RAPPORT DE RECHERCHE

LRI

SOME EULERIAN PARAMETERS ABOUT PERFORMANCES OF A CONVERGENCE ROUTING IN A 2D-MESH NETWORK

BARTH D / BERTHOME P / LAFOREST C / VIAL S

Unité Mixte de Recherche 8623
CNRS-Université Paris Sud-LRI

01/2003

Rapport de Recherche N° 1346

CNRS – Université de Paris Sud
Centre d'Orsay
LABORATOIRE DE RECHERCHE EN INFORMATIQUE
Bâtiment 650
91405 ORSAY Cedex (France)

Some Eulerian parameters about performances of a convergence routing in a 2D-mesh network *

Dominique Barth† Pascal Berthomé‡ Christian Laforest§ Sandrine Vial¶

7th January 2003

Abstract

In this paper, we focus on the capacity of an optical routing strategy called the *Eulerian Routing* on the 2-D mesh. This strategy provides some guaranties for the routing such as the ending guaranty, i.e., any packet in the network will eventually reach its destination within a bounded delay. Coupled with the shortcut strategy, this routing may obtain good static performance: a message may reach its destination in merely the diameter of the underlying graph.

The quality of these strategies is based on the design of Eulerian circuits on the underlying graph. Several bounds and results are given in the 2-D mesh.

Résumé

Dans ce papier, nous nous intéressons aux possibilités d'une stratégie de routage optique (sans stockage) appelée *routage eulérien* sur la grille bidimensionnelle. Cette stratégie apporte quelques garanties au routage parmi lesquelle la garantie de terminaison: tout message entrant dans le réseau atteindra sa destination au bout d'un délai borné. En associant cette stratégie avec la technique des raccourcis, le routage peut avoir de bonnes propriétés statique: un message pourra atteindre sa destination en un nombre d'étape équivalent au diamètre du graphe sousjascent.

Les qualités de ces stratégies sont basées sur la donnée de circuits eulériens dans le graphe. Plusieurs bornes et résultats sont présentés dans la grille.

---

*This work has been supported by the French *RNRT* Telecommunication Project n. 99S0201 *ROM*.

†PRiSM, FRE 2510 du CNRS, Université de Versailles-Saint-Quentin. email barth@prism.uvsq.fr

‡LRI, UMR 8623 du CNRS, Université Paris-Sud. email berthome@lri.fr

§LaMI, UMR 8042 du CNRS, Université d'Evry; email laforest@lami.univ-evry.fr

¶LaMI UMR 8042 du CNRS, Université d'Evry; email svial@lami.univ-evry.fr

1 Introduction

In this paper, we focus on some digraph parameters to evaluate the quality of a network, from which the digraph is the topology, in terms of performances of specific packet routing algorithms. We deal with packet routing strategies without intermediate storage of data packets (hereafter simply called packets) [1][12], such as deflection routing [4][5][13] (we especially focus on it in the RNRT project *ROM*† dealing with all-optical telecommunication networks). These techniques are known to clearly avoid deadlocks (packets in the network do not move) but livelocks could occur (packets move but never reach their destination), except for some cases of deflection routing in some classes of networks such as trees or triangulated graphs [7].

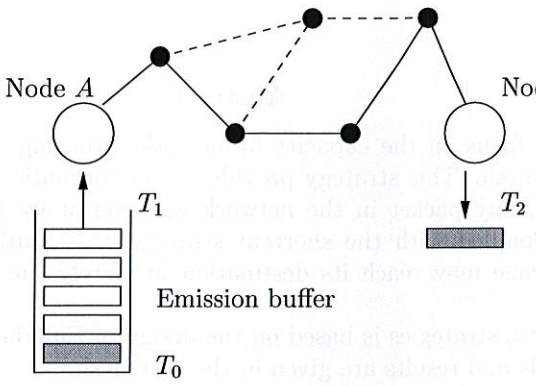


Figure 1: Life steps of a packet in a network.

We deal with techniques of routing giving some performances guarantees about the life-time of a packet into the target network, similar to the ones defined in [6]. Consider that a node A in a network modeled by a digraph G has to send a packet to a node B (see Figure 1). At time T_0, this packet is inserted in the emission buffer of A. At time T_1, it is emitted by A in the network from this buffer (remind that there is no possible intermediate storage for this packet). Finally, at time T_2, it reaches its destination node. Any routing strategy can be characterized by three parameters that define its intrinsic quality.

Ending guarantee: (i.e., no livelock) there exists a finite value $\mathcal{G}_e$ such that any emitted packet will reach its destination within a maximal number of steps $\mathcal{G}_e$ (i.e., $T_2 - T_1 \leq \mathcal{G}_e$).

Speed guarantee: there exists a finite value $\mathcal{G}_s$ such that if a packet is alone in the network, it follows a path of length at most $\mathcal{G}_s$ times the distance from its origin to its destination (i.e., in this case, $T_2 - T_1 \leq \mathcal{G}_s \cdot dist_G(A, B)$).

Emission guarantee: there exists a finite value $\mathcal{G}_m$ such that each packet inserted in the emission buffer of a node will be emitted in the network after at most $\mathcal{G}_m$ steps (i.e., $T_1 - T_0 \leq \mathcal{G}_m$).

In this paper, we mainly focus on the Ending and Speed guarantees. The Emission guarantee is rather a measure of the performance of the routing.

A speed guarantee $\mathcal{G}_s = 1$ is classically obtained by using a shortest-path deflection routing strategy, but this strategy does not always leads to a finite ending guarantee. To obtain a finite ending guarantee with this deflection routing technique, one solution is to use priorities on the packets function of the time they have spent in the network. Using the techniques given in [4], we obtain an ending guarantee of $O(n)$ in any mesh with n vertices.

To obtain a finite ending guarantee, another solution is to use a *convergence routing* technique [10][12][15]. In such a routing, packets are routed along a global sense of direction, which gives an ending guarantee. As proposed in various works [6][10][15], such a global sense of direction can be created by using some decompositions of the target digraphs (or of a covering subdigraph of it) into circuits [2][16]. In [6], Feige gives such a technique, based on an Eulerian circuit in a subdigraph, ensuring an ending guarantee equal to $O(n^{3/2})$ for any graph with a minimal number of edges.

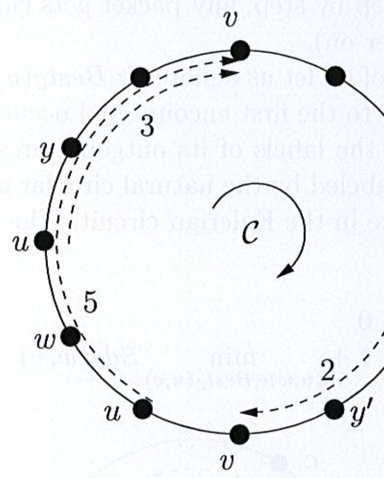


Figure 2: Example of Eulerian circuit $\mathcal{C}$ in a digraph G with $d_{\mathcal{C}}(u, v) = 5$.

Here, as a particular case, we focus on the use of an Eulerian routing, i.e., in which packets follow an Eulerian circuit in the network. Consider an Eulerian circuit $\mathcal{C}$ in a digraph G (see for example [8] for results about Eulerian circuits), where G represents the network. Thus, such an Eulerian circuit can be seen as a circuit of $|A(G)|$ arcs, where a vertex v of G has $\Delta_G(v)$ occurrences. Each emitted packet follows $\mathcal{C}$ and, at each step, is having priority on the next arc on this circuit. Then, a packet emitted by a node u and having node v as destination will hopefully reach it. Then, $d_{\mathcal{C}}(u, v)$, the maximal number of arcs on $\mathcal{C}$ between one occurrence of the source vertex u and the first following occurrence in $\mathcal{C}$ of the destination vertex v, is the major parameter of this routing strategy. It represents the longest delay for packet delivery from vertex u to vertex v. In Figure 2, this maximal distance is $d_{\mathcal{C}}(u, v) = 5$, even if there exists a path of length 2 in $\mathcal{C}$ between u and v. Using $\mathcal{C}$, any packet emitted in G reaches its destination in at most $stretchW_{\mathcal{C}}$ steps, where

$$stretchW_{\mathcal{C}} = \max_{\substack{u,v \in V(G) \\ u \neq v}} d_{\mathcal{C}}(u, v). \tag{1}$$

Thus, in this case, the ending guarantee obtained by using an Eulerian routing on $\mathcal{C}$ in G is $\mathcal{G}_e = stretchW_{\mathcal{C}}$. To obtain the best ending guarantee for G by using an Eulerian routing, we focus on the *Eulerian stretch* of a digraph G defined as follows. Let $Eul(G)$ be the set of all the Eulerian circuits of G. The Eulerian stretch of G is defined by:

$$\mathcal{E}(G) = \min_{\mathcal{C} \in Eul(G)} stretchW_{\mathcal{C}}. \tag{2}$$

Thus, this Eulerian routing technique is clearly interesting in terms of ending guarantee. However, it also gives bad performances considering the speed guarantee. To improve this latter parameter using an Eulerian routing strategy, some works propose to use *shortcuts* [2][16], i.e., a packet can jump from an occurrence of a vertex to another one on the Eulerian circuit. In the example of Figure 2, consider that a packet having destination vertex v reaches vertex u on edge (w, u) on $\mathcal{C}$. Thus, this packet has the highest priority to be sent on edge (u, y). From this edge, the first occurrence of vertex v is encountered after three steps on $\mathcal{C}$. But, the first occurrence of vertex v from edge (u, y') is encountered after only two steps. Thus, if this arc is free, the packet will be emitted on this arc (u, v'). If so, we say that this packet have used a *shortcut* (the path followed by the packet is then w, u, y', v). Note that, by using shortcuts in an Eulerian routing, we still have a convergence routing since step by step, any packet gets closer to its destination (even if the packet cannot use any shorcut frther on).

