Centrum voor Wiskunde en Informatica
Centre for Mathematics and Computer Science

E. Kranakis, D.D.M. Krizanc, L.G.L.T. Meertens

Placing mirrors in grids

Computer Science/Department of Algorithmics & Architecture Report CS-R8906 March

The Centre for Mathematics and Computer Science is a research institute of the Stichting Mathematisch Centrum, which was founded on February 11, 1946, as a nonprofit institution aiming at the promotion of mathematics, computer science, and their applications. It is sponsored by the Dutch Government through the Netherlands Organization for the Advancement of Research (N.W.O.).

PLACING MIRRORS IN GRIDS

Evangelos Kranakis
(eva@cwi.nl)

Danny Krizanc
(krizanc@cwi.nl)

Lambert Meertens
(lambert@cwi.nl)

Centre for Mathematics and Computer Science
P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

ABSTRACT

We study instances of the following "Mirror Placement" problem on a (multidimensional) grid. Namely, we are given a light source S to be located at a node of the grid which is emitting a light beam in a single direction (e.g. a laser). We want to determine what is the minimum number of mirrors that must be placed on individual nodes of the grid in such a way that the light beam emanating from the source S will eventually "hit" all the vertices of the grid by traversing only edges of the grid. In this paper we develop an asymptotically optimal algorithm for placing mirrors on the vertices of complete multidimensional grids and analyze the worst-case behavior of any possible mirror-placement algorithm.

1980 Mathematics Subject Classification: 68R05, 68R10
CR Categories: F.2.2, G.2.2
Key Words and Phrases: Face-peeling algorithm, Multidimensional-Grid, Mirrors, Synective partition.
Note: This paper will be submitted for publication elsewhere.

1. Introduction

The present paper is concerned with the study of instances of the following "Mirror Placement" problem on multidimensional grids. We are given a light source S to be located at a node of the grid G and which is emitting a light beam in a single direction (e.g. a laser). We want to determine what is the minimum number of mirrors that must be placed on individual nodes of the grid in such a way that the light beam emanating from the source S will eventually "hit" all the vertices by traversing only edges of the grid. Here of course we assume that the standard law of reflection holds: "angle of reflection" = "angle of incidence". We call this number $s(G)$.

There is an equivalent geometric interpretation of the "Mirror Placement" problem that will turn out to be very well suited to our subsequent analysis. Call a walk P **synective** if it traverses all vertices of the grid. To every synective walk $P = v_0, {}_1, \ldots, v_k$, we associate a "partition" $L_1, L_2, \ldots, L_s$ consisting of straight lines in the following way. Start with vertex v_1 and determine the largest index $r_1 \leq k$ such that the edges $(v_1, v_2), (v_2, v_3), \ldots (v_{r_1-1}, v_{r_1})$ form a straight line; call this line L_1. Next, start with vertex v_{r_1} and determine the largest index $r_1 < r_2 \leq k$ such that the

Report CS-R8906
Centre for Mathematics and Computer Science
P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

edges $(v_{r_1}, v_{r_1+1}), (v_{r_1+1}, v_{r_1+2}), \ldots, (v_{r_2-1}, v_{r_2})$ form a straight line; call this line L_2. Continue in this fashion until eventually all the vertices of P are exhausted. This determines a unique "partition" of the given walk P into straight lines $L_1, L_2, \ldots, L_s$, which we will also call the **synective partition** of P, and the number $s := s(P)$ the **synective number** of the walk P. It is not hard to see that in fact $s(P)$ is exactly the number of times one must change direction moving along P in order to traverse all the vertices of P. We can now easily relate the synective numbers with the previously mentioned "Mirror Placement" problem. Assuming the above notation, it is clearly possible to place a mirror at each of the nodes $v_{r_2}, \ldots, v_{r_s}$ and the light source at node v_{r_1} in such a way that for each $1 \le i < s$ the incidence light beam moves along the straight line L_i while the reflecting light beam along L_{i+1}. These last observations make it clear that for graphs G as above,

$$s(P) = \min\{s(P) : P \text{ is a walk traversing all vertices of } G\}.$$

The above discussion and terminology will turn out to be extremely useful in our subsequent analysis of the "Mirror Placement" problem.

