

Link Length of Rectilinear Hamiltonian Tours in Grids *

Evangelos Kranakis^(1,2) Danny Krizanc^(2,3) Lambert Meertens⁽²⁾

(1) Carleton University, School of Computer Science
Ottawa, Ontario, K1S 5B6, Canada

(2) Centrum voor Wiskunde en Informatica (CWI)
P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

(3) University of Rochester, Department of Computer Science
Rochester, New York, 14627, U.S.A

ABSTRACT. The link length of a walk in a multidimensional grid is the number of straight line segments constituting the walk. Alternatively, it is the number of turns that a mobile unit needs to perform in traversing the walk. A rectilinear walk consists of straight line segments which are parallel to the main axis. We wish to construct rectilinear walks with minimal link length traversing grids. If G denotes the multidimensional grid, let $s(G)$ be the minimal link length of a rectilinear walk traversing all the vertices of G . In this paper, we develop an asymptotically optimal algorithm for constructing rectilinear walks traversing all the vertices of complete multidimensional grids and analyze the worst-case behavior of $s(G)$, when G is a multidimensional grid.

1 Introduction.

The link length of a polygonal path connecting two points in a polygon is defined to be the number of straight line segments constituting the path. The link distance of two points is the minimum link length of any polygonal path connecting the points. Alternatively, it is the number of turns that a mobile unit will need to take when traversing a minimum-turn-walk connecting the two points in order to move from one point to the other. This distance was introduced in [12] and was designed to measure the cost

*A short version of the paper has appeared in the proceedings of the 2nd Canadian Conference in Computational Geometry, Ottawa, August 6-10, 1991, pp. 328-331.

of moving along a path in a simple polygon when straightline motion is easy but turns are expensive.

In this paper, we introduce a new problem which we call the “Minimum Link Length Hamiltonian Tour”: Given a set of distinguished vertices of a polygon determine the link length of a minimum link length Hamiltonian path which visits all the vertices. It can be shown [3] that determining the minimum link length Hamiltonian path for points in general position is NP-complete, since the “edge embedding on a grid problem” [5] can be reduced to it. The discussion below is restricted to rectilinear tours which visit all the vertices of a complete d -dimensional grid. This restriction greatly simplifies the problems considered above, but as will be seen the problem at hand remains interesting and non-trivial even in this case. We give the exact solution for the problem for 2-dimensional grids and give nontrivial bounds for d -dimensional grids, for $d > 2$. Despite the seeming simplicity of the question, the exact solution for 3-dimensional grids remains open.

1.1 Preliminaries

Next we introduce some definitions and notations. The d -dimensional grid of size n , denoted G_n^d , is the graph with vertices $v: = (v_1, v_2, \dots, v_d)$ such that $1 \leq v_i \leq n$, for $i = 1, \dots, d$, and edges (u, v) such that $\sum_{i=1}^d |u_i - v_i| = 1$. To every rectilinear path P , we associate the unique partition (called the **rectilinear partition**) of P , L_1, L_2, \dots, L_s , consisting of the “maximal straight line segments” of P . The number $s: = s(P)$ is the **link length** of the path P . It is not hard to see that for a Hamiltonian path P , $s(P) - 1$ is exactly the number of times one must change direction moving along P in order to traverse all the vertices of P . A rectilinear path traversing all the vertices of the grid G is called a **complete**, rectilinear tour. We also define the rectilinear number of the grid by

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

The present paper studies the problem of determining $s(G)$ for the complete multidimensional grids. In particular, we estimate the value of the quantity $s(G_n^d)$, for $d \geq 2$, $n \geq 1$. A straightforward estimate is given in the following theorem.

Theorem 1.

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

Proof: For the upper bound, form a complete rectilinear tour of G_n^d by linking the complete rectilinear tours of n copies of G_n^{d-1} . For the lower

bound, let L_1, L_2, \dots, L_s be a rectilinear partition of a complete rectilinear tour of G_n^d such that $s = s(G_n^d)$. Let $|L_i|$ denote the number of vertices in L_i . Counting the number of vertices involved we obtain that

$$sn - (s - 1) \geq |L_1| + \sum_{i=2}^s (|L_i| - 1) \geq n^d.$$

Simplifying terms we obtain the desired lower bound. □

The main result of the paper is to show that the actual value of $s(G_n^d)$ satisfies much sharper upper and lower bounds than those implied by the above inequality of Theorem 1.