For any pair of vertices u and v of G, let us denote by $Best_{\mathcal{C}}(u, v)$ the set of arcs (u, y) of G such that the distance on $\mathcal{C}$ from this arc to the first encountered occurrence of v is minimal (this can be locally computed by u by knowing the labels of its outgoing arcs on $\mathcal{C}$ and those of the incoming arcs of v on $\mathcal{C}$, each arc in $\mathcal{C}$ being labeled by the natural circular order induced by this circuit). Let call $best_{\mathcal{C}}(u, v)$ this shortest distance in the Eulerian circuit. The *shortcut distance from a vertex u to the vertex v* on $\mathcal{C}$ is defined by:

$$Sd_{\mathcal{C}}(u, v) = \begin{cases} 0 & \textit{if } u = v; \\ 1 + \min_{(u,w) \in Best_{\mathcal{C}}(u,v)} Sd_{\mathcal{C}}(w, v) & \textit{otherwise.} \end{cases} \quad (3)$$

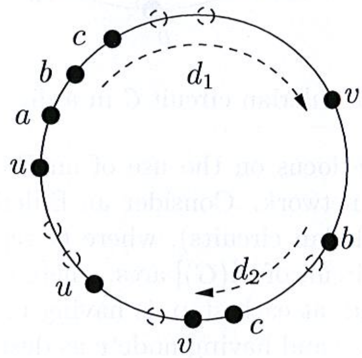


Figure 3: Eulerian circuit $\mathcal{C}$ of a graph G with $\delta_G(u) = \delta_G(v) = 2$, $dist_G(u, v) = 4$ and $Sd_{\mathcal{C}}(u, v) = d_2 + 2$.

Notice that, as it is shown in Figure 3, in general for any pair of vertices u, v of a graph G and any Eulerian circuit $\mathcal{C}$ of G, $dist_G(u, v) < Sd_{\mathcal{C}}(u, v)$.

Thus, $Sd_{\mathcal{C}}(u, v)$ is the smallest number of steps needed by a packet to go from u to v using the shortcut strategy. Let us now define the *Eulerian Efficiency* of this routing as:

$$EE_{\mathcal{C}} = \max_{u,v \in V(G)} \frac{Sd_{\mathcal{C}}(u, v)}{dist_G(u, v)} . \quad (4)$$

Thus, by precisely considering the definition of the speed guarantee, for this routing technique, $\mathcal{G}_s = EE_{\mathcal{C}}$. It is obvious that the ratio $\frac{Sd_{\mathcal{C}}(u,v)}{dist_G(u,v)}$ is significant only if $dist_G(u, v)$ is sufficiently large. To obtain a more significant evaluation of the routing performances, we take into account the average shortcuts distances in the network. Thus, we focus in fact on two parameters about speed guarantee, i.e., the *shortcuts diameter* $Sd_{\mathcal{C}}$ and the *average eulerian efficiency* $AEE_{\mathcal{C}}$ defined by:

$$Sd_{\mathcal{C}} = \max_{u,v \in V(G)^2} Sd_{\mathcal{C}}(u, v) \quad (5)$$

$$AEE_{\mathcal{C}} = \frac{ASd_{\mathcal{C}}}{ADist_G}, \tag{6}$$

where:

$$ASd_{\mathcal{C}} = \frac{\sum_{u,v \in V(G)} Sd_{\mathcal{C}}(u,v)}{|V(G)^2|} \tag{7}$$

and

$$ADist_G = \frac{\sum_{u,v \in V(G)} dist_G(u,v)}{|V(G)^2|}. \tag{8}$$

Finally, to compute the best speed guarantee that can be obtained in G by using an Eulerian routing, we focus on the *itinerary index* of G defined by

$$\mathcal{I}(G) = \min_{\mathcal{C} \in Eul(G)} Sd_{\mathcal{C}}. \tag{9}$$

The network topology we focus on in this paper is the 2D-mesh $M(p,q)$, considered as a symmetric digraph made of p lines numbered from 0 to $p-1$ and q columns numbered from 0 to $q-1$. Most results about deflection routing performances have been especially studied in the 2D-mesh [14][11][9][5][13].

In the next section, we deal with the Eulerian stretch of any 2d-mesh by giving Eulerian circuits with near-optimal stretch. In Section 3.2, we give some upper bounds of the Shortcuts diameter of the 2d-mesh, and we deal with the average slowing up of Eulerian routing strategies on it.

2 The Eulerian stretch of the 2D-mesh

In this section, we only consider square meshes, where $p = q$. Results can be extended to the general case.

As a first result, the worst Eulerian diameter must be odd. This is due to the relation between this parameter and the maximal distance between two consecutive occurrences of the same vertex on an Eulerian circuit ($\mathcal{S}_{\mathcal{C}}$): $\mathcal{S}_{\mathcal{C}} = stretchW_{\mathcal{C}} + 1$ [3]. Since cycles in grids have even size, this implies that $stretchW_{\mathcal{C}}$ must be odd.

Theorem 1 *For any integer p, there exists an Eulerian circuit $\mathcal{C}$ on the square 2D-mesh such that:*

$$stretchW_{\mathcal{C}} = \begin{cases} 5 & \text{if } p = 2 \\ 13 & \text{if } p = 3 \\ 2p(p-1)+3 & \text{if } p \text{ is odd} \\ 2p(p-1)+2p-3 & \text{otherwise.} \end{cases}$$

Let first give an idea of the proof of this theorem. A more detailed proof is given in Sections 2.1 and 2.2.

Sketch of proof: For the simple cases, note that if $p = 2$, the mesh degenerates into a cycle, and if $p = 3$, a simple case study gives an Eulerian circuit with Eulerian stretch 13. An extensive search finds 8 such Eulerian circuits, one of them is presented in Figure 4. Both cases are optimal in terms of this parameter.

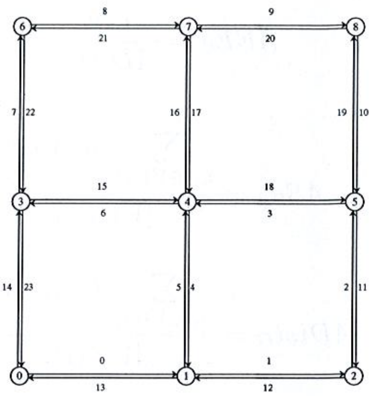


Figure 4: Eulerian circuit with eulerian stretch 13.

For the other case, we have built an Eulerian circuit that tries to balance the points in the borders. In Figure 5, we present half of the Eulerian circuit for the square mesh of odd size. The other part is very similar. Starting from the Eulerian circuit presented in the right part of the figure, we exchange the dotted portions in the two parts. Using this technique several times ($p/2-1$ times), we obtain the desired Eulerian stretch for the grid. The complete proof turns into a finite number of cases, each of them being quite simple to solve (compute explicitly the distances).

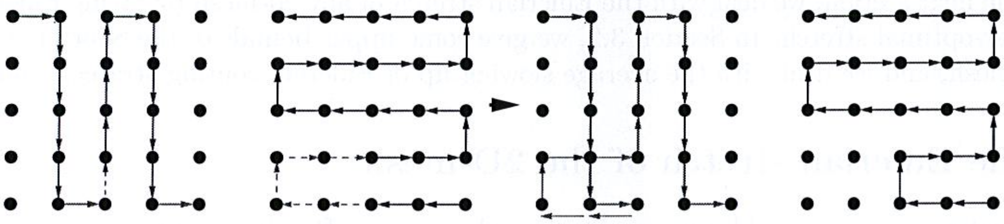


Figure 5: 2 portions of the Eulerian circuit of the 2D-mesh, the other part is symmetric. Dotted parts should be exchanged to improve the quality of the Eulerian stretch.

In order to compare this result with general bounds, let recall one of the results obtained in [3]. We have shown that, given a vertex v of a digraph G and an integer k, the problem of knowing whether there exists an Eulerian circuit $\mathcal{C}$ of G such that $\max_{u \in V(G)} d_{\mathcal{C}}(u,v) < k$ is NP-complete. We also give the following bounds on the Eulerian stretch of a digraph.

Proposition 1 ([3]) *Let G be a digraph with minimal degree δ, n vertices and m arcs. Then*

$$\frac{m}{\delta} - 1 \leq \mathcal{E}(G) \leq m - 2\delta - 3 \ .$$

Moreover, if G is a δ-regular digraph, with $\delta > 3$, then $n + 1 \leq \mathcal{E}(G)$.

Remarks: Note that Proposition 1 gives as lower bound $2p(p-1)-1$ for square meshes. Thus, when p is odd, the construction of Theorem 1 is very good. When p is even, we can find experimentally better Eulerian circuits. However, no systematic construction have been described yet.

2.1 Proof of Theorem 1: odd case

In this section, we consider the case of an odd size square grid having p^2 elements, and $p = 2q + 1$. Each node will be equivalently defined by its coordinates (i, j) or its label l, such that $l = i{*}p{+}j$, $0 \leq i, j < p$. For any node (i, j) in the mesh, we denote the arcs $((i, j), (i + 1, j))$ by N, $((i, j), (i - 1, j))$ by S, $((i, j), (i, j + 1))$ by E and $((i, j), (i, j - 1))$ by W; clearly, if the node belongs to the border of the mesh, one of these arc does not exists.

This notation allows us to describe path in the mesh with a source node and a sequence of directions. For example, the path $(0, 1, 2, 3)$ can be denoted by its first point 0, and the sequence E^3.

2.1.1 description of the Eulerian circuit

First of all, we need to described properly the Eulerian circuit, and then prove that it really correspond to an Eulerian circuit of the 2D-mesh of size p^2. Let decompose the circuit into 4 parts as follows:

- $\mathcal{P}_1$ starts at point 0 and the sequence of directions is:

$$\left(E^{p-1}NNSW^{p-1}N\right)^{q-2}\left(E^{p-1}NW^{p-1}N\right)E^{p-1}NWWNEE.$$

- $\mathcal{P}_2$ starts at point $p^2 - 1$ and the sequence of directions is:

$$WS^{p-1}WN^{p-2}WWNEEWS^{p-1}W\left(N^{p-2}WWNES^{p-1}W\right)^{q-2}.$$

- $\mathcal{P}_3$ starts at point 0 and the sequence of directions is:

$$\left(NE^{p-1}SWNNW^{p-2}\right)^{q-2}NE^{p-1}NSSWNNW^{p-2}NE^{p-1}N.$$

- $\mathcal{P}_4$ starts at point $p^2 - 1$ and the sequence of directions is:

$$SSWNNWS^{p-1}\left(WN^{p-1}WS^{p-1}\right)\left(WN^{p-1}EWWS^{p-1}\right)^{q-2}.$$

- Let $\mathcal{C} = \mathcal{P}_1\mathcal{P}_2\mathcal{P}_3\mathcal{P}_4$.

