Lower Bounds on the Pathwidth of some Grid-like Graphs *

JOHN ELLIS[†] AND ROBERT WARREN[⋆]

[†]*Department of Computer Science*

University of Victoria, P.O. Box 3055

Victoria, British Columbia, V8W 3P6, Canada

[⋆]*School of Information Technology and Engineering*

University of Ottawa, 800 King Edward Avenue

Ottawa, Ontario, K1N 6N5, Canada

December 2, 2006

Abstract

We present proofs of lower bounds on the node search number of some grid-like graphs including 2-dimensional grids, cylinders, tori and a variation we call "orb-webs". Node search number is equivalent to pathwidth and vertex separation, which are all important graph parameters. Since matching upper bounds are not difficult to obtain, this implies that the pathwidth of these graphs is easily computed, because the bounds are simple functions of the graph dimensions. We also show matching upper and lower bounds on the node search number of equidimensional tori which are one less than the obvious upper bound.

Keywords: pathwidth, vertex separation, node search number, grids, cylinders, tori, lower bounds.

*This work was supported by the Natural Sciences and Engineering Research Council of Canada

1 Introduction

Pathwidth is a graph parameter whose importance in determining the algorithmic complexity of width bounded instances of intractable problems is well known. It is also well known that the related decision problem is NP-complete. In [MS88] it is shown that the problem remains NP-complete even for planar graphs of degree 3.

Here we examine some grid-like graph families, including 2-dimensional grids, cylinders, tori and a variation thereof. We note that, although is it not hard to establish certain upper bounds on the pathwidth of these graphs, we do not know of proofs (except for grids, see below) that these bounds are also lower bounds. Once both lower and matching upper bounds are established and since the bounds are simple functions of the graph dimensions, it becomes easy to compute the pathwidth of these graph families.

The pathwidth of a graph is identical to its vertex separation [Kin92] and the node search number is equal to vertex separation + 1 [KP86]. Hence these concepts are equivalent. Here our proofs are worked exclusively in terms of node search number.

There is a substantial literature surrounding these concepts. Surveys can be found in [Bie91], [DPS02] and [Als04]. The lower bound for grids is implied by the lower bound for the treewidth of grids shown in [Bod98], Corollary 89. Also, the concept of a "bramble/screen" developed in [ST93] could possibly be used to obtain the other lower bounds we prove here.

2 Definitions

Although the pathwidth of a graph, which we denote $pw(G)$, is perhaps the best known and widely applied of the three parameters listed in the introduction, our proofs are worked entirely in terms of *node search number*, denoted $ns(G)$. We then use the equation $pw(G) = ns(G) - 1$ to obtain the theorems for the pathwidth. Informally, the search number of a graph is the minimum number of guards necessary to guarantee the capture of a fugitive who can move with arbitrary speed about the edges of the graph. Several versions of the game have been proposed. We are concerned here only with so called *node search number*, which is defined as follows.

A *search step* is either the placing of a guard on a node, or the removal of a guard from a node. A *search sequence* is a sequence of search steps. Initially, all the edges of the graph are *contaminated*. A contaminated edge $\{u, v\}$ is cleaned whenever guards are simultaneously occupying both u and v. A clean edge will become recontaminated if the removal of a guard results in a path without guards from a contaminated edge to the clean edge. A *search strategy* S for a graph is a search sequence that results in all edges being simultaneously clean and no guards remaining on G. The cost of a strategy, *cost(S)*, is the maximum number of guards on G after any move in S. The node search number of a graph, denoted $ns(G)$, is $\min\{cost(S) \mid S \text{ is a strategy for } G\}$. In the rest of the paper we use "search number" to mean "node search number". When we say "at step t" we will mean "at the completion of the t^{th} step".

A most useful result [LaP93], [KP86], assures us that we do not need to consider strategies that permit recontamination. No saving in guards can be gained in that way. We deduce that we need only consider strategies in which no guard on a node incident with a dirty edge is ever removed. This is because, firstly, if the guard is also incident with a clean edge, then it cannot be removed without causing recontamination and we need not consider recontaminating strategies. But if the guard is incident with nothing but dirty edges and is removed, the original placement was redundant. A strategy of no greater cost can be created by removing this pair of placement, removal moves.

We call guards on nodes incident with a dirty edge *critical* guards and the associated nodes *critical* nodes. It follows that if at some stage in a strategy all guards are critical, the next move must be a guard placement, and conversely, immediately prior to a guard placement, all nodes are critical. We will refer to this as the *criticality principle*.

The graph families we consider are variations on 2-dimensional grids. A 2-dimensional grid of height h and width w, which we call an (h, w)-grid, is the graph comprising the node set: $\{(x, y) \mid 0 \leq x < h, 0 \leq y < w\}$ and the edge set: $\{\{(u, v), (x, y)\} \mid \mid u - x \mid + \mid v - y \mid = 1\}$. Equivalently, the (h, w)-grid is the product of P_h and P_w where P_n is the simple path with n vertices. We assume in all cases that $w \geq 3$ and $h \geq 3$.

The (h, w)-cylinders are (h, w)-grids with wrap-around edges in the rows, i.e. the product of P_h and C_w where C_w is the simple cycle with w vertices. The (h, w)-tori are (h, w)-grids with wrap-around edges in both rows and columns, i.e. the product of C_h and C_w. We also consider what we call "(h, w)-orb-webs" which are a small variation on cylinders and are defined in their section.