The present paper studies the "Mirror Placement" problem for a specific class of graphs, the complete multidimensional grids. To be more specific, the complete d-dimensional grid of size n, denoted G_n^d, is the graph with set of vertices

$$V = \{v := (v_1, v_2, \ldots, v_d) : 1 \le v_i \le n, \text{ for } i = 1, \ldots, d\},$$

and edge set

$$E = \{(u, v) : \sum_{i=1}^{d} |u_i - v_i| = 1\}.$$

In particular, we estimate the value of the quantity $s(G_n^d)$, for $d \ge 2, n \ge 1$. A straightforward estimate is

$$\frac{n^d - 1}{n - 1} \le s(G_n^d) \le n \cdot s(G_n^{d-1}) + n - 1.$$

Indeed, to see the upper bound notice that the the d-dimensional grid G_n^d can be thought of as n copies of the $(d-1)$-dimensional grid G_n^{d-1} joined by lines along the dth dimension. Thus if we "join together" n solutions of the "Mirror Placement" problem for the $d-1$-dimensional grid we obtain a solution of the "Mirror Placement" problem for the d-dimensional grid which satisfies the above upper bound. The lower bound is also easy. Let P be a synective walk of G_n^d with synective partition

$$L_1, L_2, \ldots, L_s,$$

such that $s = s(P)$. Clearly each line L_i contains at most n vertices of the grid. We count the number of vertices in each line L_i as we move along the trail P and obtain that

$$n + (s - 1)(n - 1) \ge n^d,$$

which implies the lower bound $(n^d - 1)/(n - 1)$. As a matter of fact it is easy to see that the only way equality would hold is if all the lines of the synective partition of a given walk are of length $n-1$. But this is impossible unless $n = 2$. It should be pointed out that for the case $n = 2$ one easily obtains that $s(G_2^3) = 7$. More generally, for the d-dimensional grid G_2^d we can show easily that $s(G_2^d) = 2^d - 1$. The main result of the paper is to develop and analyze the face-peeling algorithm which implies that the actual value of $s(G_n^d)$ satisfies much stronger upper and lower bounds.

At this point it is worth mentioning that the "Mirror Placement" problem, although related, is different from the well-known "Art Gallery" problem, first proposed by Klee [O'Rourke], in which we want to determine the minimum number of watchmen (watchmen are not allowed to move but they can see in all directions) needed so that every point in the gallery is seen by at least one watchman at any time. For example, in the art-gallery problem and for the case of the complete d-dimensional grid considered above, a guard must be located in every line segment of the grid. It is therefore not difficult to see that in this case, exactly n^{d-1} watchmen are necessary and sufficient [Ntafos, O'Rourke]. Our problem is also related to the well-known n-queens problem: what is the minimum number of queens which can be placed on an $n \times n$ chessboard so that no queen is guarding any other queen [Berge, Guy], as well as Riordan's "non-attacking rooks" problem: in how many ways can k non-attacking rooks be placed on a given side of the main diagonal of an $n \times n$ chessboard [Knuth]?

Here is an outline of the contents of the paper. In section 2 we describe our main algorithm, the so-called face-peeling algorithm, for placing mirrors on the vertices of the d-dimensional grid. To facilitate understanding and in order to clarify the main ideas of our algorithm we give the construction in different steps starting from dimension 2, next proceeding with dimension 3, and finally handling the general case $d \geq 4$. In section 3 we proceed with an analysis of the complexity of the algorithm.

2. The Face-Peeling Algorithm

In this section we give a complete intuitive description of the face-peeling algorithm. We begin with the simple case $d = 2$.

2.1. Two Dimensional Grids

Theorem 2.1.

For all $n \geq 2$, $s(G_n^2) = 2n - 1$.

Proof. To prove $s(G_n^2) \leq 2n - 1$ consider the walk of figure 1.

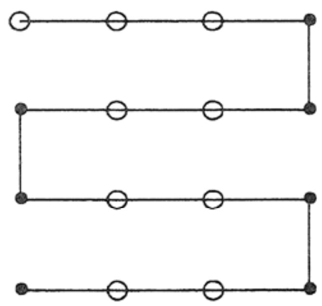

Figure 1. Establishing the upper bound $s(G_n^2) \leq 2n - 1$.