1.2 Related results and relevant literature

It is worth mentioning that the following “Mirror Placement” problem on multidimensional grids is, in fact, equivalent to the above rectilinear path problem. 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 (for example, 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 can now easily relate the rectilinear numbers with the previously mentioned “Mirror Placement” problem. Assuming the above notation, let L_i consist of the sequence $v_{r_{i+1}}, \dots, v_{r_{i+1}}$ of vertices, where $i = 0, 1, \dots, s-1$, $r_0 = 0 < r_1 < \dots < r_s$. It is clearly possible to place a mirror at each of the nodes v_{r_2}, \dots, v_{r_s} , and the light source at node v_{r_1} in such a way that for each $1 \leq i < s$ the incidence light beam moves along the straight line L_i while the reflecting light beam along L_{i+1} .

It should also be noted that the “Mirror Placement” problem, although related, is different from the well-known “Art Gallery” problem, first proposed by Klee [11], 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 [10, 11].

A number of researchers have also considered restricted versions of this art gallery problem [11]. [7], has recently shown that $\lceil n/4 \rceil$ watchman are

necessary and sufficient for the case of rectilinear polygons with an arbitrary number of holes. [10] studied the question of gallery watchman in incomplete grids. He gave a polynomial time algorithm for placing watchmen in incomplete 2-dimensional grids and showed the problem is NP-complete in the case of incomplete 3-dimensional grids. A related problem was introduced in [2], that of optimum watchman routes. Here there is a single mobile watchman and we are asked to determine a minimum length route for the watchman with the property that every point in the gallery is visible from at least one point along the route.

There are also studies of the link distance metric which concentrate on algorithms for constructing the link center of an n vertex polygon (in time $O(n^2)$), that is, the set of points inside the polygon whose maximal link distance to any other point inside the polygon is minimized [9], finding a point in the link center of the polygon (in time $O(n \log n)$) [4], or finding the link diameter of the polygon, that is, the maximal link distance between any two points (in time $O(n \log n)$) [13].

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 [1, 6], 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 [8]?

1.3 Outline of the paper

In Section 2 we describe our main algorithm, the so-called face-peeling algorithm, for rectilinear Hamiltonian tours 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.

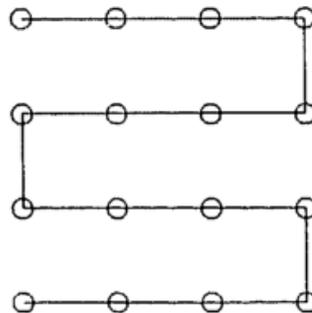


Figure 1: Establishing the upper bound $s(G_n^2) \leq 2n - 1$.

2 The face-peeling algorithm.

In this section we give a complete intuitive description of the face-peeling algorithm for constructing rectilinear paths traversing all the vertices of multidimensional grids. (The term face-peeling arises from the fact that our algorithm generates the required rectilinear path by peeling the faces of the grid level-by-level.)

2.1 Two dimensional grids

We begin with the simple case $d = 2$. We can prove the following theorem.

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

Proof: To prove $s(G_n^2) \leq 2n - 1$ consider the path depicted in Figure 1. Straightforward counting shows that $s(G_n^2) \leq 2n - 1$.

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

$$|h - v| \leq 1.$$

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 the above inequality 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, $s \geq 2n - 1$, as desired. \square

It is easy to see that the same argument will work for the $m \times n$ -grid. We single out this simple observation as a corollary which will be used frequently in the sequel.

Corollary 3. *Exactly $2 \cdot \min(m, n) - 1$ turns are necessary and sufficient in order to solve the "rectilinear path" problem for the $m \times n$ -grid.*

2.2 Three-dimensional grids

Next we consider the case $d = 3$. We construct a rectilinear path traversing vertices of the three dimensional grid by first traversing a certain identical portion of each of the horizontal plane grids. There remains a three dimensional rectangular grid which we can traverse easily with a rectilinear path.

Joining these path-portions we form the desired rectilinear path traversing all the vertices of the grid.

In more detail, the three dimensional face-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 in a spiral-like fashion and covering each time all of the corresponding vertices. The idea for doing this is depicted in Figure 2.