These graphs and all the variations contain *rows*, all the nodes with the same first number in the vertex number pair, and *columns*, all the nodes with the same second number in the vertex number pair. We will say that a row or column is: *partly dirty* if some edges in that row or column are clean and some are dirty, *completely clean* if all edges are clean and *completely dirty* if all edges are dirty. A row or column will be called *almost clean* if the placement of one guard on some node in the row or column causes it to become completely clean.

3 Observations

We make a number of observations that we will use repeatedly in our proofs.

Observation 3.1 *A partly dirty column or row, with or without wrap-around edges, contains at least one critical guard, incident, respectively, with a dirty column or row edge, and it contains an unguarded node incident with a dirty edge.*

There must be a node incident with both a clean and a dirty edge. This node must be guarded. If all nodes incident with dirty edges are guarded then there are no dirty edges.

Observation 3.2 *A partly dirty column or row, with wrap-around edges contains at least two critical guards and two dirty edges, which share an unguarded node.*

There must be two nodes incident with both a clean and a dirty edge and, if there were only one dirty edge, it would have to be guarded at each end, and hence clean.

Observation 3.3 *An almost clean row, without wrap-around edges, contains $w-1$ nodes that are either incident with clean edges or guarded.*

By definition, an almost clean row can be cleaned by the placement of just one guard. Hence it cannot contain more than two dirty edges, so it contains just one or two dirty edges. If it contains just one dirty edge, it must be the first or last edge in the row and, for all $w \geq 3$, there must be $w-1$ nodes incident with clean edges.

If it contains two dirty edges then, for all $w \geq 3$, there must be two guarded nodes incident with the dirty edges, so that the one placement cleans both edges. The remaining $w-3$ nodes must be incident with clean edges.

Observation 3.4 *If there is a step in a search strategy at which there exists no completely clean column, no completely clean row and one almost clean row, say R, without wrap-around edges, then there are at least $w-1$ critical guards, each incident with a dirty column edge.*

All columns are either partly or completely dirty. By Observation 3.1, any partly dirty column contains at least one critical guard, incident with a dirty column edge. By Observation 3.3, $w-1$ nodes in row R are either incident with clean edges or guarded. So as many as $w-1$ completely dirty columns either share a guarded node with row R or share a node with R that is incident with both a clean and a dirty edge. In the latter case, that node must also be guarded and incident with a dirty column edge.

Observation 3.5 *A completely clean column that is adjacent to a partly or completely dirty column must contain at least one critical guard, and a completely clean row that is adjacent to a partly or completely dirty row must contain at least one critical guard.*

By Observation 3.1, a partly dirty column or row contains an unguarded node incident with a dirty edge, and the same is true of completely dirty columns or rows. The node in the completely clean row or column that is adjacent to this unguarded node must be guarded by a critical guard.

Observation 3.6 *If there is a step in a search strategy at which there exists, exactly one completely clean column and exactly one completely clean row, then there are at least w critical guards.*

By assumption there are $w-1$ partly or completely dirty columns. By Observation 3.1, every partly dirty column contains a critical guard. Every completely dirty column shares a node with the completely clean row and hence must contain a critical guard at the intersection. Hence there are at least $w-1$ critical guards. By Observation 3.5, the completely clean column contains at least one critical guard. Hence there are in total at least w critical guards.

Observation 3.7 *Consider a search strategy on one of our grid families with wrap-around edges. If there is a step at which r rows are either partly or completely dirty and such that each such row contains two nodes incident with some clean column edge, then there are at least $2r$ critical guards on these rows.*

By Observation 3.2, any partly dirty row contains at least two critical guards. In any completely dirty row there are two nodes each incident with both a clean column edge and a dirty row edge and hence that row contains at least two critical guards. Hence there at least $2r$ critical guards overall.

Observation 3.8 *If there is a step in a search strategy at which there exist two completely clean columns and r rows, with wrap-around edges, that are either partly or completely dirty then there are at least $2r$ critical guards on these rows.*

Since there are two clean columns every row contains two nodes incident with a clean column edge. Hence, by Observation 3.7, there at least $2r$ critical guards overall.

Observation 3.9 *If there is a step in a search strategy on an (h, w)-torus where $h < w$ at which there exists a completely clean row containing just one guard and no other row contains more than two guards then there exist at least two completely clean columns.*

Let the completely clean row be R. All other rows contain no more than two guards, hence rows adjacent to R can contain no more than two dirty edges. Let the *distance* between two rows numbered i and j where $i < j$ be $\min\{j - i, h + i - j\}$. Then rows distant d from R can contain no more than $2d$ dirty edges.

If h is odd, the maximum distance between R and any other row is $(h - 1)/2$ and so the maximum number of dirty edges in any row is $h - 1$. If h is even, the maximum distance between R and any other row is $h/2$ and so the maximum number of dirty edges in any row is h. The set of dirty edges is necessarily a subset of a pair of "pyramids" as illustrated in Figure 1. The wrap-around edges are omitted from the diagram for clarity. The base of the pyramid contains no more than h dirty edges, encompassing $h + 1$ nodes. Hence, since $h < w$, there are at least two completely clean columns.

Observation 3.10 *If at some point in a search strategy a completely dirty row is adjacent to a completely clean row then there are at least w guards in total on those two rows.*

Every edge between the two rows must be guarded on at least one end.