Assume first that the above description is valid, i.e., that the paths are correctly defined (they do not go *outside* the mesh). Then, we can make the following remarks.

1. The length of each path is exactly $p(p - 1)$;
2. $\mathcal{P}_3$ can be obtained from $\mathcal{P}_2$ by reversing the order of the directions and performing the following permutations: exchange N and W, and exchange S and E;
3. id. with $\mathcal{P}_1$ and $\mathcal{P}_4$.

It remains to show that $\mathcal{C} = \mathcal{P}_1\mathcal{P}_2\mathcal{P}_3\mathcal{P}_4$ defines an Eulerian circuit on the 2D-mesh.

Lemma 1 *Given the 2D square mesh of edge p, p odd, the circuit that begins at node 0, defined by $\mathcal{C}$ as above is a correctly defined Eulerian circuit of the mesh.*

Proof: As mentioned before, the total length of the circuit is exactly the number of arcs of the 2D mesh. from the description of the $\mathcal{C}$, it is clear that it describes an Eulerian circuit. □

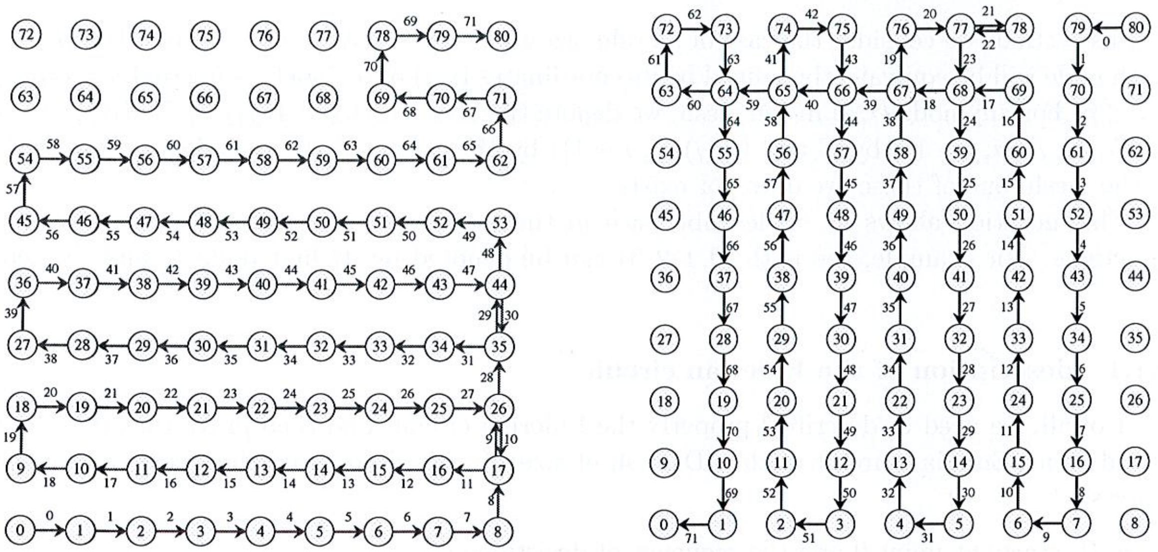


Figure 6: Parts $\mathcal{P}_1$ and $\mathcal{P}_2$ of the Eulerian circuit used in this proof, for $p = 9$.

2.1.2 properties of the circuit

From the previous description of $\mathcal{C}$, we are going to locate all the nodes in the different parts of the Eulerian circuit. Since the different parts have equal size, it will be easy to obtain the exact value of the stretch for any node.

Just recall that $\mathcal{S}_{\mathcal{C}}(i)$ is the maximal distance in $\mathcal{C}$ between two consecutive occurrences of the point i, and $\mathcal{S}_{\mathcal{C}}$ is the maximal value of this parameter over the points in the graph. We focus on this parameter since there exists a direct connection with the Eulerian stretch:

$$\mathcal{S}_{\mathcal{C}} = stretchW_{\mathcal{C}} + 1.$$

Before exploring all the possible position, we are going to divide the number of cases by 2 with the following lemma.

Lemma 2 *Let p an odd integer, and $\mathcal{C}$ the Eulerian circuit defined on the 2D square mesh of size p^2. For any node (i, j) of the mesh, we have:*

$$\mathcal{S}_{\mathcal{C}}((i,j)) = \mathcal{S}_{\mathcal{C}}((j,i))$$

Proof: We directly use the properties shown is the previous paragraph. They imply that, for any occurrence of the node (i, j) of the mesh in the $\mathcal{P}_1$ portion at position k, there exists an occurrence of the node (j, i) in the $\mathcal{P}_4$ portion at step $p(p-1) - k$. We have the same property with $\mathcal{P}_2$ and $\mathcal{P}_3$, and reversely ($\mathcal{P}_4$ with $\mathcal{P}_1$, and $\mathcal{P}_3$ with $\mathcal{P}_2$) as shown in Figure 7 below.

This imply that the maximal distance between two occurrences of node (i, j) is exactly the maximal distance between two occurrences of the node (j, i). □

Let examine all the cases of the mesh. In the left part, we show the locations of the points in the different parts of the Eulerian circuit $\mathcal{C}$; in the middle part, we complete the calculations and give

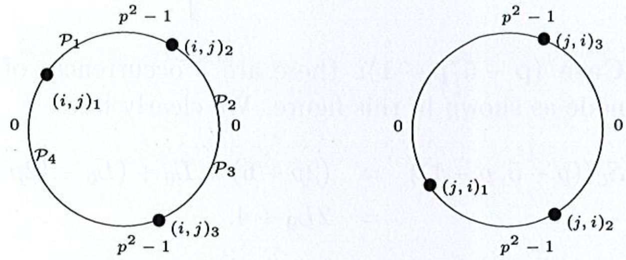
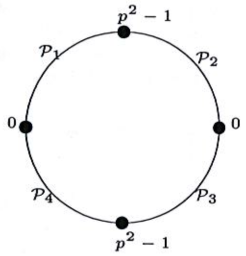


Figure 7: Positions of a node of the 2D-mesh (i,j) and its symmetric point (j,i) in the Eulerian circuit $\mathcal{C}$.

the values for the $\mathcal{S}_\mathcal{C}$ parameter in the studied point (at least an upper bound); finally, in the right part, we show the studied points in the 2D-mesh (example for $p=9$) in black (the white points show the points that have been studied in the previous cases). In the figures, we simply denote by x the studied points. Let L_0 be $p(p-1)$; considering $\mathcal{C}$, the length of each portion is thus exactly L_0.

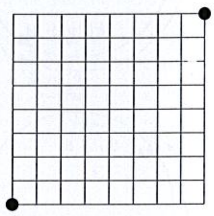


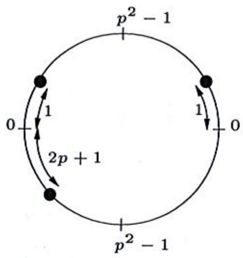
Case (0,0): As shown in the figure, the point 0 is the beginning of $\mathcal{P}_1$ and at the beginning of $\mathcal{P}_3$, thus for this point:

$$\mathcal{S}_\mathcal{C}((0,0)) = 2p(p-1) = 2L_0;$$

Case (p − 1, p − 1): as before, we have:

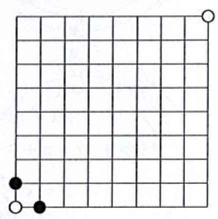
$$\mathcal{S}_\mathcal{C}((p-1,p-1)) = 2L_0.$$

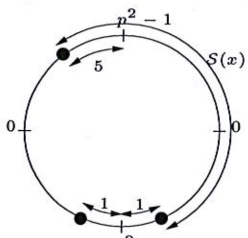




Case (0, 1): there are 3 occurrences of this node as shown in this figure. We clearly have:

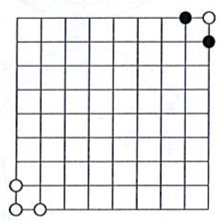
$$\mathcal{S}_\mathcal{C}((0,1)) = 2L_0 - 2.$$

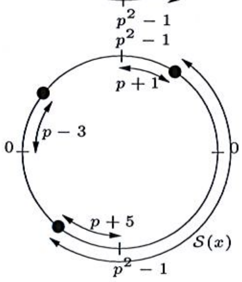




Case (p − 2, p − 1): there are 3 occurrences of this node as shown in this figure. We clearly have:

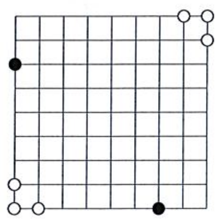
$$\mathcal{S}_\mathcal{C}((p-2,p-1)) = 5 + L_0 + (L_0 - 1) = 2L_0 + 4.$$

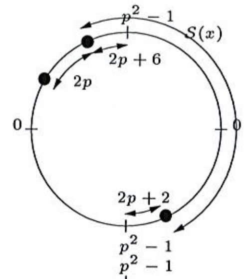




Case (0, p − 3): there are also 3 occurrences of this node as shown in this figure. We clearly have:

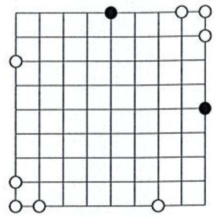
$$\begin{aligned}\mathcal{S}_\mathcal{C}((0,p-3)) &= (L_0-(p+1)) + L_0 + (p+5);\\ &= 2L_0 + 4.\end{aligned}$$

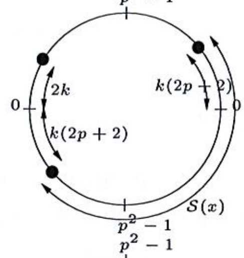




Case $(\mathbf{p-5,p-1})$: there are 3 occurrences of this node as shown in this figure. We clearly have:

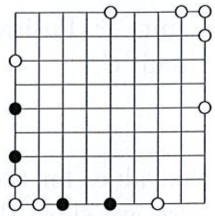
$$\begin{aligned}\mathcal{S}_{\mathcal{C}}((p-5,p-1)) &= (2p+6)+L_0+(L_0-(2p+2));\\ &= 2L_0+4.\end{aligned}$$

