16×16, 16×17, 16×18, 17×17, 17×18, and 18×18.

Let us list carefully what must be checked for each starting grid and its associated dominating set S:

1. There are five rows and five columns whose replication leads to a dominating set. To see that rows $i, \ldots, i+4$ replicate properly, we need only check that row $i+4$ contains a vertex of S every place that row $i-1$ does, and that row i contains a vertex of S every place that row $i+5$ does. Columns are similar.

2. The contribution of the rows to be replicated to the influence is $5n+6$, and the contribution of the columns is $5m+6$.

3. The vertices common to the rows and columns that will be replicated contain precisely 5 elements of S.

It is a bit tedious to identify the candidate rows and columns and then check conditions 1 through 3. Fortunately, it is easy to write a computer program to find the rows and columns and perform the checks. We have done this, and have in addition checked conditions 1 through 3 by hand.

The hardest part, of course, is discovering the dominating sets for the starting grid graphs. This would be daunting by hand, but a simple modification of the program used to compute redundance will produce a dominating set realizing the redundance. We did this to generate the needed grids, and in some cases tweaked the results by hand to enhance the pattern that is apparent in the resulting dominating sets. Note that since we are investigating an upper bound, it is not necessary to know that the dominating sets are optimal. Once the dominating sets and blocks of rows and columns have been found, the proof of the upper bound can be verified entirely by hand.

5 A Lower bound

It appears that the redundance exceeds the number of vertices due to "edge effects" in the grid graphs. This suggests that we might discover a good lower bound for the redundance by investigating a strip of constant width around the edge of a grid graph.

Define the *excess domination* $\gamma_e(G) = R(G) - |V(G)| = R(G) - mn$. It is not hard to see that

$$\gamma_e(G) = \min_S \sum_{v \in G} (|N[v] \cap S| \dot{-} 1),$$

taking the minimum over dominating sets S. Here $a \dot{-} b = \max(a-b, 0)$ and $N[v]$ denotes the closed neighborhood of v. If S is any subset of $V(G)$ (that is, not necessarily a dominating set), let $E_G(S) = \sum_{v \in G} (|N[v] \cap S| \dot{-} 1)$.

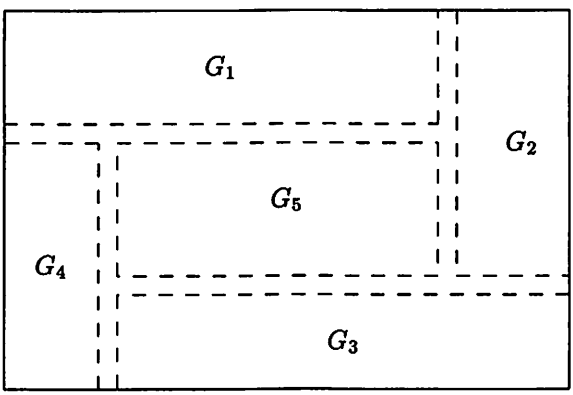

Figure 3: Partitioned grid graph.

Suppose a grid graph G is partitioned into five subgraphs as shown in Figure 3 and that S is a dominating set for G. Let $S_i = S \cap V(G_i)$. Then

$$E_G(S) \geq \sum_{i=1}^{5} E_{G_i}(S_i) \geq \sum_{i=1}^{4} E_{G_i}(S_i).$$

Note that G_i is a grid graph and that S_i is a set that dominates all the vertices of G_i except possibly the vertices in one row and one column on the boundary. Let us say that a set that dominates a grid graph G, except possibly the vertices in the first row and the last column, *almost dominates* G. Suppose $H = G_{i,j}$; what we would like to know is the value of

$$\min_S E_H(S),$$

taking the minimum over sets S that almost dominate H. If we can compute this minimum for fixed i and any j, we can apply this to G_1 through G_4 and get a lower bound on γ_e. Ideally, we want to choose i so that the resulting lower bound is close to the upper bound, i is small enough to give us a bound on $\gamma_e(G_{m,n})$ for $m, n \geq 22$, and computing the minimum is computationally feasible. Using $i = 10$ satisfies all of these criteria, giving a small constant difference between the upper and lower bounds.