It remains to prove that $s(G_n^2) \geq 2n - 1$. Put $s = s(G_n^2)$, let P be a synective walk of G_n^2, with $s = s(P)$ and let $L_1, L_2, \ldots, L_s$ be the synective partition of P. Let h (respectively, v) be the number of horizontal (respectively, vertical) L_i's. Clearly, $s = h + v$. By definition of synective partitions, for all $i < s$, if L_i is horizontal (respectively, vertical) then L_{i+1} is vertical (respectively, horizontal). Consequently,

$$|h - v| \leq 1. \tag{1}$$

Assume that $h \leq n - 1$. This means that there is a horizontal line, say L, of the grid G_n^2 which is not traversed by any of the horizontal L_i's. Consequently, the n vertices of L must be traversed by n-many vertical L_i's. This implies that $v \geq n$. It follows from inequality (1) that $h = n - 1$ and $v = n$. A symmetric reasoning shows that if $v \leq n - 1$ then $v = n - 1$ and $h = n$. In either case we conclude that if $v + h \leq 2n - 1$ then $s = 2n - 1$. Thus always $s \geq 2n - 1$, as desired. This completes the proof of the theorem in the case of the complete two-dimensional grid. •

It is easy to see that the same argument will work for the $m \times n$-grid.

Corollary.

Exactly $2 \cdot \min(m,n) - 1$ mirrors are necessary and sufficient in order to solve the "Mirror Placement" problem for the $m \times n$ grid.

This simple observation will be used extensively in the later upper-bound arguments.

2.2. Three-Dimensional Grids

The edge-peeling algorithm can be described in the following way. Traverse the bottom horizontal plane grid by moving on its periphery from the outside to the inside and covering each time all of the corresponding vertices. The idea for doing this is depicted in figure 2.

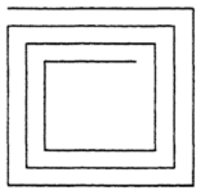

Figure 2. Traversing the vertices of the horizontal plane grids.

Proceed this way until you cover vertices of the plane grid up to a depth of $[n/4]$ vertices. This leaves an $\lceil n/2 \rceil \times \lceil n/2 \rceil$ square-grid in the middle whose vertices must be covered. At this point finish with this plane, draw a vertical line (in order to get connected with the next horizontal plane) and start moving along this new horizontal plane grid, covering its vertices in a similar way, except that now you move from the inside to the outside. When you finish traversing its outermost vertices, draw a vertical line and move to the next plane grid, and so on. Proceed this way until you cover the top horizontal plane.

At the end of traversing the top plane grid you are left with a parallelepiped grid of dimensions $\lceil n/2 \rceil \times \lceil n/2 \rceil \times n$ standing in the middle of the three-dimensional grid G_n^3 and whose vertices must be traversed. This we do just like in figure 1 traversing its vertices with vertical lines from the top to bottom plane. (Figure 3 depicts such a trail for the three-dimensional 4×4×4 grid.)

To be more exact we traverse the parallelepiped in the following way. We think of it as consisting of $\lceil n/2 \rceil$-many $\lceil n/2 \rceil \times n$ plane grids each parallel to the yz-plane. Using the corollary to theorem 2.1 we can see that we need exactly $n - 1$ straight lines to traverse each of these planes. This completes the description of the algorithm in the case of three dimensional grids.