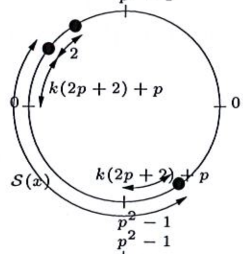




Case $(\mathbf{0,2k})$, $\mathbf{1\le k\le q-2}$: there are 3 occurrences of these nodes as shown in this figure. We clearly have:

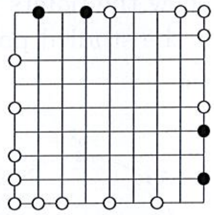
$$\mathcal{S}_{\mathcal{C}}((0,2k)) = 2L_0.$$

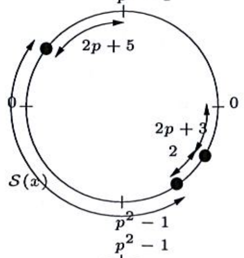




Case $(\mathbf{2k-1,p-1})$, $\mathbf{1\le k\le q-2}$: there are 3 occurrences of these nodes as shown in this figure. We clearly have:

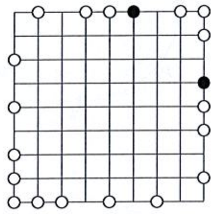
$$\mathcal{S}_{\mathcal{C}}((2k-1,p-1)) = 2L_0.$$

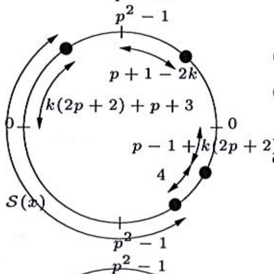




Case $(\mathbf{p-4,p-1})$: there are 3 occurrences of this node as shown in this figure. We clearly have:

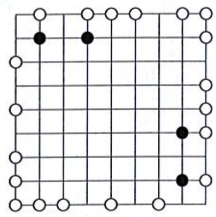
$$\mathcal{S}_{\mathcal{C}}((p-4,p-1)) = 2L_0.$$

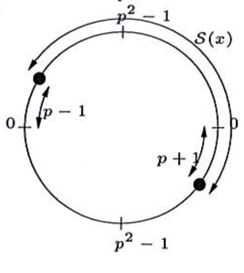




Case $(\mathbf{2k-1,p-2})$, $\mathbf{1\le k\le q-2}$: there are 4 occurrences of these nodes as shown in this figure. We clearly have:

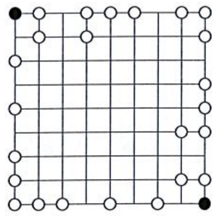
$$\begin{aligned}\mathcal{S}_{\mathcal{C}}((2k-1,p-2)) &= (p+3+k(2p+2))+L_0\\ &\quad +(L_0-(p-1+k(2p-2))-4);\\ &= 2L_0.\end{aligned}$$

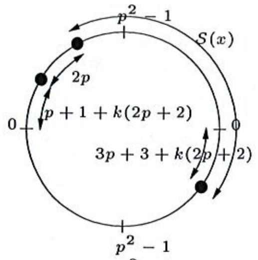




Case $(\mathbf{0,p-1})$: there are 2 occurrences of this node as shown in this figure. We clearly have:

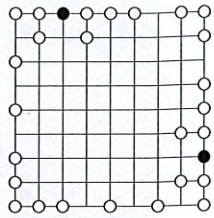
$$\begin{aligned}\mathcal{S}_{\mathcal{C}}((0,p-1)) &= (L_0-(p-1))+L_0+(p+1)\\ &= 2L_0+2.\end{aligned}$$

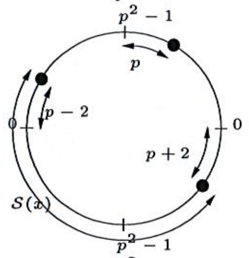




Case $(\mathbf{2k+2, p-1})$, $\mathbf{0 \le k \le q-4}$: there are 3 occurrences of these nodes as shown in this figure. We clearly have:

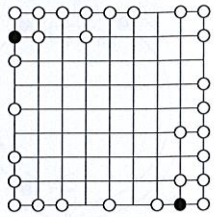
$$\begin{aligned}\mathcal{S}_{\mathcal{C}}((2k+2,p-1)) &= (L_0-(p+1+k(p+2))-2p) \\ &\quad +L_0+(3p+3+k(2p+2)) \\ &= 2L_0+2.\end{aligned}$$

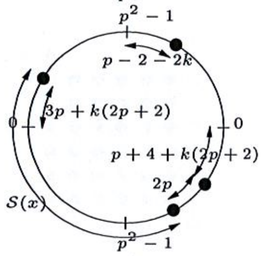




Case $(\mathbf{0, p-2})$: there are 3 occurrences of this node as shown in this figure. We clearly have:

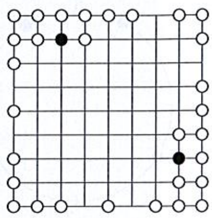
$$\mathcal{S}_{\mathcal{C}}(p^2-p-1)=2L_0-4.$$

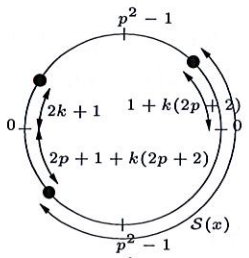




Case $(\mathbf{2k+2, p-2})$, $\mathbf{0 \le k \le q-4}$: there are 4 occurrences of these nodes as shown in this figure. We clearly have:

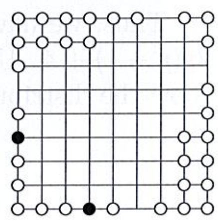
$$\begin{aligned}\mathcal{S}_{\mathcal{C}}((2k+2,p-2)) &= (3p+k(2p+2)+L_0 \\ &\quad +(L_0-2p-(p+4+k(2p+2)) \\ &= 2L_0-4.\end{aligned}$$

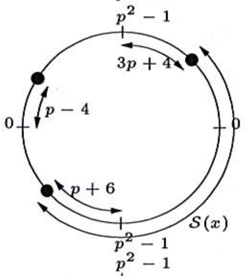




Case $(\mathbf{0, 2k+1})$, $\mathbf{1 \le k \le q-3}$: there are 3 occurrences of these nodes as shown in this figure. We clearly have:

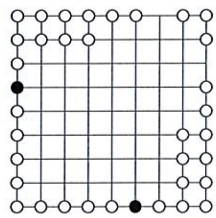
$$\begin{aligned}\mathcal{S}_{\mathcal{C}}((0,2k+1)) &= (1+k(2p+2)+L_0 \\ &\quad +(L_0-(2p+1+k(2p+2)) \\ &= 2L_0-2p.\end{aligned}$$

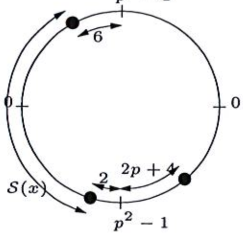




Case $(\mathbf{0, p-4})$: there are 3 occurrences of this node as shown in this figure. We clearly have:

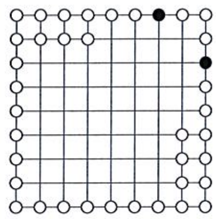
$$\begin{aligned}\mathcal{S}_{\mathcal{C}}((0,p-4)) &= p+6+L_0+(L_0-(3p+4)) \\ &= 2L_0-2p+2.\end{aligned}$$

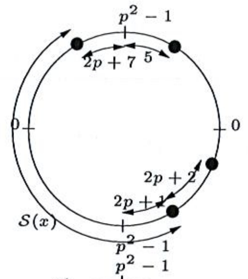




Case $(\mathbf{p-3, p-1})$: there are 3 occurrences of this node as shown in this figure. We clearly have:

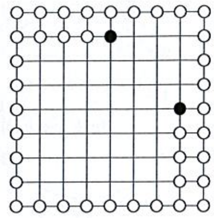
$$\mathcal{S}_{\mathcal{C}}((p-3,p-1))=2L_0-8.$$

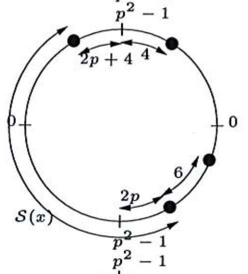




Case $(\mathbf{p-5, p-2})$: there are 4 occurrences of this node as shown in this figure. We clearly have:

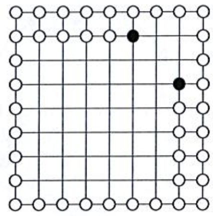
$$\mathcal{S}_{\mathcal{C}}((p-5,p-2)) = 2L_0 - 6.$$

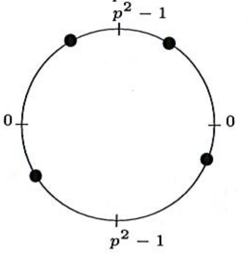




Case $(\mathbf{p-4, p-2})$: there are 4 occurrences of this node as shown in this figure. We clearly have:

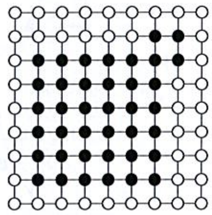
$$\mathcal{S}_{\mathcal{C}}((p-4,p-2)) = 2L_0 - 4.$$





Other cases: in this last cases, there exists one occurrence of each such node in each part of the Eulerian circuit. Thus, we have the inequality:

$$\mathcal{S}_{\mathcal{C}}(x) < 2L_0.$$



Thus, we have examined all the nodes of the 2D-mesh. The largest value found is $2L_0 + 4 = 2p(p-1) + 4$. Using the property of $\mathcal{S}_{\mathcal{C}}$, Theorem 1 is proven when p is odd. As an example, we show the distribution of the $\mathcal{S}_{\mathcal{C}}(x)$ for the 9×9 mesh in Figure 8. □

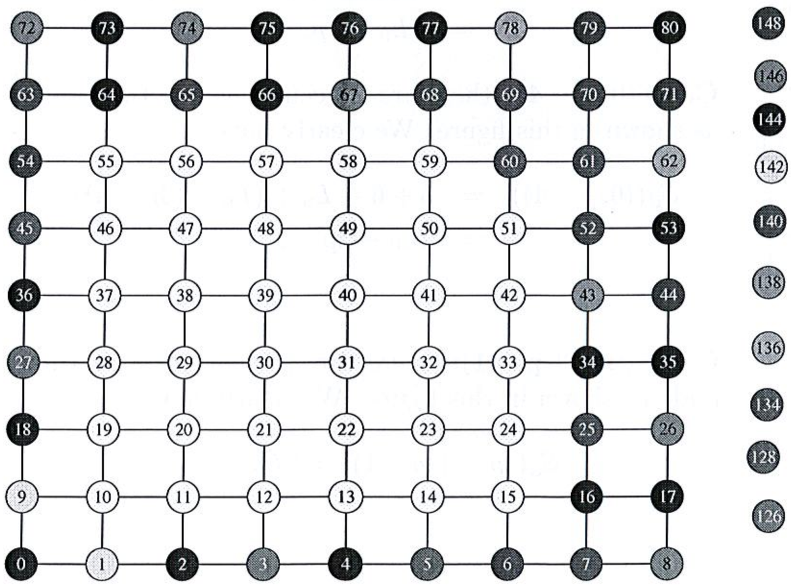


Figure 8: Values of the $\mathcal{S}_{\mathcal{C}}$ parameter for the most sensitive points of the mesh.