We want to compute the minimum possible values of $E_{G_{i,j}}$ as we did R, namely, by incrementing j and looking for a point at which the values become arithmetic. Unfortunately, we have not been able to discover a result analogous to Theorem 3, so we resort to a similar but less efficient algorithm. (The problem is that when adding a new column to the grid, the amount by which to increment the value of E depends on exactly what column is chosen to precede it.)

An s-almost-domination of $G_{m,n}$ is a subset S of the vertices that dominates the first $n-1$ columns, except possibly vertices in the first row, and for which the state vector of the final column is $\mathbf{s}$. Let

$$E_{m,n}(\mathbf{s}) = \min_{S} E_{G_{m,n}}(S),$$

taking the minimum over all s-almost-dominations of $G_{m,n}$. Suppose that $\mathbf{t}$ is the state of column $n-1$ in $G_{m,n-1}$ for subset S, and that a column n is added, with state vector $\mathbf{s}$. Let $\hat{u}(\mathbf{t}, \mathbf{s})$ be the state vector induced on column n, provided that the state vector induced on column $n-1$ is at least $(2, 1, 1, \ldots, 1)$, that is, that column $n-1$ is dominated, except (possibly) the first vertex; $\hat{u}$ is undefined otherwise. Now

$$E_{m,n}(\mathbf{s}) = \min_{\hat{u}(\mathbf{t},\mathbf{s})=\mathbf{s}} \left(\operatorname{incr}(\mathbf{t}, \mathbf{s}) + E_{m,n-1}(\mathbf{t})\right),$$

where $\operatorname{incr}(\mathbf{t}, \mathbf{s})$ is computed as follows, assuming that $\hat{u}(\mathbf{t}, \mathbf{s}) = \mathbf{s}$ and that S is the associated almost-dominating set in $G_{m,n}$. Number the vertices in $G_{m,n}$ in the obvious way: $v_{i,j}$, $1 \leq i \leq m$, $1 \leq j \leq n$.

1. Set $i = 0$.
2. For each $k \in \{1, \ldots, m\}$ for which $\mathbf{s}_k = 0$ and $\mathbf{t}_k \leq 1$, add 1 to i.
3. For each $k \in \{1, \ldots, m\}$, let $j = |N[v_{k,n}] \cap S| \dot{-} 1$, and add i to i.
4. The value of i is $\operatorname{incr}(\mathbf{t}, \mathbf{s})$.

The algorithm is now straightforward:

1. **Initialization.** Set $E_{m,0}(\mathbf{s}) = 0$ if $\mathbf{s} = \mathbf{1}$, and ∞ otherwise.

2. **Iteration.** Suppose that $E_{m,n-1}(\mathbf{s})$ has been computed for all $\mathbf{s}$. Then set

$$E_{m,n}(\mathbf{s}) = \min_{\hat{u}(\mathbf{t},\mathbf{s})=\mathbf{s}} (\mathrm{incr}(\mathbf{t},\mathbf{s}) + E_{m,n-1}(\mathbf{t}))\,.$$

3. The final values of interest are the values $\min_{\mathbf{s}} E_{m,n}(\mathbf{s})$, taking the minimum over all $\mathbf{s}$, since we do not require that the rightmost column be dominated.

We have not been able to prove that the sequence of vectors $E_{m,n}$ becomes arithmetic, but it is easy to see that once $E_{m,N} = E_{m,N-p} + q$, $E_{m,n}$ is indeed arithmetic for $n > N$. We can thus start the algorithm and hope that we discover $E_{m,N} = E_{m,N-p}$ for small N. This does in fact happen for $m = 10$.