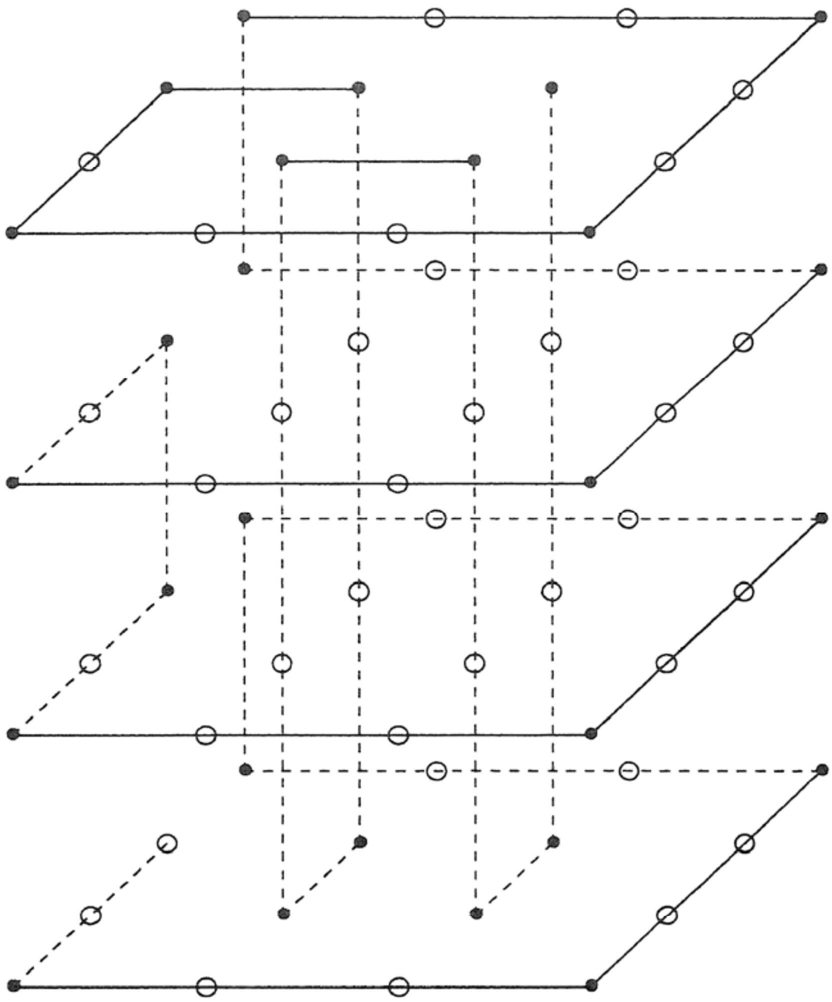

Figure 3. A walk P traversing G_4^3 with $s(P) = 27$.

2.3. d-Dimensional Grids

The d-dimensional face-peeling algorithm can in fact be considered as $(d-2)$-many iterations of the three-dimensional face-peeling algorithm. The idea is to apply the above three-dimensional algorithm to each triple (x_i, x_{i+1}, x_{i+2}) of components of the d-dimensional grid, for $i = 1,2,\ldots,d-2$. In the ith iteration, the variables (x_i, x_{i+1}, x_{i+2}) play the role of the variables (x, y, z) in the three-dimensional face-peeling algorithm. This will give a sequence of d-dimensional rectangular grids, $G_1 = G_n^d, G_2, \ldots, G_{d-2}$, in such a way that the ith iteration of the algorithm transforms G_i into G_{i+1}, but by only affecting the components x_i and x_{i+1} of G_i. The final grid G_{d-2} resulting after $d-2$ iterations of this algorithm can now be traversed in an "efficient" way by a synective walk. (It will be shown in the next section how efficient this method is).

It remains to describe more formally the ith iteration of the face-peeling algorithm. We show how to transform G_i into G_{i+1}. Let $G = G_i$ be a complete d-dimensional rectangular grid of dimensions $(a_1 n) \times \cdots \times (a_d n)$, where $1 \geq a_1, \ldots, a_d > 0$, $i \leq d-2$ and let $0 < \delta < 1$. The ith iteration of the face-peeling algorithm gives a rectangular grid $H = G_{i+1}$ of dimensions

$(b_1 n)\times \cdots \times(b_d n)$ such that $b_k = a_k$, for $k \le i-1$ or $k > i+1$. The values b_i, b_{i+1} are determined as follows. Consider the four faces parallel to the x_{i+2}-axis covering the outside part of the grid. Peel these faces (as in the case of the three-dimensional algorithm) and stretch them like a rectangle on the 2-dimensional plane. Again peel the outside faces (which are parallel to the x_{i+2}-axis) of this new grid and stretch them adjacent to the previous rectangle. Continue peeling "outermost" faces up to a depth δn until you are left with the rectangular grid H, where $b_i = a_i - 2\delta$ and $b_{i+1} = a_{i+1} - 2\delta$. Now traverse the resulting rectangle, just like in the case of the two-dimensional grid described above, and then bend the rectangle at the appropriate points in order to bring it back to its original 3-dimensional shape. This gives a rather intuitive description of the algorithm.

To sum up, our algorithm starting from the complete grid G_n^d generates a sequence

$$G_1, G_2, \ldots, G_{d-2}$$

of rectangular grids. By summing the "cost" of mirror-placement in each of these iterations we will obtain an efficient upper bound on the value of $s(G_n^d)$.

3. Analysis of the Face-Peeling Algorithm

Our analysis of the algorithm consists of two parts, namely determining both an upper bound and a lower bound for the quantity $s(G_n^d)$. The upper bound will be simply a careful analysis of the cost of the face-peeling algorithm. The lower bound proof however is more difficult and will be geometrical in nature. Moreover, to facilitate understanding we will carry out this analysis first in the three-dimensional case. We will later indicate all the necessary changes in order to extend this argument to d-dimensional grids.

3.1. Three Dimensional Grids

Theorem 3.1. There is a constant $c > 1$ such that for all $n \ge 3$,

$$c \cdot n^2 \le s_1(G_n^3) \le \frac{3}{2} \cdot (n^2 + n) - 1$$

Proof of the upper bound.