2.2 Proof of Theorem 1: even case

In this section, we simply describe the Eulerian circuit that leads to the desired result. Using the same techniques as in the previous section, the interested reader can complete the proof. It can be broken into merely 30 sub-cases.

As in the odd case, let consider $p = 2q$ an even integer. We are going to describe the Eulerian circuit in four different parts as follows:

- $\mathcal{P}_1$ starts at point 0 and the sequence of directions is:

$$(EN^{p-1}EWWSEES^{p-2})^{q-1}\,EN^{p-1};$$

- $\mathcal{P}_2$ starts at point $p^2 - 1$ and the sequence of directions is:

$$WSESW^{p-1}S\,(E^{p-1}SW^{p-1}S)^{q-2};$$

- $\mathcal{P}_3$ starts at point 0 and the sequence of directions is:

$$N^{p-1}ES^{p-1}EN^{p-1}E\,(S^{p-2}EESWWEN^{p-1}E)^{q-2};$$

- $\mathcal{P}_4$ starts at point $p^2 - 1$ and the sequence of directions is:

$$(SW^{p-1}SE^{p-1})^{q-2}\,SW^{p-1}SEEESWWW.$$

- Let $\mathcal{C} = \mathcal{P}_1\mathcal{P}_2\mathcal{P}_3\mathcal{P}_4$.

The 4 parts have not the same symmetry property as in the odd case. However, there exists some similarities between $\mathcal{P}_1$ and $\mathcal{P}_3$, and also between $\mathcal{P}_2$ and $\mathcal{P}_4$. The four parts are presented in Figures 9 and 10 below. Figure 11 shows the distribution of the $\mathcal{S}_\mathcal{C}(x)$ for most important points to be studied.

As a remark, there exists only few points that have a *large* value of $\mathcal{S}_\mathcal{C}$ parameter, i.e., equals to $2p(p-1) + 2p + k$, where k is a constant. Actually, it only corresponds to 3 cases: Point $(O, 1)$, Point $(0, 3)$ and Points $(p - 1 - 2\alpha, p - 1)$, $1 \leq \alpha \leq q - 1$. For all the other points, the $\mathcal{S}_\mathcal{C}$ parameter is of the form $2p(p-1) + k'$, for some constant k'.

2.3 Miscellaneous

Considering the Eulerian stretch, extensive search is clearly not a solution for enumerating eulerian circuits for graphs of reasonable size. However, it may be acceptable to search for *good* eulerian circuits, i.e., remove in the search all the beginning of circuits that will give bad things, whatever the remaining of the circuit is.

Using such paradigm, we enumerated all the *good* eulerian circuits in a 4 by 4 mesh. 9560 circuits with eulerian stretch equal to 26 were found. For each of them, we studied the shortcut diameter. The repartition is given in Figure 12 below. This figure shows that the amplitude of the shortcut diameters is merely twice the diameter of the graph. Figure 13 shows the distribution of the sum of the shortcut distances for this family of eulerian circuits. The best AEE obtained for this serie is 1.05 (to be compared to 1.03 in the table at the end of this paper. In order to appreciate the sparcity of good eulerian circuits, we have found 199160 eulerian circuits having eulerian stretch equal to 28. The distribution of the shortcut diameters is shown in Figure 14.

Based on previous results, we can conjecture that there exists very good eulerian circuits in terms of both eulerian diameter and eulerian shortcut diameter.

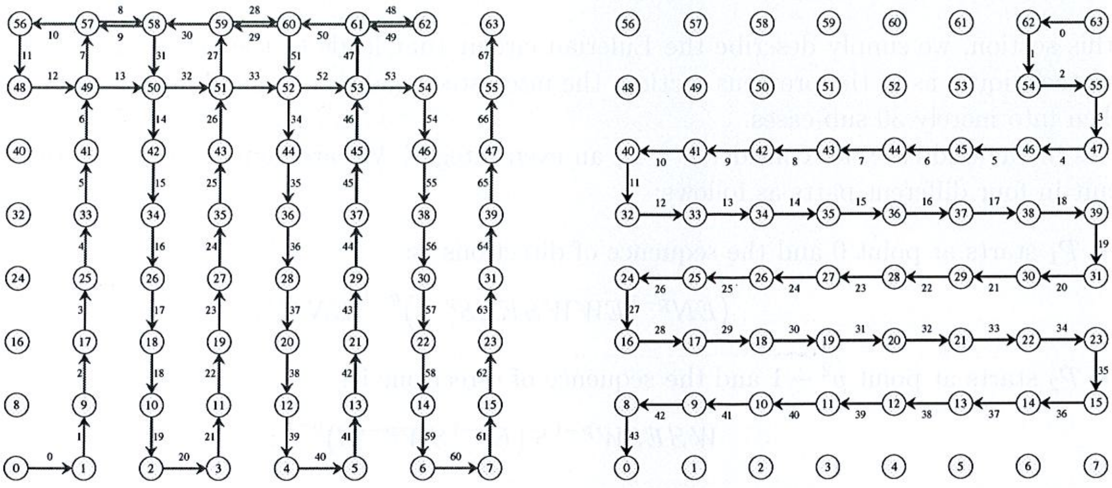


Figure 9: Two first parts of the Eulerian circuit studied in Section 2.2.

Conjecture 1 *In the mesh $M(p,p)$, there exists eulerian circuits $\mathcal{C}$ for which:*

$$\begin{cases} stretchW_{\mathcal{C}} & = & 2p(p-1)+1 \\ Sd_{\mathcal{C}} & = & 2(p-1) \end{cases}$$

This conjecture is statisfied for $p = 4$ as shown before. The first part of this conjecture is verified also for $p = 5$ with The eulerian circuit shown in Figure 15. This Eulerian circuit has the further proprety that there is no cycle of length 2. The main consequence is that the eulerian shortcut diameter is not very good (14). In $M(4,4)$, there remain only 61 circuits with this latter property, the best shortcut diameter found being 7.

3 About the itinerary index

3.1 General bounds

Proposition 2 *Let G a graph and $\mathcal{C}$ an Eulerian circuit of G. Let n be the number of vertices of G, D_G be its diameter and δ its smallest (in) degree. The following equation holds:*

$$D_G \leq Sd_{\mathcal{C}} \leq n - \delta.$$

Proof: Clearly, we have $D_G \leq Sd_{\mathcal{C}}$.

Let u and v be two vertices of G. Let us denote by $N^-(v)$ the set of in-neighbors of v. A shortcut path $\mathcal{P}(u,v)$ from u to v in $\mathcal{C}$ contains at most only one vertex of $N^-(v)$ (because a shortcut path from any vertex w of $N^-(v)$ to v is the arc (w,v)). Moreover, each vertex of G is contained at most once in $\mathcal{P}(u,v)$. Hence, $\mathcal{P}(u,v)$ contains at most one vertex of $N^-(v)$, vertex v itself and at most the $n - d(v) - 1$ other vertices (where $d(v)$ is the degree of v in G); i.e., at most $n - d(v) + 1$ vertices and at most $n - d(v)$ arcs. This means that $Sd_{\mathcal{C}}(u,v) \leq n - d(v) \leq n - \delta$ and that $Sd_{\mathcal{C}} \leq n - \delta$. □

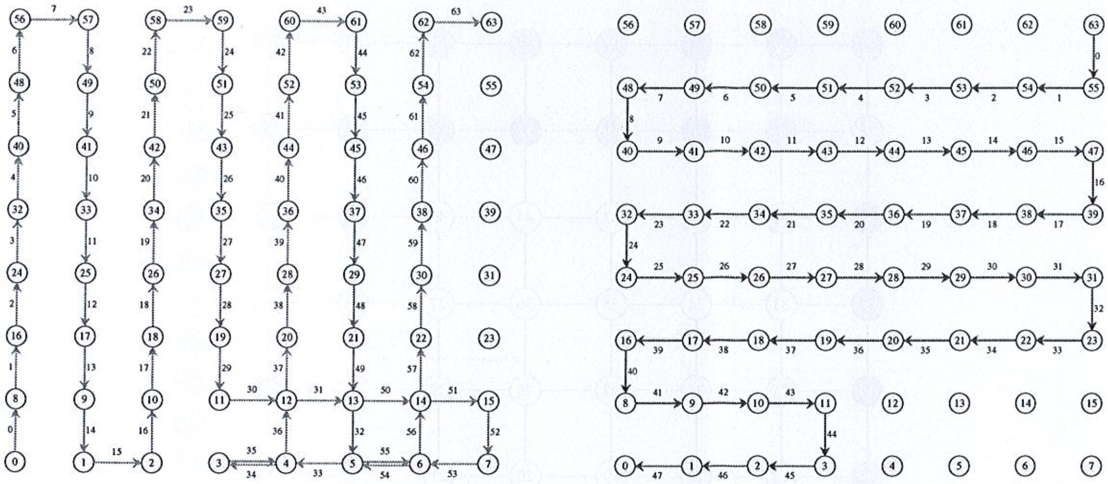


Figure 10: Two last parts of the Eulerian circuit studied in Section 2.2.