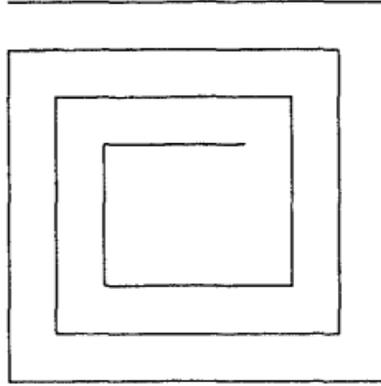


Figure 2: Traversing the vertices of horizontal plane grids.

Proceed this way until you cover vertices of the plane grid up to a depth of $\lfloor n/4 \rfloor$ vertices. This leaves an $\lfloor n/2 \rfloor \times \lfloor n/2 \rfloor$ 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 with similar straight lines.

At the end of traversing the top plane grid you are left with a parallelepiped grid of dimensions $\lfloor n/2 \rfloor \times \lfloor n/2 \rfloor \times n$ 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 the bottom plane. (Figure 3 depicts such a path for the three-dimensional $4 \times 4 \times 4$ grid.) To be more exact we traverse the parallelepiped in the following way. We think of it as consisting of $\lfloor n/2 \rfloor$ -many $\lfloor n/2 \rfloor \times n$ plane grids each parallel to the yz -plane. Using the algorithmic construction in the corollary to Theorem 2 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.

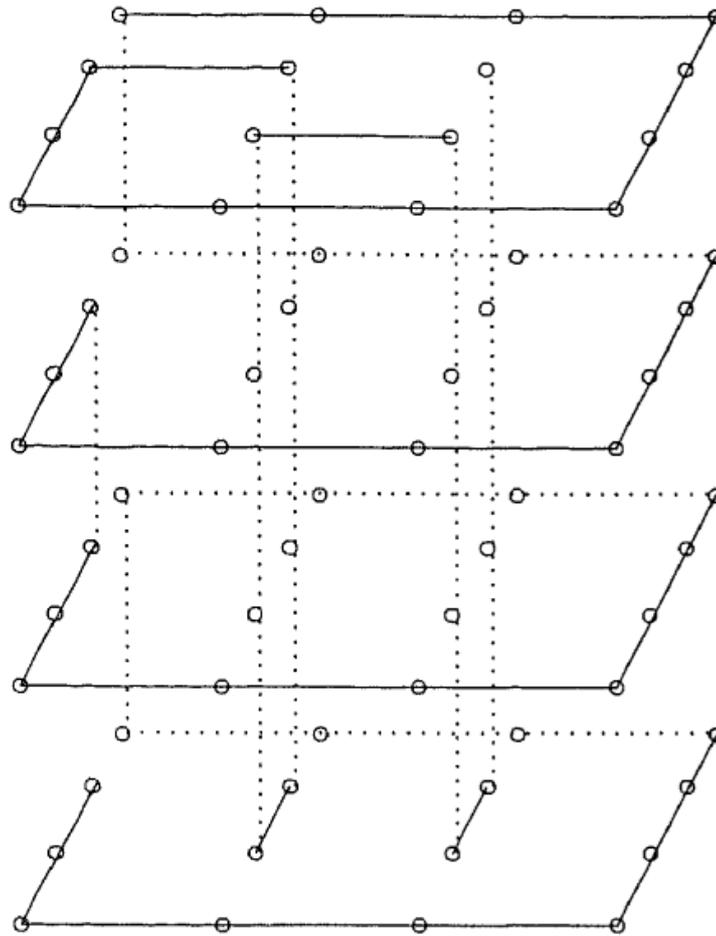


Figure 3: A rectilinear tour T traversing G_4^3 with $s(T) = 27$.

2.3 d -dimensional grids

The d -dimensional ($d \geq 4$) 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 $i = 1, 2, \dots, d - 2$. During the i th 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, $G_1 = G_n^d, G_2, \dots, G_{d-2}$, in such a way that the i th 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 rectilinear path. (It will be determined in the next section how efficient this method is.)

It remains to describe more formally the i th 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 $(a_1n) \times \dots \times (a_dn)$, where $1 \leq a_1, \dots, a_d > 0$, $i \leq d - 2$ and let $0 < \delta < 1$. The i th iteration of the face-peeling algorithm gives a rectangular grid $H = G_{i+1}$ of dimensions $(b_1n) \times \dots \times (b_dn)$ such that $b_k = a_k$, for $k \leq i - 1$ 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 $(d - 1)$ -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 3-dimensional grid described above, and then bend the rectangle at the appropriate points in order to bring it back to its original d -dimensional shape. This gives a description of the algorithm.