To count the number of straight-line changes required think of the three-dimensional grid G_n^3 as n horizontal copies of the two-dimensional grid G_n^2 joined by vertical lines. Now the edge-peeling algorithm given in the previous section traverses the bottom horizontal plane grid by moving on its periphery from the outside to the inside and covering each time all of the corresponding vertices. Proceeding this way you cover vertices of the plane grid up to a depth of $[n/4]$ vertices. This leaves an $\lceil n/2 \rceil \times \lceil n/2 \rceil$ square-grid in the middle whose vertices must be covered. At this point we finished with this plane, drew a vertical line (in order to get connected with the next horizontal plane) and started moving along this new horizontal plane grid, covering its vertices in a similar way, except that now you move from the inside to the outside. After finishing with the outermost vertices, we drew a vertical line and moved to the next plane grid, and so on. Proceed this way until you cover the top horizontal plane. The number of straight lines traversed in each plane is $4[n/4]$, giving a total of $4[n/4]n$ straight lines lying on these planes. To move from plane to plane we need $n-1$ straight lines just for making the connections. It follows that the total number of straight lines used is

$$4[n/4]n + n - 1. \qquad (1)$$

At the end of traversing the top plane grid we were left with a parallelepiped grid of dimensions $\lceil n/2 \rceil \times \lceil n/2 \rceil \times n$ standing in the middle of the three-dimensional grid G_n^3 and whose vertices must be traversed. This we did just like in figure 1 traversing its vertices with vertical lines from the top to bottom plane. In traversing the parallelepiped we think of it as consisting of $\lceil n/2 \rceil$ -many $\lceil n/2 \rceil \times n$ plane grids each parallel to the yz-plane. Using the result of theorem 2.1 we can see that we need exactly $n-1$ straight lines to traverse each of these planes. The total number of straight lines used in this case is $\lceil n/2 \rceil (n-1)$ for straight lines lying on the planes concerned and $\lceil n/2 \rceil - 1$ for making the plane-to-plane connections, i.e. a total of

$$n \lceil n/2 \rceil - 1 \tag{2}$$

straight lines. Summing the number of straight lines used in (1) and (2) above plus 1 (because one additional straight-line is needed when one moves from the first type of traversing to the second type) we obtain the desired result.

Optimal Choice of Depth in the Face-Peeling Algorithm.

Next we prove that in fact the optimal behavior of the peeling algorithm is obtained when the size of the remaining, middle grid is $(n/2)\times(n/2)\times n$. Indeed, suppose that we proceed covering vertices of the horizontal planes constituting G_n^3 up to a depth of x-many vertices. This leaves a grid in the middle of dimensions $(n-2x)\times(n-2x)\times n$. Using the previous counting method we obtain that

$$s(G_n^3) \leq 4xn + n - 1 + (2(n-2x)-1)(n-2x) + n - 2x.$$

If we simplify the right-hand side of the above inequality we obtain

$$s(G_n^3) \leq 2(n-2x)^2 + 4xn + n - 1.$$

Differenciating the right-hand side we obtain that the optimal value is obtained for $x = n/4$, which proves the optimality of the choice of depth in the face-peeling algorithm described above.

Proof of the lower bound.

Let P be a synective trail of G_n^3 with synective partition $L_1, L_2, \ldots, L_s$, such that $s = s(P)$. For each k let s_k (respectively, $\overline{s}_k$) be the number of lines in the above synective partition of length exactly (respectively, $\leq$) k. It is then clear that

$$s = s_{n-1} + s_{n-2} + \cdots + s_2 + s_1,$$

$$s = s_{n-1} + s_{n-2} + \cdots + s_{n-k} + \overline{s}_{n-k-1},$$

for each k. Hence, counting the number of lines of corresponding lengths, replacing s_{n-1} with the quantity $s - s_{n-2} - \cdots - s_{n-k} - \overline{s}_{n-k-1}$ and simplifying we obtain that for each k,

$$\begin{aligned} n^3 - 1 &\leq (n-1)s_{n-1} + (n-2)s_{n-2} + \cdots + (n-k)s_{n-k} + (n-k-1)\overline{s}_{n-k-1} \\ &= (n-1)s - s_{n-2} - 2s_{n-3} - \cdots - (k-1)s_{n-k} - k\overline{s}_{n-k-1}. \end{aligned}$$

Dividing through by $n-1$ and simplifying we obtain that for each $k = 1,2,\ldots,n-1$,

$$\frac{n^3-1}{n-1} + \frac{s_{n-2} + 2s_{n-3} + \cdots + (k-1)s_{n-k} + k\overline{s}_{n-k-1}}{n-1} \leq s.$$