Note that these two bounds can be reached. The first one is easily obtain for the complete graph, and we will show that it is also true for square meshes in Corollary 2.1 below. The other bound is reached for the ring network.

Let G be a graph and $\mathcal{C}$ an Eulerian circuit of G. We describe now the scheme of an algorithm that computes $Sd_{\mathcal{C}}(u,v)$ for each u in G and for a given vertex v. The main idea is to start form vertex v and, at each step, to visit a level of $\mathcal{C}$. A level is the set of vertices w such that there is an Eulerian segment from w to v of a given length. The levels of $\mathcal{C}$ are visited in reverse order, from v.

For each vertex w visited for *the first time* in this process is the first extremity of the shortest Eulerian segment of $\mathcal{C}$ from w to v. Hence $Sd_{\mathcal{C}}(u,v)$ is 1 plus the shortcut distance from w' to v where w' is the successor of w on $\mathcal{C}$ (so $Sd_{\mathcal{C}}(u,v)$ has already been calculated in a previous step). When all the vertices have been reach, the process can be stopped.

We can note that we compute the shortcurt dstances in the reverse order of the classical diameter computations: we usually compute (using, for exemple, the Dijsktra algorithm) the distance for going from a point to reach any point of the graph; whereas in this cshortcut computation, we compute all the distances to go from any vertex to a given vertex of the graph.

We must note here that several technical difficulties must be taken into account.

- During the reverse search from v, one (or more) segment(s) can "return" to v. In this case, cut the search on this (these) segment(s) for the following steps (otherwise this is equivalent to restart the search on an already visited segment).

- The first time a non-visited vertex v is discovered, many occurrences of the same vertex v can be discovered at the same step, on different segment of $\mathcal{C}$. In this case, according to the definition, $Sd_{\mathcal{C}}(u,v)$ is the minimum of $1 + Sd_{\mathcal{C}}(u',v)$ for each vertex u' successor of u in each segment of $\mathcal{C}$.

This technique leads to a linear procedure to compute $Sd_{\mathcal{C}}(u,v)$ for each u and a given v. Hence, $Sd_{\mathcal{C}}$ can be efficiently computed by computing $Sd_{\mathcal{C}}(u,v)$ for each pair u and v in G.

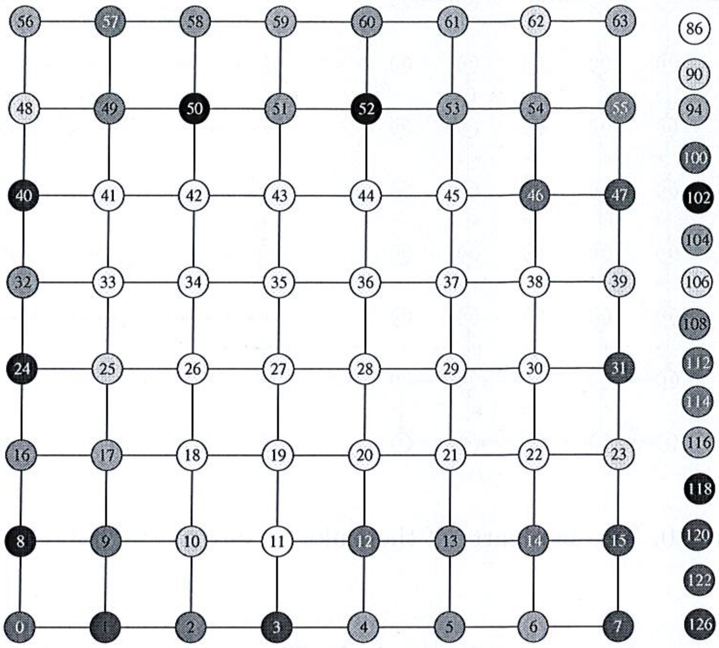


Figure 11: Distribution of the $\mathcal{S}_\mathcal{C}(x)$ when $p = 8$ in the Eulerian circuit studied in Section 2.2.

Figure 16 shows an example of calculation for $Sd_\mathcal{C}(u, b)$. The 3 occurrences of b are represented in solid box and the first visited occurrences of each vertex is represented in a dashed box. The corresponding integer is the shortcut distance to b. We can note here that vertex d has two occurrences in level 2.

The following lemma and definitions give useful tools to obtain bounds or exact values of the itinerary index of any graph.

Lemma 3 *Let $\mathcal{C}$ be an Eulerian circuit in a graph G. Let us consider 2 vertices u and v such that $best_\mathcal{C}(u, v) = dist_G(u, v)$ on the portion $\mathcal{P}$ of $\mathcal{C}$ then $Sd_\mathcal{C}(u, v) = dist_G(u, v)$.*

Proof: First, if u and v are neighbors in G, the lemma clearly holds.

Assume that the property holds for any pair of nodes (a, b) such that $best_\mathcal{C}(u, v) = dist_G(u, v) = k - 1$. Let consider a pair (u, v) such that $best_\mathcal{C}(u, v) = dist_G(u, v) = k$.

Assume first that there exists only one portion $\mathcal{P}$ of $\mathcal{C}$ such that $best_\mathcal{C}(u, v) = dist_G(u, v)$. Let us denote w the neighbor of u on $\mathcal{P}$. Considering the shortcut strategy, any message being in u and going to v should go to vertex w in $\mathcal{P}$. Then, we have:

$$Sd_\mathcal{C}(u, v) = 1 + Sd_\mathcal{C}(w, v)$$

The part going from w to v in $\mathcal{P}$ is also a shortest path from w to v of length $k - 1$.

Thus, using the recurrence hypothesis, the property holds for (u, v). If there exist several portions of $\mathcal{C}$ of length $best_\mathcal{C}(u, v)$, we can apply the same strategy on any of them, leading to the desired result.

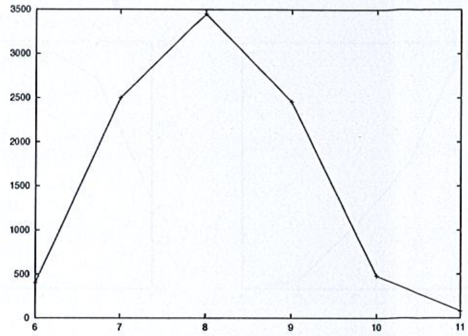


Figure 12: Distribution of the shortcut diameters for eulerian circuits with eulerian stretch of 26 in M(4,4).

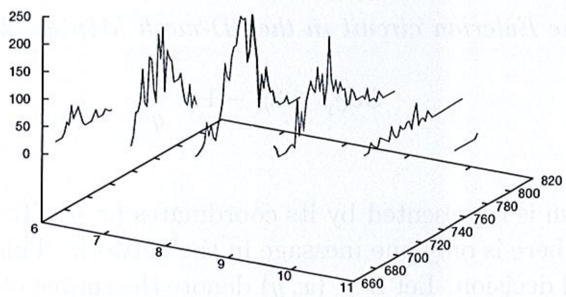


Figure 13: Distribution of the sum of distances eulerian circuits with eulerian stretch of 26 in M(4,4).

Definition 1 *Let us consider an Eulerian circuit $\mathcal{C}$ in a graph G. Let us consider two vertices u and v. A portion of $\mathcal{C}$ from u to v, without any occurrence of v and with at least one occurrence of u, is called a* ***dead-end*** *from u to v.*

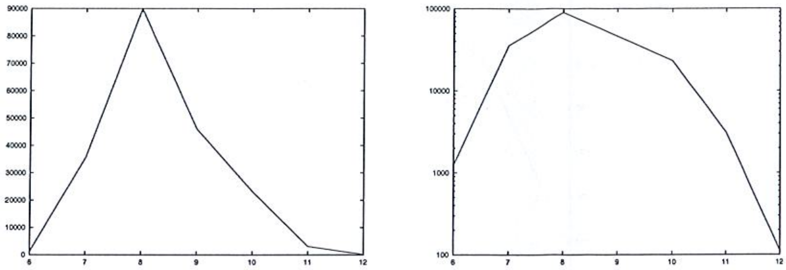
Definition 2 *Let us consider an Eulerian circuit $\mathcal{C}$ in a graph G. Let us consider two vertices u and v and a path $\mathcal{P}$ with shortcuts from u to v. Let w be any vertex of $\mathcal{P} = uu_1 \ldots u_l w \ldots v$. An* ***indirect dead-end*** *from w to v is a portion of $\mathcal{C}$ from w to v with at least one vertex $u, u_1, \ldots, u_l, w$.*

Remark. When looking for a shortcut between u and v, we do not have to consider any portion of $\mathcal{C}$ leading to a dead-end or to an indirect dead-end from u to v.

Moreover, to determine the length of a shortcut path between the origin and the final destination of the message, we will consider an ideal case, that is the case where the message is alone in the network. We also always suppose that the message will always do the best choices of routes.

3.2 Itinerary index of the 2D-mesh

We now describe an Eulerian circuit in the 2D-mesh, where the shortcuts diameter is optimal in the square mesh ($M(p,p)$). In fact, we will study the general case of $M(p,q)$. The Eulerian circuit



(a) classical axes (b) logscale y axe

Figure 14: Distribution of the shortcut diameters for eulerian circuits with eulerian stretch of 28 in M(4,4).

we have considered can be split in two parts (A and B) as shown on Figure 17.

Theorem 2 *Let C_1 be the Eulerian circuit in the 2D-mesh $M(p,q)$, $2 \leq q \leq p$ described above.*

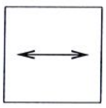
$$Sd_{C_1} \leq 2p - 1 - \frac{p}{q}.$$

Proof:

Any vertex in the mesh is represented by its coordinates (x, y). To prove the result, we consider an ideal situation where there is only one message in the network. This message is supposed to take at each step the best local decision. Let $S = (x, y)$ denote the source of the message and $D = (x', y')$ the final destination of the message.

In the remaining, we distinguish 4 major cases:

1. Source and destination are on the same row or on the same column;
2. $\{x < x'$ and $y > y'\}$ or $\{x > x'$ and $y < y'\}$: message goes fom *South-West* to *North-East* or vice versa;
3. $\{x > x'$ and $y > y'\}$: message goes from *South-East* to *North-West*;
4. $\{x < x'$ and $y < y'\}$ message goes from *North-West* to *South-East*.