4 The Pathwidth of Grids

The (h, w)-grid is the simplest of the graph families that we consider. We describe an obvious search strategy and go on to show that it is optimal.

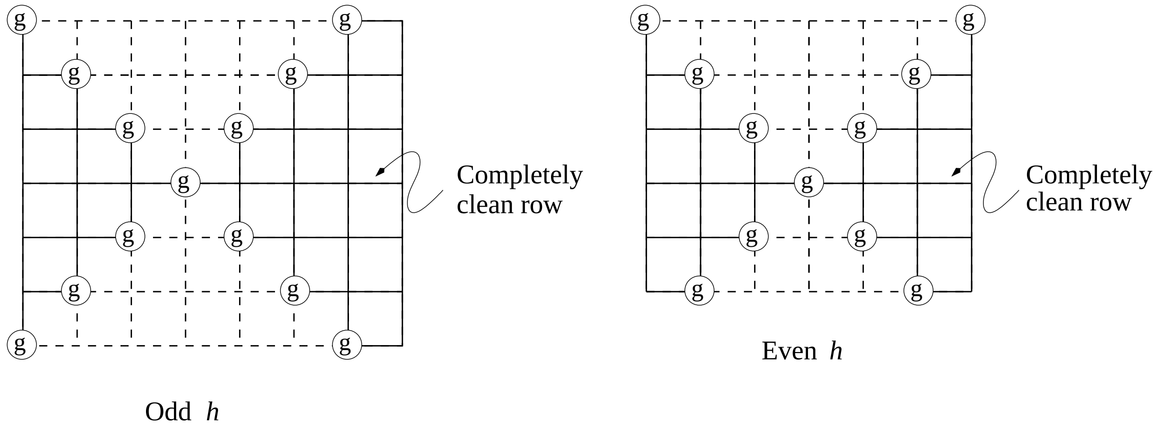

Figure 1: There exist at least two clean columns

Lemma 4.1 *If G is an (h, w)-grid then $ns(G) \leq \min(h+1, w+1)$.*

Proof

Consider filling the leftmost columns with guards and "sweeping" that guard column across the width of the grid until it reaches the rightmost column. The movement of a column of guards to the adjacent dirty column is accomplished by the use of one extra guard. The placement of the extra guard on a node in the adjacent dirty column frees one guard in the clean column which can be used on the next node in the adjacent dirty column, and so on till the adjacent column is completely clean and completely occupied by guards. Exactly $h+1$ guards are used. The same process can sweep a row across the height of the grid, using $w+1$ guards. □

Lemma 4.2 *If G is an (h, w)-grid then $ns(G) \geq \min(h+1, w+1)$.*

Proof

Let step t_c be the first step at which there exists a completely clean column and let t_r be the first step at which there exists a completely clean row, say R. Both of these steps are guard placements because they necessarily increase the number of clean edges.

Suppose $t_r \leq t_c$. Then, at step $t_r - 1$, no column or row is completely clean, but R is almost clean. Hence, by Observation 3.4, there are at least $w-1$ critical guards, each incident with a dirty column edge. If there are more than $w-1$ guards at step $t_r - 1$, we are done. So suppose there are exactly $w-1$ guards. Then one column, say H, contains no guards. Because no column is completely clean at step $t_r - 1$, H is either completely or partly dirty. By Observation 3.1, it cannot be partly dirty because it contains no guard. Hence H is completely dirty.

Suppose the guard placement at step t is to node v. Then v must be in H, because, if not, there remains a node in R with no guard and incident with both a clean row edge and a dirty column edge. The last placement cleans no edge in H, because H contains no guards. Hence, v is incident with a dirty edge at step t_r. Further, the guard placement to v cannot clean any column edge in any other column, hence all the guards in those columns are still critical. Hence there are w critical guards at step t_r. Then, by the criticality principle, step $t_r + 1$ must be another guard placement, and so there are $w + 1$ guards at step $t_r + 1$.

By identical reasoning, but exchanging rows and columns, if $t_c \leq t_r$ then there are at least $h + 1$ guards at step t_c or $t_c + 1$. Since $t_c \geq t_r$ or $t_c \leq t_r$, the lemma follows. □

From Lemmas 4.1 and 4.2 and the relation between pathwidth and search number, we derive the following theorem.

Theorem 4.1 *If G is an (h, w)-grid then $pw(G) = \min(h, w)$.*

5 The Pathwidth of Cylinders

We have defined (h, w)-cylinders to be (h, w)-grids with wrap-around edges in the rows.

Lemma 5.1 *If G is an (h, w)-cylinder then $ns(G) \leq \min(2h + 1, w + 1)$.*

Proof
Because the wrap-around edges exist in one dimension only, two different strategies are possible. The first is identical to that used on grids. Consider filling the bottom row with guards and "sweeping" that row across the height of the cylinder until it reaches the top row. The movement of one row of guards to the adjacent dirty row is accomplished by the use of one extra guard. The placement of the extra guard on a node in the adjacent dirty row frees one guard in the clean row which can be used on the next node in the adjacent dirty row, and so on till the adjacent row is completely clean and completely occupied by guards. Exactly $w + 1$ guards are used.