In particular, for $k = n-2$ we obtain that

$$\frac{n^3-1}{n-1} + \frac{s_{n-2} + 2s_{n-3} + \cdots + (n-3)s_2 + (n-2)s_1}{n-1} \leq s.$$

This last inequality is equivalent to

$$\frac{n^3 - 1}{n - 1} + \frac{\bar{s}_{n-2} + \bar{s}_{n-3} + \cdots + \bar{s}_2 + \bar{s}_1}{n - 1} \leq s. \tag{3}$$

So now we concentrate on getting a lower bound for $\bar{s}_{n-2} + \bar{s}_{n-3} + \cdots + \bar{s}_2 + \bar{s}_1$. The idea for doing this is the following. Each of the straight-lines constituting the synective partition of the given trail is parallel to one of the main axis: x, y, z. It follows that there exists an axis, say z, such that at least $s/3$-many of these lines are parallel to the z-axis. Now consider the plane grid G_n^2 lying on the x,y-plane. Draw within this grid a new co-centric grid Δ_k with side $n - 2k$ and edges parallel to those of G_n^3 (see figure 4).

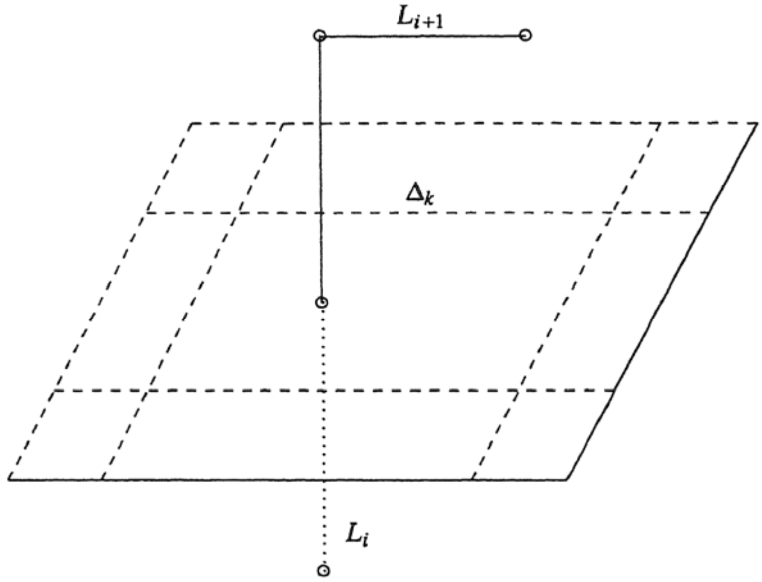

Figure 4. Proving that $s/3 - 4k(n - k) \leq \bar{s}_{n-k}$.

It follows that there exist exactly

$$n^2 - (n - 2k)^2 = 4k(n - k)$$

vertices lying inside G_n^3, but outside Δ_k. Moreover, for any straight-line L_i from the above trail, if L_i is parallel to the z-axis and in addition L_i "crosses a vertex that lies" inside the grid Δ_k then the length of L_{i+1} must be $\leq n - k$. It follows that

$$\frac{s}{3} - 4k(n - k) \leq \bar{s}_{n-k}. \tag{4}$$

In fact we can do better than inequality (4). Let $s(x), s(y), s(z)$ be the number of lines in the above synective partition which are parallel to the x, y, z-axis, respectively. Further let $\bar{s}_k(x, y)$, $\bar{s}_k(y, z)$, $\bar{s}_k(x, z)$, be the number of lines in the above synective partition which are parallel to the (x, y)-, (y, z)-, (x, z)-plane, respectively. Now as before we can show that

$$s(x) - 4k(n - k) \leq \bar{s}_{n-k}(y, z),$$

$$s(y) - 4k(n - k) \leq \bar{s}_{n-k}(x, z),$$

$$s(z) - 4k(n - k) \leq \bar{s}_{n-k}(x, y).$$

Adding these inequalities we obtain

$$\frac{s}{2} - 6k(n-k) \le \bar{s}_{n-k}, \tag{5}$$

which is an improvement over inequality (4). Now the idea is to sum inequalities (5) for different values of k in order to get the desired lower bound. First notice that the quantity on the left-hand side is zero exactly when

$$k = \frac{n \pm \sqrt{n^2 - s/3}}{2}.$$

Since $k \le n/2$, the largest of the two roots, which is $> n/2$, must be rejected. Call