Case 1: $\{\mathbf{y = y'}\}$:
If $[x > x'$ and $y \neq 0]$ or $[x < x'$ and $y \neq 0]$ then Lemma 3 applies directly. In the first case, the message goes West using the part B of C_1, and in the second case it goes East.
If $x < x'$ and $y = 0$ then the message goes East using the part A of the circuit (the 2 others directions are dead-ends). It will go through exactly $x' - x - 1$ shortcuts. The path used is a shortest path.
If $x > x'$ and $y = 0$ then Lemma 3 also applies directly. The message goes West using the part A of the circuit C_1.

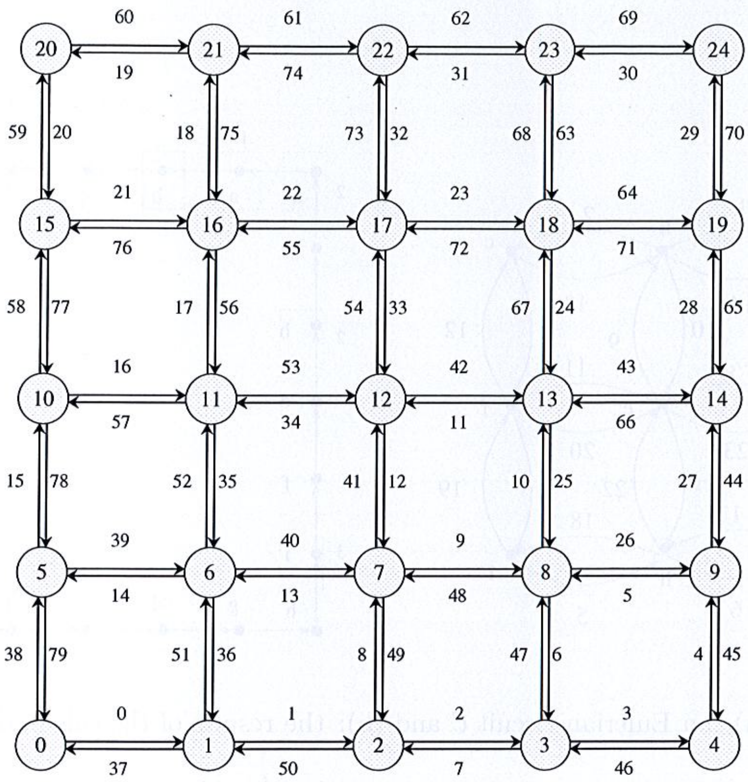
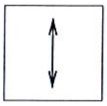
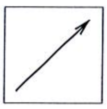


Figure 15: Eulerian circuit in $M(5,5)$ with optimal stretch



Case 2: $\{\mathbf{x = x'}\}$**:**
If $[y < y'$ and $x \neq 0]$ or $[y > y'$ and $x \neq 0]$ then Lemma 3 applies. In the first case, the message goes South using the part A of $\mathcal{C}_1$, and in the second case it goes North. If $y < y'$ and $x = 0$ the message goes South using the part B of the circuit (others directions are dead-ends). One can note that it uses $y' - y - 1$ shortcuts, but the path used is not a shortest path.
If $y > y'$ and $x = 0$ then Lemma 3 applies and the message goes North using the part B of the circuit.



Case 3: $\{\mathbf{x < x' | y > y'}\}$**:**
Source can send the message in the four directions. All of these are dead-ends but the North.
The message will be closer than the final destination (y will decrease, there is only one direction North that is not a dead-end or an indirect dead-end). Once the message is arrived in vertex (x, y') (same row as D), the situation is the same as Case 1. The message has used only one shortcut except $y' = 0$. In all cases, the message uses a shortest path.

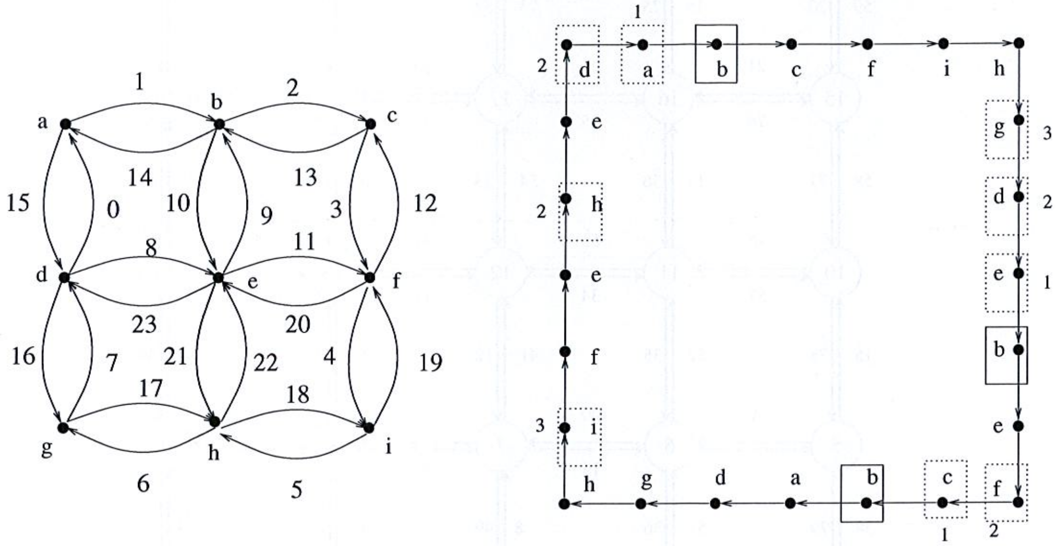


Figure 16: (a): an Eulerian circuit $\mathcal{C}$ and (b): the results of the calculations of $Sd_{\mathcal{C}}(u, b)$.

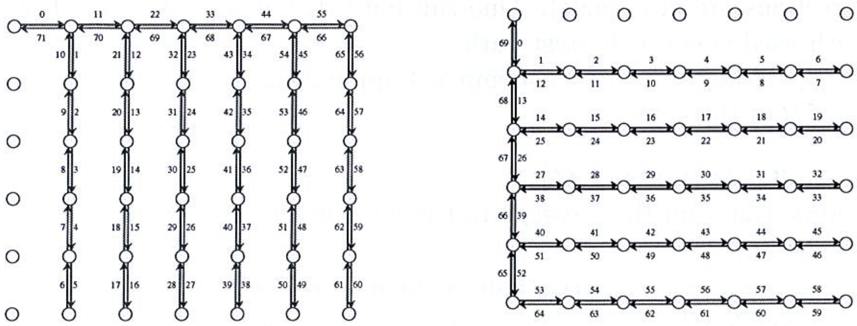
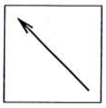


Figure 17: The two parts of the Eulerian circuit for $q = 7$ and $p = 6$

Case 4: $\{\mathbf{x > x' | y < y'}\}$:
Source can send the message in the four directions. All of these are dead-ends but the West.

The message will be closer than the final destination (x will decrease, there is only one direction West that is not a dead-end or an indirect dead-end). Once the message is arrived in vertex (x', y) (same row as D), the situation is the same as in Case 2. Message has used only one shortcut except of $x' = 0$. In all cases, the message uses a shortest path.

Case 5: $\{\mathbf{x > x' | y > y'}\}$:
We will describe analytically the length of the portion of $\mathcal{C}_1$ in different cases. In fact, there are always 2 possibilities because East and South are dead-ends.



The two possible directions (North and West) bring the message closer to the destination. At each new vertex the same choice of directions is proposed, except if the message has reached the right row (or column) (See Cases 1 and 2). In all cases, the shortcut path is a shortest path.

Case 6: $\{\mathbf{0 < x < x' | 0 < y < y'}\}$:
We can see first that East and South are dead-ends. There are only two possibilities: North or West.
We will compute the length of the portion on the part A denoted by L_A:
- y steps to come back to row 0;
- $x' - x$ steps from row 0 to East;
- $2(p-1)(x'-x)$ rows to East;
- y' steps to go to the destination.

Figure 18 shows the path of length L_A. Thus, we have:

$$\begin{aligned} L_A &= y + (x' - x) + 2(p-1)(x'-x) + y' \\ &= y + y' + (2p-1)x' - (2p-1)x \end{aligned}$$

by symmetry, we get:

$$L_B = x + x' + (2q-1)y' - (2q-1)y$$

We are interested in the sign of the difference between these two lengths in order to know which part of the circuit $\mathcal{C}_1$ is the shortest:

$$\begin{aligned} L_A - L_B &= y + y' + (2p-1)x' - (2p-1)x - x - x' - (2q-1)y' + (2q-1)y \\ &= 2qy - 2(q-1)y' + 2(p-1)x' - 2px \end{aligned}$$

It is sufficient to study the sign of $qy - (q-1)y' + (p-1)x' - px$.

$\mathbf{L_A - L_B < 0}$. We use the part A of $\mathcal{C}_1$. In this case, shortcut paths are not shortest paths. The first arc used is the North one and it is not on a shortest path between S and D on the mesh. The shortcut path between S and D goes to the North until row 0. From the source North, arcs East and South are dead-ends and arc West is an indirect dead-end.