The second strategy starts by filling two adjacent columns with guards and continues by "sweeping" one guarded column around the width of the cylinder until it comes up against the other side of the stationary guarded column. The movement of one column of guards to the adjacent dirty column is accomplished by the use of one extra guard, in the same manner as was done for rows. Exactly $2h + 1$ guards are used.

The cheaper of the two strategies gives an upper bound. □

We complete the analysis by establishing a matching lower bound.

Lemma 5.2 *If G is an (h, w)-cylinder then $ns(G) \geq \min(2h + 1, w + 1)$.*

Proof
Let step t_c be the first step at which there exist two completely clean columns and let t_r be the first step at which there exists a completely clean row, say R. Both steps are guard placements because they necessarily clean at least one edge.

We distinguish three possibilities: either $t_r < t_c$ or $t_r = t_c$ or $t_r > t_c$.

1. Suppose $t_r < t_c$. Then suppose there exists a clean column at step t_r, so that there exists exactly one completely clean column and exactly one completely clean row. By Observation 3.6, there are then at least w critical guards. If there are more than w guards, we are done. If there are exactly w critical guards at step t_r then by the criticality principle, step $t_r + 1$ must be another guard placement, yielding at least $w + 1$ guards.

   If there is no completely clean column at step t_r then at step $t_r - 1$ there in neither a completely clean column nor a completely clean row, but R is almost clean. That case was considered in the proof of Lemma 4.2 where it was shown that at least $w + 1$ guards are used.

2. Suppose $t_r = t_c$. Then there are two completely clean columns and $h - 1$ rows, not R, that are not completely clean at step t_r. Hence, by Observation 3.8, there are at least $2h - 2$ guards on those rows. Since R was partly clean at step $t_r - 1$, by Observation 3.2, it contained at least two guards at that step. Step t_r is a placement to R which must then contain at least three guards. Hence there are at least $2h + 1$ guards at step t_r.

3. Suppose $t_r > t_c$. Then there are h rows that are not completely clean at t_c and, by Observation 3.8, there are at least $2h$ critical guards. If there are more than $2h$ guards we are done. If there are exactly $2h$ critical guards then, by the criticality principle, the next step is a guard placement. So there are $2h + 1$ guards at step $t_c + 1$. □

From Lemmas 5.1 and 5.2 and the relation between pathwidth and search number, we derive the following theorem.

Theorem 5.1 *If G is an (h, w)-cylinder then $pw(G) = \min(2h, w)$.*

6 The Pathwidth of Orb-webs

We define (h, w)-orb-webs to be cylinders plus a *hub* node, b, which is adjacent to all nodes in the top row, i.e. we add the edge set $\{(b, (0, y)) | 0 \leq y \leq w - 1\}$. See Figure 2. The hub node does have an effect on the node search number.

We first describe some obvious search strategies thus establishing some upper bounds.

Lemma 6.1 *If G is an (h, w)-orb-web then $ns(G) \leq \min(2h + 2, w + 1)$.*

Proof
We describe two search strategies. Firstly, we can start by placing a guard on the hub node and every node in the top row. This cleans all the edges incident with the hub and in the top row and uses $w + 1$ guards. We now remove the guard on the hub, leaving w guards on

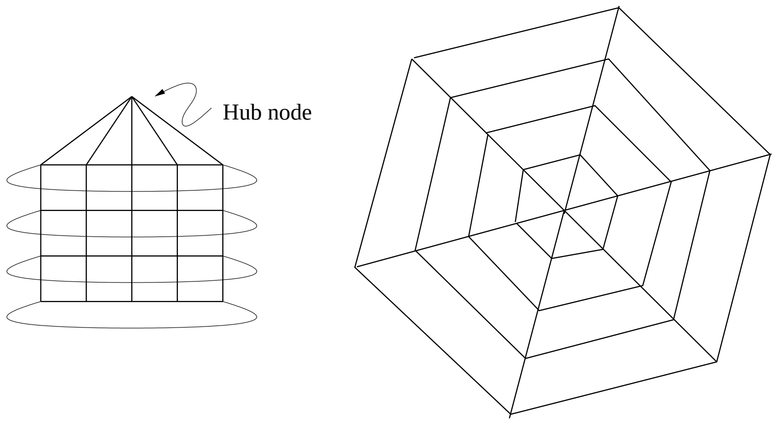

Figure 2: A $(4,5)$-orb-web

the graph. We can now sweep the row of guards through the height of the graph using just one extra guard, as was done for cylinders. Overall, $w+1$ guards are used.

Alternatively, we can place a guard on every node in two adjacent columns and on the hub node, using $2h+1$ guards. We can now sweep one of the columns of guards around the width of the graph, using just one extra guard, as was done for cylinders. Overall, $2h+2$ guards are used. The cheapest of these two strategies yields an upper bound. $\square$

We continue by showing that these bounds are also lower bounds.

Lemma 6.2 *If G is an (h,w)-orb-web then $ns(G) \geq \min(2h+2, w+1)$.*

Proof
Let step t_r be the first step at which there exists one completely clean row and let t_c be the first step at which there exist two completely clean columns and two clean edges between these columns and the hub. We distinguish three possibilities, either $t_r < t_c$ or $t_r = t_c$ or $t_r > t_c$.

1. Suppose $t_r < t_c$. This situation was analysed in the first part of the proof of Lemma 5.2 where it was shown that at least $w+1$ guards are used. That argument is unaffected by the presence of the hub.