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

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

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

3 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 3.1 Three dimensional grids

Theorem 4. *There is a constant $c > 1$ such that for all $n \geq 3$,*

$$c \cdot n^2 \leq s(G_n^3) \leq \frac{3}{2} \cdot n^2 + n - 1.$$

Proof: We give separately the upper and lower bounds stated above.

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 face-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 $\lceil n/4 \rceil$ 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 $4\lceil n/4 \rceil$, giving a total of $4\lceil n/4 \rceil n$ 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

$$4\lceil n/4 \rceil n + n - 1. \quad (1)$$

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 corollary of Theorem 2 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, that is, a total of

$$n\lceil n/2 \rceil - 1 \quad (2)$$

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

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

Differentiating 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 rectilinear path of G_n^3 with rectilinear partition L_1, L_2, \dots, L_s , such that $s = s(P)$. For each k let s_k (respectively, \bar{s}_k) be the number of lines in the above rectilinear partition of length exactly (respectively, \leq) k . It is then clear that

$$\begin{aligned} s &= s_{n-1} + s_{n-2} + \dots + s_2 + s_1, \\ s &= s_{n-1} + s_{n-2} + \dots + s_{n-k} + \bar{s}_{n-k-1}, \end{aligned}$$

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

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

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

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

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

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

This last inequality is equivalent to

$$\frac{n^3 - 1}{n-1} + \frac{\bar{s}_{n-2} + \bar{s}_{n-3} + \dots + \bar{s}_2 + \bar{s}_1}{n-1} \leq s. \quad (3)$$

So now we concentrate on getting a lower bound for $\bar{s}_{n-2} + \bar{s}_{n-3} + \dots + \bar{s}_2 + \bar{s}_1$. The idea for doing this is the following. Each of the straight-lines constituting the rectilinear partition of the given path 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).

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 path, 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

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

In fact, we can do better than inequality (4). Let $s(x)$, $s(y)$, $s(z)$, be the number of lines in the above rectilinear 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 rectilinear partition which are parallel to the (x, y) -, (y, z) -, (x, z) -plane, respectively. Now as before we can show that