$$k_0 = \frac{n - \sqrt{n^2 - s/3}}{2}.$$

Hence the quantity on the left-hand side of (5) is non-negative exactly when $k \le k_0$. Fix $k \le k_0$ and use inequalities (5) for $i = 2,3,...,k$ in order to obtain from (3) that

$$\begin{aligned} s &\ge \frac{n^3-1}{n-1} + \frac{\bar{s}_{n-2} + \bar{s}_{n-3} + \cdots + \bar{s}_{n-k}}{n-1} \\ &\ge \frac{n^3-1}{n-1} + \frac{(k-2)s}{2(n-1)} - \sum_{i=2}^{k} \frac{6i(n-i)}{n-1} \\ &= \frac{n^3-1}{n-1} + \frac{(k-2)s}{2(n-1)} - \frac{k(k+1)(3n-2k-1)}{n-1} + 6. \end{aligned}$$

It follows that

$$s - \frac{(k-2)s}{2(n-1)} \ge \frac{n^3-1}{n-1} + \frac{k(k+1)(3n-2k-1)}{n-1} + 6.$$

Factoring out s and dividing through by $n - k/2$ we obtain

$$s \ge \frac{n^3 - k(k+1)(3n-2k-1) + 6n - 7}{n - k/2}. \tag{6}$$

Now we need to maximize the quantity in the right-hand side of (6). Setting $k = \alpha \cdot n$, simplifying, and maximizing the resulting fraction (with respect to α) we obtain after some calculations that

$$s \ge (1.02324576) \cdot n^2,$$

which proves the existence of a constant $c > 1$ satisfying the desired lower-bound result. •

3.2. d-Dimensional Grids

As before we first discuss the upper bound. As a first approximation we iterate the face-peeling algorithm $d-2$ successive steps, up to a depth $\delta = 1/16$, i.e. a depth of $k = n/16$ lines. We will later indicate what depth should be used in order to optimize the cost. We give a table below of the $d-2$ iterations executed. For each such iteration we count the number of straight-line-turns (mirrors) used, as well as the dimensions of the solid resulting by peeling the faces of the ith iterate. These are indicated in the table below. The resulting solid after application of the $(d-2)$th iteration can be considered as consisting of $(\frac{3}{4})^{d-3} \cdot n^{d-3}$-many solids each of dimension

$$\frac{7}{8} \cdot n \times \frac{7}{8} \cdot n \times n.$$

We cover each of these solids with straight lines by using the three-dimensional face-peeling algorithm up to a depth of $n/4$. This requires $\frac{41}{32}\cdot n^2$ lines per $\frac{7n}{8}\times\frac{7n}{8}\times n$ parallelepiped, for a total of at most $(\frac{3}{4})^{d-3}\cdot\frac{41}{32}\cdot n^{d-1}$ lines.

Step	*Number of Lines Used*	*Dimensions of Resulting Solid*
1	$\frac{1}{4}\cdot n^{d-1}$	$\frac{7n}{8}\times\frac{7n}{8}\times n\times n\times n\times n\times\cdots\times n\times n$
2	$\frac{7}{8}\cdot\frac{1}{4}\cdot n^{d-1}$	$\frac{7n}{8}\times\frac{3n}{4}\times\frac{7n}{8}\times n\times n\times n\times\cdots\times n\times n$
3	$\frac{7}{8}\cdot\frac{3}{4}\cdot\frac{1}{4}\cdot n^{d-1}$	$\frac{7n}{8}\times\frac{3n}{4}\times\frac{3n}{4}\times\frac{7n}{8}\times n\times n\times\cdots\times n\times n$
4	$\frac{7}{8}\cdot(\frac{3}{4})^2\cdot\frac{1}{4}\cdot n^{d-1}$	$\frac{7n}{8}\times\frac{3n}{4}\times\frac{3n}{4}\times\frac{3n}{4}\times\frac{7n}{8}\times n\times\cdots\times n\times n$
.	...	...
.	...	...
.	...	...
$d-2$	$\frac{7}{8}\cdot(\frac{3}{4})^{d-4}\cdot\frac{1}{4}\cdot n^{d-1}$	$\frac{7n}{8}\times\frac{3n}{4}\times\frac{3n}{4}\times\frac{3n}{4}\times\frac{3n}{4}\times\frac{3n}{4}\times\cdots\times\frac{7n}{8}\times n$