The following observation is relevant to the two remaining cases. Suppose the hub is incident with one or more clean edges. By Observation 3.2, if the top row is partly dirty, it contains an unguarded node incident with a dirty edge row. Likewise, if the top row is completely dirty it must contain an unguarded node incident with a dirty row edge. Hence, in both of these cases, the hub is incident with a dirty edge. Since it is also incident with one or more clean edges, it must be occupied by a critical guard.

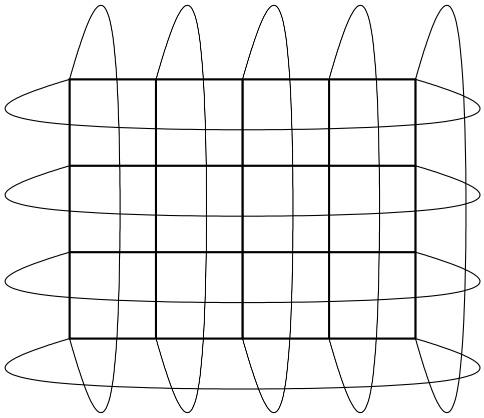

Figure 3: A (4,5)-torus

2. Suppose $t_r = t_c$. Then there are $h-1$ rows at t_r that are partly or completely dirty. Hence, by Observation 3.8, there are at least $2h-2$ guards on those rows. R is partly dirty at step t_c-1 and so, by Observation 3.2, contains at least two guards which remain at step t_c, since that step is a placement. Since all rows are partly or completely dirty at step t_c-1 and, by assumption, there is at least one clean edge between the hub and a column then, by the observation above, the hub is guarded. That guard remains at step t_c, since that step is a placement. Hence there are at least $2h+2$ guards after that placement.

3. Suppose $t_r > t_c$. Then all rows are partly or completely dirty at step t_c. Hence, by Observation 3.8, there are $2h$ critical guards on the rows at t_c. The top row is partly or completely dirty and, by assumption, the hub is incident with clean edges. Hence, by the observation above, the hub must be occupied by a critical guard. Thus there are at least $2h+1$ critical guards at step t_c.

   If there are more than $2h+1$ guards, we are done. If there are exactly $2h+1$ guards then, by the criticality principle, the next move must be another placement and there are $2h+2$ guards at step t_c+1. □

From Lemmas 6.1 and 6.2 and the relation between pathwidth and search number, we derive the following theorem.

Theorem 6.1 *If G is an (h,w)-orb-web then $pw(G) = \min(h, 2w+1)$.*

7 The Pathwidth of Tori

Firstly, we outline the obvious strategy for cleaning an (h,w)-torus.

Lemma 7.1 *If G is an (h,w)-torus then $ns(G) \leq \min(2h+1, 2w+1)$.*

Proof
Consider filling two adjacent columns with guards and "sweeping" one guard column around the width of the torus until it comes up against the other side of the stationary guarded column. The movement of one column of guards to the adjacent dirty column is accomplished by the use of one extra guard. The placement of the extra guard on a node in the adjacent dirty column frees one guard in the clean column which can be used on the next node in the adjacent dirty column, and so on till the adjacent column is completely clean and completely occupied by guards. Exactly $2h+1$ guards are used and since the torus is symmetrical with respect to rows and columns the same process can sweep a row around the height of the torus, using $2w+1$ guards. □

We have found that the obvious searching strategy for tori is not optimal for equi-dimensional, (h,h)-tori. We describe a strategy that uses just $2h$ guards.

Lemma 7.2 *If G is an (h,h)-torus then $ns(G) \leq 2h$.*

Proof
Consider first the case where $h \geq 3$ is odd. The process is defined in Table 7. Starting at some arbitrary vertex, an expanding diamond shaped component is cleaned by a sequence of guard placements forming a spiral. See Figure 4 a), where the case for $h = 7$ is illustrated. The wrap-around edges are not shown in the figure. Note that the algorithm generates coordinates that are not in the torus and which are ignored. The criteria determining whether or not a pair of coordinates is "in the torus" are: $-\lfloor h/2 \rfloor \leq x \leq \lfloor h/2 \rfloor$ and $-\lfloor h/2 \rfloor \leq y \leq \lfloor h/2 \rfloor$.

Let D_i be the clean component existing at the completion of the i^{th} iteration of the outer loop, and let the *distance* of a node v from the origin be $|x| + |y|$, where x and y are the coordinates of v. Then D_i is the subgraph of the torus induced by the vertices whose distance from the origin is $\leq i$.

For $1 \leq i \leq \lfloor h/2 \rfloor$, D_i contains $4i$ critical vertices. We obsereve that, within this range of i, during the execution of the i^{th} iteration no more than $4i+2$ guards are used. Hence, within this range, the maximum number of guards used is $4\lfloor h/2 \rfloor + 2 = 2h$, since h is odd. At the completion of that iteration, D_i includes an entire column and an entire row, for the first time. Figure 4 b) illustrates a point during the third iteration where 14 guards are used, for the case where $h = 7$. In that figure the black circles represent guards that had been placed in the previous iteration, the white circles represent guards placed during the current iteration.