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

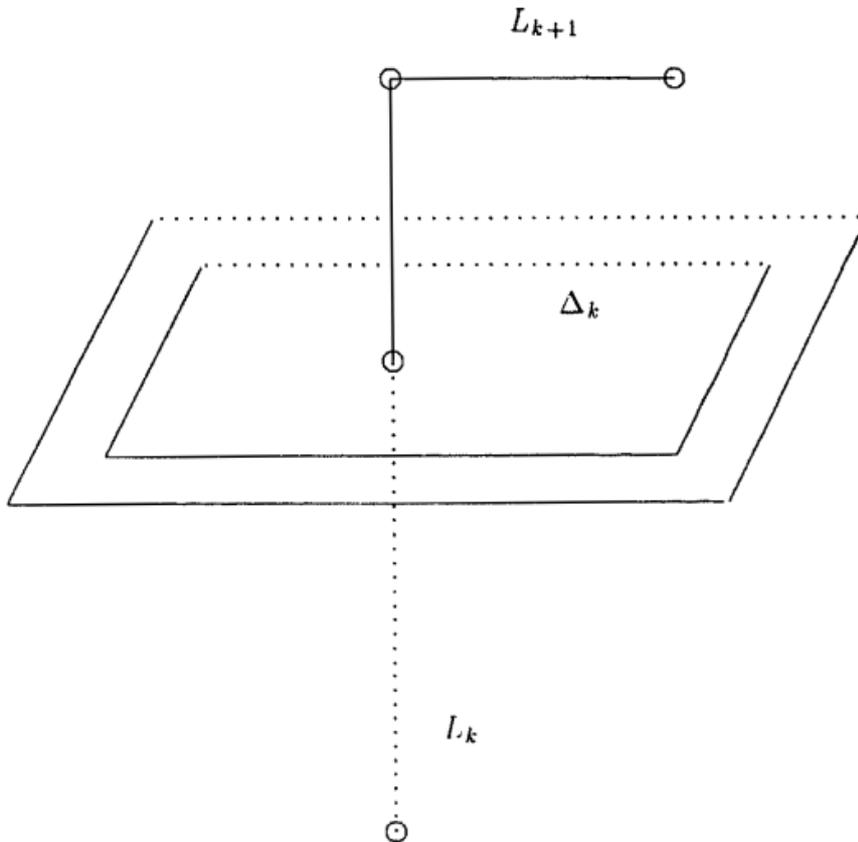


Figure 4: Proving that $s/3 - 4k(n - k) \leq \bar{s}_{n-k}$

Adding these inequalities we obtain

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

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 $k \leq n/2$, 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 $k \leq k_0$. Fix $k \leq k_0$ and use inequalities (5) for $i = 2, 3, \dots, k$ in order to obtain from (3) that

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

It follows that

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

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

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

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

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

which proves the existence of a constant $c > 1$ satisfying the desired lower-bound result. This completes the proof of the theorem. \square

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 $\delta = 1/16$, that is, a depth of $k = n/16$ lines. We will later indicate what depth should be used in order to optimize the cost. For each such iteration we count the number of straight-line-turns used, as well as the dimensions of the solid resulting by peeling the faces of the i th 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 3-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	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	$\frac{7}{8} \cdot \left(\frac{3}{4}\right)^2 \cdot \frac{1}{4} \cdot n^{d-1}$	$\frac{7n}{8} \times \frac{3n}{4} \times \frac{3n}{4} \times \frac{3n}{4} \times \frac{7n}{8} \times n \times \cdots \times n \times n$
·
·
·
$d - 2$	$\frac{7}{8} \cdot \left(\frac{3}{4}\right)^{d-4} \cdot \frac{1}{4} \cdot n^{d-1}$	$\frac{7n}{8} \times \frac{3n}{4} \times \frac{3n}{4} \times \frac{3n}{4} \times \frac{3n}{4} \times \frac{3n}{4} \times \frac{3n}{4} \times \cdots \times \frac{7n}{8} \times n$

By summing the quantities obtained above we obtain that

$$\begin{aligned} \frac{s(G_n^d)}{n^{d-1}} &\leq \frac{1}{4} + \frac{1}{4} \cdot \frac{7}{8} \cdot \sum_{j=0}^{d-4} \left(\frac{3}{4}\right)^j + \left(\frac{3}{4}\right)^{d-3} \cdot \frac{41}{32} \\ &= \frac{1}{4} + \frac{7}{8} \cdot \left(1 - \left(\frac{3}{4}\right)^{d-3}\right) + \left(\frac{3}{4}\right)^{d-3} \cdot \frac{41}{32} \\ &= \frac{1}{4} + \frac{7}{8} + \frac{13}{32} \cdot \left(\frac{3}{4}\right)^{d-3}. \end{aligned}$$

(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 $\delta = 2^{-t}$ and apply

the above mentioned face-peeling algorithm. A repetition of the above argument will show that

$$\begin{aligned}\frac{s(G_n^d)}{n^{d-1}} &\leq 1 + 2^{-t+1} + (1 - 2^{-t+2})^{d-3} \cdot (2^{-1} + 2^{-t+2} - 2^{-2t+3}) \\ &\leq 1 + 2^{-t+1} + (1 - 2^{-t+2})^{d-3}\end{aligned}$$

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

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

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

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

With respect to lower bounds it is easy to see, using the argument for proving inequality (5) of Theorem 4, that

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

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

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

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

$$\begin{aligned}s \cdot \left(1 - \frac{k_0 - 1}{(d - 1)(n - 1)}\right) &\geq \left(1 - \frac{k_0 - 1}{(d - 1)(n - 1)}\right) \cdot n^{d-1} \\ &\quad + \frac{d((n - 4)^{d-1} + (n - 6)^{d-1} + \dots + (n - 2k_0)^{d-1})}{(d - 1)(n - 1)} \\ &\geq \left(1 - \frac{k_0 - 1}{(d - 1)(n - 1)}\right) \cdot n^{d-1} + \frac{k_0 - 1}{n - 1} \cdot n^{d-1} \\ &\geq \left(1 + \frac{(k_0 - 1)(d - 2)}{(n - 1)(d - 2)}\right) \cdot n^{d-1}.\end{aligned}$$

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

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

To sum up we have proved the following theorem.