By summing the quantities obtained above we obtain that

$$\frac{s(G_n^d)}{n^{d-1}} \le \frac{1}{4} + \frac{1}{4}\cdot\frac{7}{8}\cdot\sum_{j=0}^{d-4}(\frac{3}{4})^j + (\frac{3}{4})^{d-3}\cdot\frac{41}{32}$$

$$= \frac{1}{4} + \frac{7}{8}\cdot(1-(\frac{3}{4})^{d-3}) + (\frac{3}{4})^{d-3}\cdot\frac{41}{32}$$

$$= \frac{1}{4} + \frac{7}{8} + \frac{13}{32}\cdot(\frac{3}{4})^{d-3}.$$

(As a matter of fact, for four-dimensional grids we obtain an even better upper bound concerning the 3rd order term than the one above, if we move up to a depth of $n/6$ lines, namely $(38/27)\cdot n^3$.) This last upper bound generalizes easily to more general "depths". Put $\delta = 2^{-i}$ and apply the above mentioned face-peeling algorithm. A repetition of the above argument will show (after some tedious calculations) that

$$\frac{s(G_n^d)}{n^{d-1}} \le 1 + 2^{-i+1} + (1-2^{-i+2})^{d-3}\cdot(2^{-1}+2^{-i+2}-2^{-2i+3})$$

$$\le 1 + 2^{-i+1} + (1-2^{-i+2})^{d-3}$$

For $0<\varepsilon<1$ put $i-2=(1-\varepsilon)\cdot\log(d-3)$ and we easily obtain that asymptotically

$$(1-2^{i+2})^{d-3} \sim \exp[-(d-3)^{\varepsilon}].$$

Hence, asymptotically, we have that for all $0<\varepsilon<1$,

$$\frac{s(G_n^d)}{n^{d-1}} \le 1 + \frac{1}{2}\cdot\frac{1}{(d-3)^{1-\varepsilon}} + \exp[-(d-3)^{\varepsilon}].$$

With respect to lower bounds it is easy to see, using the argument for proving inequality (5) of theorem 3.1, that

$$s - d(n^{d-1}-(n-2k)^{d-1}) \le (d-1)\bar{s}_{n-k}.$$

Arguing as before we obtain that the quantity on the left-hand side of the above inequality is non-negative exactly when $k \le k_0$, where

$$k_0 = \frac{n}{2}\cdot(1 - (1 - d^{-1})^{\frac{1}{d-1}})$$

Using inequality (1), formula (3) of section 3 and simplifying we obtain that

$$s \cdot \left[1 - \frac{k_0 - 1}{(d-1)(n-1)} \right] \ge$$

$$\left[1 - \frac{k_0 - 1}{(d-1)(n-1)} \right] \cdot n^{d-1} + \frac{d}{(d-1)(n-1)} \cdot \left[(n-4)^{d-1} + (n-6)^{d-1} + \cdots + (n-2k_0)^{d-1} \right] \ge$$

$$\left[1 - \frac{k_0 - 1}{(d-1)(n-1)} \right] \cdot n^{d-1} + \frac{k_0 - 1}{n-1} \cdot n^{d-1} \ge$$

$$\left[1 + \frac{(k_0 - 1)(d-2)}{(n-1)(d-2)} \right] \cdot n^{d-1}$$

Substituting the above value of k_0 we obtain that asymptotically in d

$$s(G_n^d) \ge \left[\frac{3}{2} - \frac{1}{2} \cdot \exp[-1/d\,(d-1)] \right] \cdot n^{d-1}$$

To sum up we have proved the following theorem.

Theorem 3.2.

For all $0 < \varepsilon < 1$ the following inequality holds asymptotically in d,

$$1 + \frac{1}{2} \cdot [1 - \exp[-1/d\,(d-1)]] \le \frac{s(G_n^d)}{n^{d-1}} \le 1 + \frac{1}{2} \cdot \frac{1}{(d-3)^{1-\varepsilon}} + \exp[-(d-3)^{\varepsilon}]. \bullet$$

Acknowledgements

Discussions with P. Clote, A. Pheidas, and K. Yap are gratefully acknowledged.

References

[1] Berge, C., *Theory of Graphs*, Methuen, 1962.

[2] Guy, R., *Unsolved Problems in Number Theory*, Unsolved Problems in Intuitive Mathematics, Vol. 1, Springer Verlag, Berlin, 1981.

[3] Knuth, D., *The Art of Computer Programming: Sorting and Searching*, Vol. 3, Addison Wesley, 1973.

[4] Ntafos, S., *On Gallery Watchmen in Grids*, Information Processing Letters 23(1986) 99-102, North Holland.

[5] O'Rourke, J., *Art Gallery Theorems and Algorithms*, The International Series of Monographs on Computer Science, Oxford University Press, 1987.