We observe that, during iteration $\lfloor h/2 \rfloor + 1$, again $2h$ guards are used. Figure 4 c) illustrates a point during the fourth iteration where 14 guards are used, for the case where $h = 7$. At the completion of that iteration there remain $4\lfloor h/2 \rfloor$ critical vertices, but only one extra guard is needed during the next iteration. For all subsequent iterations the required number of guards decreases because the number of critical nodes in D_i decreases.

We use the same stategy for the case where h is even with just one change. The outer loop will now run up to $i = h$, rather than $h-1$. The criteria determining whether or not a

```
procedure SEARCH-TORUS (h : integer);
    {x and y are x and y-coordinates relative to the origin.}

    Choose a vertex arbitrarily to be the origin, location (0,0);
    Place a guard at location (0,0);

    for i ← 1 to h − 1 do
        {Initialise at the right apex and run through each side separately}
        x ← i; y ← 0;
        for j ← 1 to i do {the bottom right side}
            if location (x, y) is in the torus
            then place a guard at location (x, y); Remove any redundant guards fi;
            xc ← x - 1; y ← y - 1;
        for j ← 1 to i do {the bottom left side}
            if location (x, y) is in the torus
            then place a guard at location (x, y); Remove any redundant guards fi;
            xc ← x - 1; y ← y + 1;
        for j ← 1 to i do {the top left side}
            if location (x, y) is in the torus
            then place a guard at location (x, y); Remove any redundant guards fi;
            xc ← x + 1; y ← y + 1;
        for j ← 1 to i do {the top right side}
            if location (x, y) is in the torus
            then place a guard at location (x, y); Remove any redundant guards fi;
            x ← x + 1; y ← y - 1;
```

Table 1: Searching an equidimensional torus - h odd.

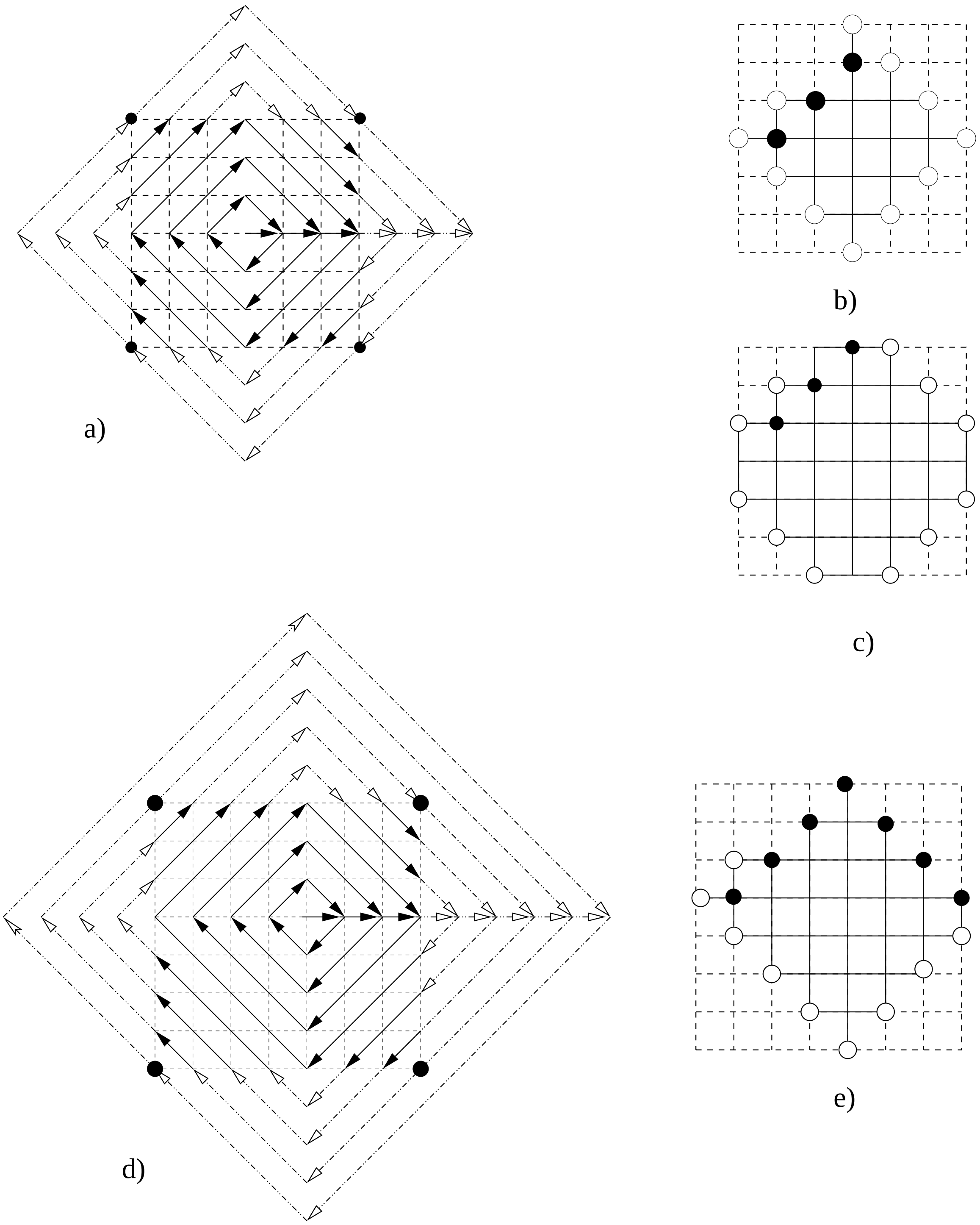

Figure 4: The searching process for equidimensional tori.

pair of coordinates is "in the torus" are: $-h/2\rfloor \leq x \leq h/2-1$ and $-h/2y \leq h/2-1$. For this h even case, we note that it is essential to begin each iteration at the right apex, i.e. location {1,0} at the first iteration, because of the asymmetrical situation. The process is illustrated in Figure 4 d) for the case where $h = 8$.