Theorem 5. For all $0 < \epsilon < 1$ the following inequality holds asymptotically in d ,

$$1 + \frac{1}{2} \cdot [1 - \exp[-1/d(d-1)]] \leq \frac{s(G_m^d)}{n^{d-1}} \leq 1 + \frac{1}{2} \cdot \frac{1}{(d-3)^{1-\epsilon}} + \exp[-(d-3)^\epsilon].$$

Thus, we see that there exists a constant $c_d > 0$ depending only on d and an arbitrarily small constant $c'_{d,\epsilon} > 0$ depending only on d and ϵ such that

$$1 + c_d \leq \frac{s(G_m^d)}{n^{d-1}} \leq 1 + \frac{1}{2} \cdot \frac{1}{(d-3)^{1-\epsilon}} + c'_{d,\epsilon},$$

asymptotically in d .

4 Conclusion.

We have studied the asymptotic behavior of the link length of rectilinear paths traversing all the vertices of multidimensional grids and have given nontrivial bounds of the optimal link length of such rectilinear walks. Exact bounds for grids of dimension 3 or greater are still unknown. The results for the 2 and 3 dimensional cases discussed above (which are, of course, often notoriously misleading) lead us to conjecture

$$s(G_n^d) = \frac{d}{d-1} \cdot n^{d-1} + O(n^{d-2}).$$

The problem appears to be even more difficult in the case of arbitrary (as opposed to rectilinear) walks. For example, our lower bound proof in Theorem 4 is not valid anymore if we were to allow "turns" with other angles as well, for example, 45, 90 and 135. (However, for such walks in two dimensional grids we can show that $s(G_n^2) = 2n - 2$, $n \geq 3$.)

Questions concerning the link length of tours in incomplete grids, more general polygons, or even when the tour is permitted to pass through vertices more than once, are entirely open.

5 Acknowledgements.

Discussions with P. Clote, A. Pheidas, and C. Yap, as well as the comments of the anonymous referee are gratefully acknowledged.

References

- [1] C. Berge, "Theory of Graphs", Methuen, 1962
- [2] Chin and S. Ntafos, Optimum watchman routes, Proc. of 2nd ACM Conference on Computational Geometry, (1986), 1-10.

- [3] P. Clote, Personal Communication, 1989.
- [4] H. Djidjev, A. Lingas, and J.- R. Sack, An $O(N \log N)$ algorithm for computing a link center in a polygon, in "STACS", Springer-Verlag, 1989. *Lecture Notes in Computer Science* 349, pp. 96–107.
- [5] F. Gavril, Some NP-complete problems on graphs, Proc. 11th Conference on Information Sciences and Systems, (1977), 91–95, Johns Hopkins University, Baltimore, MD.
- [6] R. Guy, Unsolved problems in number theory, in "Unsolved Problems in Intuitive Mathematics", Vol. 1, Springer-Verlag, Berlin, 1981.
- [7] F. Hoffman, On the rectilinear art gallery problem, in "Proc. of ICALP", M. Paterson, ed., Springer-Verlag *Lecture Notes in Computer Science*, Vol 443, 1990.
- [8] D. Knuth, "The Art of Computer Programming: Sorting and Searching", Vol. 3, Addison Wesley, 1973.
- [9] W. Lenhart, R. Pollack, J.- R. Sack, R. Seidel, M. Sharir, S. Suri, G. Toussaint, S. Whitesides, and C. Yap, Computing the link center of a simple polygon, in "Proceedings 3rd ACM Conference on Computational Geometry," 1987, 1–10; *Discrete and Computational Geometry* 3(3) (1988), 281–293.
- [10] S. Ntafos, On gallery watchmen in grids, *Information Processing Letters*, 23, (1986) 99–102, North Holland.
- [11] J. O'Rourke, Art gallery theorems and algorithms, "The International Series of Monographs on Computer Science", Oxford University Press, 1987.
- [12] S. Suri, A polygon partitioning technique for link distance problems, Manuscript, Department of Computer Science, Johns Hopkins University, November 1986.
- [13] S. Suri, The all-geodesic furthest neighbors problem for simple polygons, Proceedings 3rd ACM Conference on Computational Geometry pp. 64–75, 1987.
- [14] S. Suri, Minimum link paths in polygons and related problems, Ph.D. Thesis, Department of Computer Science, Johns Hopkins University, August 1987.