The shortcut path will then goes East until the column containing D. From this particular point, the message falls in Case 2. Hence, $x' - x + y + y'$ is the length of the path between S

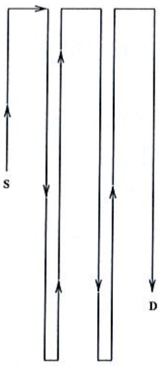


Figure 18: Path of length L_A when $x < x'$ and $y < y'$

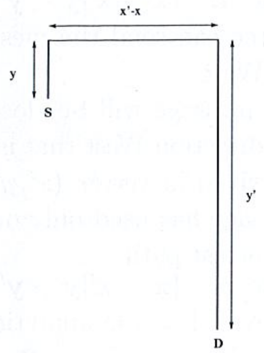


Figure 19: path with shortcuts

and D. Figure 19 shows this path. Let us prove that $x' - x + y + y' \leq 2p - 1 - \frac{p}{q}$.

$$\begin{aligned}
\frac{1}{2}(L_A - L_B) &= p(x'-x) + q(y-y') + (y'-x') < 0 \\
q(x'-x+y-y') + (y'-x') &\leq p(x'-x) + q(y-y') + (y'-x') < 0 \\
q(x'-x+y+y') &< x' + (2q-1)y'
\end{aligned}$$

However, $x' \leq q-1$ and $y' \leq p-1$ hence:

$$\begin{aligned}
q(x'-x+y+y') &\leq (q-1) + (2q-1)(p-1) \\
q(x'-x+y+y') &\leq q - 1 + 2pq - 2q - p + 1 \\
q(x'-x+y+y') &\leq 2pq - q - p \\
x' - x + y + y' &\leq 2p - 1 - \frac{p}{q}
\end{aligned}$$

$\mathbf{L_A - L_B > 0.}$ We use the part B of the circuit. As in the preceding case, the message from S doesn't use a shortest path, we must study the length of a shortcut path in the mesh. This case is the same as the previous one. It is sufficient to change rows in columns and vice-versa. The length of a shortcut path is: $x + x' + y' - y$.

$$\begin{aligned}
\frac{1}{2}(L_A - L_B) &= p(x'-x) + q(y-y') + (y'-x') > 0 \\
p(x'-x+y-y') + (y'-x') &\geq p(x'-x) + q(y-y') + (y'-x') > 0 \\
p(x-x'-y+y') &< y' - x' \\
p(x+x'+y'-y) &< y' - x' + 2px' \\
p(x+x'+y'-y) &< y' + (2p-1)x'
\end{aligned}$$

However, $x' \leq q-1$ and $y' \leq p-1$ hence:

$$\begin{aligned}
p(x+x'+y'-y) &\leq p - 1 + (2p-1)(q-1) \\
p(x+x'+y'-y) &\leq p - 1 + 2pq - 2p - q + 1 \\
p(x+x'+y'-y) &\leq 2pq - p - q \\
x + x' + y' - y &\leq 2q - 1 - \frac{q}{p}
\end{aligned}$$

$\mathbf{L_A - L_B = 0.}$ The shortcut path between S and D in C_1 will be contained completely in the part A or in the part B of C_1, in function of the first arc chosen. In these two cases, we are the 2 situations described above. If we take the North, the length of the path with shortcut is:

$$1 + (y - 1) + x' - x + y'$$

with an initial condition

$$\frac{1}{2}(L_A - L_B) = p(x' - x) + q(y - y') + (y' - y) = 0$$

If we use same computation as above, we have:

$$x' - x + y + y' \leq 2p - 1 - \frac{p}{q}.$$

So, a shortcut path between S and D is of length at most: $2p - 1 - \frac{p}{q}$.

Case 7: $\mathbf{x = 0, y \neq 0}$**:**

East is a dead-end. The length of the east portion (L'_B) may be compute in the same way as L_B in the general case. The length of the North portion (L'_A)may be compute in the same way as L_A in the general case. We get:

$$\begin{aligned} L'_A &= y + y' + (2p - 1)x' \\ L'_B &= x' + (2q - 1)y' - (2q - 1)y \end{aligned}$$

The analysis of the sign and the length of paths are the same as the general case. In the case where $L'_A - L'_B > 0$, we must note that the arc chosen will be the South one and not the West one.

Case 8: $\mathbf{x \neq 0,\ y = 0}$**:**

The south is a dead-end. The length of the West portion (L'_B) may be compute in the same way as L_B in the general case. The length of the East portion (L'_A) may be computed in the same way as L_A in the general case. We get:

$$\begin{aligned} L'_A &= y' + (2p - 1)x' - (2p - 1)x \\ L'_B &= x + x' + (2q - 1)y' \end{aligned}$$

The analysis of the sign and the length of paths are the same as the general case. In the case where $L'_A - L'_B < 0$ we must note that the arc chosen will be the East one and not the North one.

Case 9: $\mathbf{x = 0,\ y = 0}$**:**

It is the union of the two previous cases. Directions North and West do not exist. Path can only use South and East.

Considering all the cases of the 2D-mesh, we obtained the desired result, i.e., the length of shortcuts paths is at most $2p - 1 - \frac{p}{q}$. □

The previous theorem, combined with the fact that for any Eulerian circuit C', $Sd_{C'}$ is always greater than or equal to the (graph) diameter of the mesh (i.e. $2p - 2$ for $M(p, p)$), leads to the following corollary.

Corollary 2.1

$$\mathcal{I}(M(p,p)) = 2p - 2$$

Note that in the special case of the 2D-mesh $M(p,p)$, $Ad_{M(p,p)} = \frac{2}{3}(p^5 - p^3)$. We have summarized some results about the average eulerian efficiency parameter in the following table. In this table, $\mathcal{C}_1$ is the Eulerian circuit constructed like on Figure 17. $\mathcal{C}_2$ is another Eulerian circuit we have defined.

		$M(25,25)$	$M(45,45)$	$M(101,101)$
$Diameter = D(M(p,p))$		48	88	200
$ADist_{M(p,p)}$		16.64	29.98	67.32
$\mathcal{C}_1$	$Sd_{\mathcal{C}_1}$	48	88	200
	$EE_{\mathcal{C}_1}$			
	$AEE_{\mathcal{C}_1}$	1.16	1.19	1.19
$\mathcal{C}_2$	$Sd_{\mathcal{C}_2}$	48	88	200
	$EE_{\mathcal{C}_2}$			
	$AEE_{\mathcal{C}_2}$	1.03	1.03	1.03

4 Concluding remarks

As in the case of the grid, it seems difficult to find an Eulerian circuit $\mathcal{C}$ in any digraph G such that $StrechW_{\mathcal{C}} = \mathcal{E}(G)$ and $Sd_{\mathcal{C}} = \mathcal{I}(G)$. In [3], we have given some digraphs G where $\mathcal{E}(G)$ is close to $(m/\delta) - 1$.

Given an Eulerian circuit $\mathcal{C}$ in a digraph G, it is clear that $stretchW_{\mathcal{C}}$ is the interesting parameter to focus on if the network is not too loaded, otherwise we have to deal with $Sd_{\mathcal{C}}$. We have now to see how to find "good" Eulerian routing strategies in a digraph from these two points of view.

To end this paper about Eulerian routing parameters in a graph G, we make a remark on the size of the routing table located in each node. An Eulerian circuit consists in labeling the arc of G from 0 to $m - 1 = |A(G)| - 1$; each vertex v knows the labels of its outgoing arcs. Thus, to know what are the interesting shortcuts for a packet reaching vertex v with destination w, v has to know all the labels of the incoming arcs in w, i.e., at most $\Delta(G)$ integers between 0 and $m - 1$ for each vertex. Thus, the size of the routing table in each node is $n \cdot \Delta(G) \cdot log_2\, m$ bits, where $n = |V(G)|$. It is an open problem to know if the size of such a routing table can be decreased.

References

[1] P. Baran. On distributed communication networks. *IEEE Transactions on Communication Systems*, CS-12, 1964.

[2] D. Barth. Une approche algorithmique du routage dans les réseaux de télécommunication. Mémoire d'habilitation n. 351 (in French), 1998. Université de Paris-Sud.

[3] D. Barth, P. Berthomé, and J. Cohen. The Eulerian stretch of a digraph and the ending guarantee of a convergence routing. Technical Report A00-R-400, LORIA-Nancy, 2000. `http://www.loria.fr/~jcohen/publications.html`.

[4] J. C. Brassil and R. L. Cruz. Bounds on maximum delay in networks with deflection routing. *IEEE Transactions on Parallel and Distributed Systems*, 6(7):724–732, 1995.

[5] T. Chich. *Optimisation du routage par déflexion dans les réseaux de télécommunications métropolitains.* PhD thesis, ENS-Lyon, December 1997.

[6] U. Feige. Observations on hot potato routing. In *ISTCS: 3rd Israeli Symposium on the Theory of Computing and Systems*, 1995.

[7] U. Feige and R. Krauthgamer. Networks on which hot-potato routing does not livelock. *Distributed Computing*, 13:53–58, 2000.

[8] J. Jackson. Compatible Euler tours for transition systems in Eulerian graphs. *Discrete Mathematics*, 66:127–131, 1987.

[9] C. Kaklamanis, D. Krizanc, and S. Rao. Hot-potato routing on processor arrays. In *ACM Symposium on Parallel Algorithms and Architectures*, pages 273–282, 1993.

[10] A. Mayer, Y. Ofek, and M. Yung. Local fairness in general-topology networks with convergence routing. In *Infocom*, volume 2, pages 891–899. IEEE, June 1995.

[11] F. Meyer auf der Heide and M. Westermann. Hot-potato routing on multi-dimensional tori. In M. Nagl, editor, *Graph Theoretical Concepts in Computer Science (WG)*, volume 1017 of *Lecture Notes in Computer Science*, pages 209–221, 1995.

[12] Y. Ofek and M. Yung. Principles for high speed network control: loss-less and deadlock-freeness, self-routing and a single buffer per link. In *ACM Symposium On Principles of Distributed Computing*, pages 161–175, 1990.

[13] A. Schuster. *Optical Interconnections and Parallel Processing: The Interface*, chapter Bounds and analysis techniques for greedy hot-potato routing, pages 284–354. Kluwer Academic Publishers, 1997.

[14] E.A. Varvarigos and P.J. Lang. A virtual circuit deflection protocol. *IEEE/ACM Transactions on Networking*, 7(3):335–349, June 1999.

[15] B. Yener, T. Boult, and Y. Ofek. Hamiltonian decompositions of regular topology networks with convergence routing. Technical Report CUCS-011094, Columbia University, Computer Science Department, 1994.

[16] B. Yener, S. Matsoukas, and Y. Ofek. Iterative approach to optimizing convergence routing priorities. *IEEE/ACM Transactions on Networking*, 5(4):530–542, August 1997.

RAPPORTS INTERNES AU LRI - ANNEE 2003

N°	Nom	Titre	Nbre de pages	Date parution
1345	FLANDRIN E LI H WEI B	A SUFFICIENT CONDITION FOR PANCYCLABILITY OF GRAPHS	16 PAGES	01/2003