We observe that during the iteration where $i = h/2$, $2h$ guards are used. All the earlier iterations use less guards. Figure 4 e) illustrates a point during the fourth iteration where 16 guards are used, for the case where $h = 8$. The next iteration also uses $2h$ guards. All remaining iterations use less. We finish by showing that these upper bounds are also optimal.

Lemma 7.3 *If G is an (h,w)-torus then if $h = w$ then $ns(G) \geq 2h$. If $h \neq w$ then $ns(G) \geq \min(2h+1, 2w+1)$.*

Proof

Let step t_c be the first step at which there exist two completely clean columns and let t_r be the first step at which there exist two completely clean rows. Each of these steps must be a guard placement, since they necessarily clean at least one edge. We distinguish three possibilities, either $t_c < t_r$ or $t_c = t_r$ or $t_c > t_r$.

1. Suppose $t_c < t_r$. Then at step t_c there are exactly two completely clean columns and either exactly one completely clean row or no completely clean row.

   If there is no completely clean row, we have exactly the conditions analysed under item 3 of the proof of Lemma 5.2 where it was shown that at least $2h+1$ guards are used.

   If there is a completely clean row, say R, consider the situation at $t_c - 1$. If R is not completely clean at $t_c - 1$, there is one completely clean column, one almost clean column and one almost clean row, R. Then the placement at t_c cleans R and the second column. All rows except R are partly or completely dirty and share two nodes in the clean columns that are adjacent with a clean edge. Hence, by Observation 3.7, they contain at least two guards. R is partly dirty at $t_c - 1$ and so, by Observation 2, contains at least two guards. Hence there are at least $2h$ guards at step $t_c - 1$ and then at least $2h+1$ guards at step t_c.

   One case remains, namely, at $t_c - 1$, there is one completely clean column, one almost clean column and one completely clean row, R. Let the node, say v, that receives the guard placement at step t_c be in row R'. We note that R and R' must be distinct, else v is unguarded and adjacent to both clean and dirty edges at step $t_c - 1$. Since R is the only completely clean row, R' is either partly or completely dirty. R is adjacent to completely or partly dirty rows and so, by Observation 3.5, contains at least one critical guard.

   - Suppose R' is partly dirty. Then, by Observation 2, it contains at least two critical guards. Likewise, by Observation 3.7, all other rows except R, contain at least two critical guards. Hence there are at least $2h-1$ guards at $t_c - 1$ and at least $2h$ guards at t_c.

If there are exactly $2h$, then there is only one guard on R at $t_c - 1$. Then, by Observation 3.9 and since there is only one completely clean column, $h \geq w$. If $h > w$, $2h > 2w + 1$, which is one of the already established upper bounds. So in that case the corresponding search strategy would be not optimal. Hence it must be that $h = w$.

- Suppose R' is completely dirty. Then, because it shares a node incident with a clean edge in the completely clean column, it contains at least one critical guard. Hence there are at least $2h - 2$ critical guards at step $t_c - 1$. To complete the argument we consider the two possibilities, either there is one guard on R or more.

  If R contains just one guard then there are at least $2h - 2$ critical guards. In which case, since R' is completely dirty and contains just one guard, the guard placement to v at t_c leaves that guard critical. All other guards are critical because they are adjacent to dirty row edges, hence that placement does not change their criticality. Hence there are at least $2h - 1$ critical guards at step t_c and, by the criticality principle, the next step is another placement, leaving $2h$ guards. But, if R contains just one guard then, by Observation 3.9 and since there is only one completely clean column, $h \geq w$. If $h > w$, $2h > 2w + 1$, which is one of the already established upper bounds. So in that case the corresponding search strategy would be not optimal. Hence it must be that $h = w$.

  Suppose R contains two guards at $t_c - 1$. Then there are at least $2h - 1$ guards. By the criticality principle, they are all critical. As just argued, the guard placed on v at t_c is critical and leaves the other guard on R' critical. All guards other than those on R and R' are adjacent to dirty row edges which is not changed by the placement to v, and so they also remain critical.

  But the guards on R are critical at $t_c - 1$ because they are incident with dirty column edges, since the row edges are all clean. So if R and R' are adjacent, then we need another argument. In that case, by Observation 3.10, R and R' contain between them at least w guards. We need only consider the case where $w = 3$, else we immediately have at least $2h + 1$ guards at t_c. By definition, $h \geq 3$. If $h = w = 3$ we have five guards at $t_c - 1$ and six, i.e., $2h$, at t_c. If $h > w$ then $2h > 2w + 1$ and the corresponding strategy is not optimal, since an upper bound of $2w + 1$ has already been established.

In summary, for the case where $t_c < t_r$, we have established a lower bound of $2h$ on some strategies that can only be optimal if $h = w$, and for all others we have a lower bound of $2h + 1$.