Performing this calculation with $m = 10$, we find that

$$\min_{\mathbf{s}} E_{10,i}(\mathbf{s}) = \lfloor (3i-4)/5 \rfloor, i \geq 7.$$

Then the desired lower bound is given by

$$\begin{aligned} E_G(S) &\geq 2\min_{\mathbf{s}} E_{10,m-10}(\mathbf{s}) + 2\min_{\mathbf{s}} E_{10,n-10}(\mathbf{s}) \\ &\geq 2\lfloor (3(m-10)-4)/5 \rfloor + 2\lfloor (3(n-10)-4)/5 \rfloor, \end{aligned}$$

for $m, n \geq 22$. Finally, combining this with the upper bound, we have the following theorem.

Theorem 5 *Let* $\ell(m,n) = mn + 2\left\lfloor \frac{3m-4}{5} \right\rfloor + 2\left\lfloor \frac{3n-4}{5} \right\rfloor - 24$. *When* $m, n \geq 22$, $\ell(m,n) \leq R(G_{m,n}) \leq g(m,n)$ *and* $g(m,n) - \ell(m,n) \leq 16$.

We conjecture that the redundance of the grid graphs is equal to the upper bound when $m, n \geq 22$.

An implementation of the Algorithm. We used three main programs to get the results presented here: a program to compute the redundance, a program to produce examples of optimal dominating sets, and a program to compute $E_{m,n}$. The second of these is a straightforward modification of the first; it keeps additional information that allows us to identify a sequence of state vectors that provide the minimum redundance.

As noted above, the third program uses a different algorithm; while similar to the first it is actually simpler to program.

A few points are worthy of discussion. The obvious way to check for the point at which the values become arithmetic is to keep the values $R^+_{m,n}(\mathrm{s})$ for all n and s. The principal drawback here is space: the amount of storage required increases rapidly with m, and becomes prohibitively large on the machines available to us when m is around 16.

A substantial reduction in storage can be achieved at what turns out to be a modest cost in running time. Instead of keeping $R^+_{m,n}(\mathrm{s})$ for all previous values of n, we keep the values for only one value of n at a time. Each time we compute a new set of values $R^+_{m,n}(\mathrm{s})$, we compare them to the values $R^+_{m,r}(\mathrm{s})$, and periodically we increase the value of r. To ensure that we eventually detect the arithmetic nature of the sequence, we must make sure that r increases and also that we check for ever larger intervals. We use powers of 2 for r so that, for example, $r = 8$ for $n = 9, \ldots, 16$, $r = 16$ for $n = 17, \ldots, 32$, and so on. Every time we check $R^+_{m,n}(\mathrm{s})$ against $R^+_{m,r}(\mathrm{s})$, we actually gain some time over the old method, which would have us compare $R^+_{m,n}(\mathrm{s})$ to (typically) all previous values $R^+_{m,i}(\mathrm{s})$, $i < n$. On the other hand, we detect the arithmetic property later, which means we compute $R^+_{m,n}(\mathrm{s})$ for more values of n, and this slows the program down. For example, $R^+_{19,31}(\mathrm{s}) - R^+_{19,26}(\mathrm{s})$ is constant, but we do not detect the constant difference until $R^+_{19,37}(\mathrm{s}) - R^+_{19,32}(\mathrm{s})$.

We can further minimize storage by reducing the storage required for a single value $R^+_{m,n}(\mathrm{s})$. Instead of allocating a 32 bit integer (an "int" in C) or even a 16 bit integer (a "short"), we allocate only 8 bits, as an "unsigned char". This means we can store values up to 255. To make this work, we normalize the values $R^+_{m,n}(\mathrm{s})$ by subtracting the minimum over all s, which is always the value $R^+_{m,n}(2)$. This also means that the check for a constant difference becomes a check for equality, which is more efficient. These two improvements in storage efficiency were used by Fisher [2].

The programs, written in C++, are available from the author; send email to `guichard@whitman.edu`.

References

[1] D. C. FISHER, *The domination number of complete grid graphs.* manuscript.

[2] ——, *The domination number of complete grid graphs.* preprint.

[3] T. W. HAYNES, S. T. HEDETNIEMI, AND P. J. SLATER, *Fundamentals of Domination in Graphs*, New York: Marcel Dekker, Inc., 1998.