2. Suppose $t_c = t_r$. Then the placement at step t_c cleans both a column and a row. Hence at step $t_c - 1$ there exist one completely clean column, one completely clean row, say R, one almost clean column and one almost clean row, say R'. The almost clean row and column share an unguarded node incident with both two dirty column edges and two dirty row edges. This node receives the guard at step t_c.

At $t_c - 1$, all the rows other than R are partly or completely dirty. All rows other than R' contain two nodes incident with clean column edges. Hence, by Observation 3.7, there are at least two guards in each such row and at least $2h - 4$ guards in total on these rows. R' is partly dirty at step $t_c - 1$ and so, by Observation 3.2, it contains at least two guards. R is completely clean and adjacent to a partly or completely dirty row. Hence, by Observation 3.5, it contains at least one guard. Hence, in total, there are at least $2h - 1$ guards at step $t_c - 1$.

If there are exactly $2h - 1$ guards at $t_c - 1$, then R contains just one guard and all other rows have two guards. Hence, by Observation 3.9 and the fact that there is only one completely clean column, $h \geq w$. If $h > w$ then $2h > 2w + 1$. There are exactly $2h$ guards at t_c and so then this particular search strategy is not optimal. Hence the $2h$ bound holds only for the case where $h = w$.

In summary, we have shown that, in the cases where $t_c = t_w$, we have established a lower bound of $2h$ on some strategies that can only be optimal if $h = w$, and for all others we have a lower bound of $2h + 1$.

3. Suppose $t_c > t_r$. Because the torus is symmetrical with respect to rows and columns, we can exchange rows and columns in part 1 of the above argument and show that if $w \neq h$ at least $2w + 1$ guards are used but if $h = w$ only the weaker lower bound of $2h = 2w$ holds. □

From lemmas 7.1, 7.2 and 7.3 and the relation between pathwidth and search number, we derive the following theorem.

Theorem 7.1 *If G is an (h, w)-torus where $h \neq w$ then $pw(G) = \min(2h, 2w)$. If $h = w$ then $pw(G) = 2h - 1$.*

8 Conclusions

We have presented unobvious proofs that some fairly obvious upper bounds on the pathwidth of some standard grid-like graphs are indeed also lower bounds. As a consequence, we now know that the pathwidth of these kinds of graph can be computed easily from their dimensions. We also discovered that the pathwidth of equidimensional tori is one less than the obvious upper bound.

These proof techniques have been used on a 3-regular version of cylinders, called "walls". The proof there is too long to be presented here but can be found in [War05]. These proof techniques might also be useful on other grid-like graph families. One such could be a 3-regular version of the torus, see Figure 5. Another might be 3-dimensional grids.

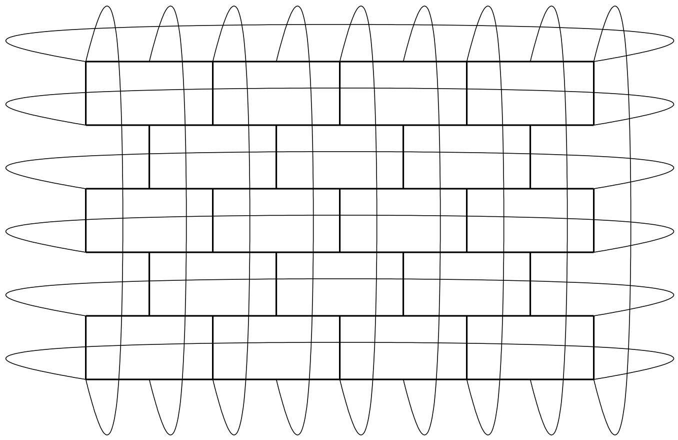

Figure 5: A (6,9)-3-regular torus

References

[Als04] B. Alspach. Searching and sweeping graphs: A brief survey. *Le Matematiche*, 59:5–37, 2004.

[Bie91] D. Bienstock. Graph searching, path-width, tree-width and related problems (a survey). *DIMACS Series in Discrete Mathematics and Theoretical Computer Sciience*, 5:33–49, 1991.

[Bod98] Hans L. Bodlaender. A partial k-arboretum of graphs with bounded treewidth. *Theor. Comput. Sci*, 209(1-2):1–45, 1998.

[DPS02] Josep Díaz, Jordi Petit, and Maria J. Serna. A survey of graph layout problems. *ACM Computing Surveys*, 34(3):313–356, 2002.

[Kin92] Nancy G. Kinnersley. The vertex separation number of a graph equals its path-width. *Information Processing Letters*, 42(6):345–350, 1992.

[KP86] L. M. Kirousis and C. H. Papadimitriou. Searching and pebbling. *Theoretical Computer Science*, 47(2):205–218, 1986.

[LaP93] A. LaPaugh. Recontamination does not help to search a graph. *Journal of the Association for Computing Machinery*, 40(2):224–245, 1993.

[MS88] B. Monien and I. H. Sudborough. Min Cut is NP-complete for edge weighted trees. *Theoretical Computer Science*, 58(1-3):209–229, 1988.

[ST93] Seymour and Thomas. Graph searching and a min-max theorem for tree-width. *JCTB: Journal of Combinatorial Theory, Series B*, 58, 1993.

[War05] R. B. Warren. Lower bounds for node search number on grid-like graphs. Master's thesis, University of Victoria, Victoria, B.C., Canada, 2005.