On the Compactness of Bounded Degree Graphs for Shortest Path Interval Routing

Cyril Gavoille[1] and Eric Guévremont[2]

[1]Laboratoire de l'Informatique du Parallélisme
École Normale Supérieure de Lyon
69364 Lyon cedex 07, France
(*gavoille@lip.ens-lyon.fr*)

[2]School of Computing Science
Simon Fraser University
Burnaby, B.C., V5A 1S6, Canada
(*ericg@cs.sfu.ca*)

Abstract. Consider *shortest path interval routing*, a popular memory-balanced method for solving the routing problem on arbitrary networks. Given a network G, let $\text{IRS}(G)$ denote the maximum number of intervals necessary to encode groups of destinations on an edge, minimized over all shortest path interval routing schemes on G. In this paper, we establish a worst case bound on $\text{IRS}(G)$. More precisely for every integer n and for every integer $d \geqslant 3$, we construct a n-node network G of maximum degree d with $\text{IRS}(G) = \Omega(n/\log^2 n)$.

1 Introduction

The *shortest path routing problem* for an arbitrary network of processors is to design a uniform strategy that the router of each processor will follow, upon reception of a message, to decide to which of its neighboring nodes the message should be sent next, such that the message arrives at its destination after passing through as few nodes as possible. The routing strategy should be simple and distributed so as to limit the costs of routing (space, time and complexity) and uniform to reduce the costs of building hardware routers, over a potentially great number of nodes. We are interested in minimizing the local memory requirements for a distributed routing strategy.

Table routing is a standard solution to the shortest path routing problem for arbitrary networks. Each node in the network stores a table that lists, for each possible destination, the output port that should be used to send a message to the destination along a shortest path. That solution guarantees shortest paths but requires $\Theta(n \log d)$ bits of space per node, where n is the number of nodes and d is the maximum degree of the node.

To alleviate the space requirements of routing tables, *compact routing schemes* were introduced: in [SK85] for arbitrary networks and in [FJ88, FJ89, FJ90] for planar and c-decomposable networks. A popular compact routing method, *interval routing*, is to group together in intervals the destination nodes corresponding to the same output port of a given node. Just as for table routing, this method requires that a header of $O(\log n)$ bits be added to the forwarded message. This routing scheme was introduced in [SK85], generalized in [vLT87] and shortest path interval routing was discussed in [vLT87, FJ88, BvLT91].

Let us model a network of processors as a connected, simple and loop-less symmetric digraph $G = (V, E)$, where V denotes the set of vertices of G (corresponding to the routers) and E the set of arcs of G (corresponding to the set of directed links of the network). We assume that the distance between two routers connected by an arc is uniform. An *interval* $[a, b]$ of the set $\{1, \ldots, n\}$ is the set of consecutive integers $\{a, \ldots, b\}$ cyclically. For example, interval $[7, 2]$ is the subset $\{7, 8, 1, 2\}$ of $\{1, \ldots, 8\}$. In the following, we use the term *graph* to refer to a connected, simple and loop-less symmetric digraph.

The formal definition of shortest interval routing scheme is the following:

Definition 1 (Interval Routing Scheme). Given a graph $G = (V, E)$ of n vertices, an *interval routing scheme* $R = (\mathcal{L}, \mathcal{I})$ on G consists of:

1. a one-to-one labeling function of the vertices, $\mathcal{L} : V \to \{1, \ldots, n\}$
2. a family $\mathcal{I} = \{I_e \mid e \in E\}$, where I_e is the union of the intervals of $\{1, \ldots, n\}$ associated with arc e.

Moreover $\mathcal{L}$ and $\mathcal{I}$ must be such that the following properties hold:

1. for every $x \in V$, $\mathcal{L}(x) \cup \{I_{(x,y)} \mid (x, y) \in E\} = \{1, \ldots, n\}$
2. for every two distinct arcs (x, y) and (x, z) of E, $I_{(x,y)} \cap I_{(x,z)} = \emptyset$

If an interval routing scheme R is defined on a graph G, then message routing is performed as follows: upon reception of a message, vertex x first compares the message header, $\mathcal{L}(y)$ with its own label, $\mathcal{L}(x)$, to check if the message has arrived at its destination. If not, then the message and its header are forwarded through the unique arc (x, z) such that $\mathcal{L}(y) \in I_{(x,z)}$.

We say that an interval routing scheme R is a *shortest path* routing scheme if the node-to-node routes induced by R always use a shortest path in G. Interval routing schemes that do not require shortest paths have been studied in [SK85, BvLT91, FG94a], where the authors give a complete characterization of the graphs requiring a small number of intervals for different restricted versions of interval routing. From now on, we will only consider shortest path interval routing schemes.

Consider Figure 1 as an example of a shortest path interval routing scheme. In this example, the labeling function $\mathcal{L}$ maps vertices a, b, e, g, d, f, c to integers $1, \ldots, 7$ respectively. In this graph, which is shown with undirected edges, the set of intervals $I_{(x,y)}$ assigned to arc (x, y) is placed close to vertex x. For example, the intervals $[1, 2]$ and $[5]$ that are close to vertex 7 correspond to the set of integers $\{1, 2, 5\}$, and form the set $I_{(7,1)}$ that labels the arc $(7, 1)$. Accordingly, if the vertex b wants to send a message to vertex f under R, the message will follow arcs (b, a), (a, c) and (c, f) in order. Note that in this model, the label of the node may or may not belong to an interval of one of its outgoing arcs (e.g. for this graph, the node labeled 5 does, while the node labeled 3 does not).

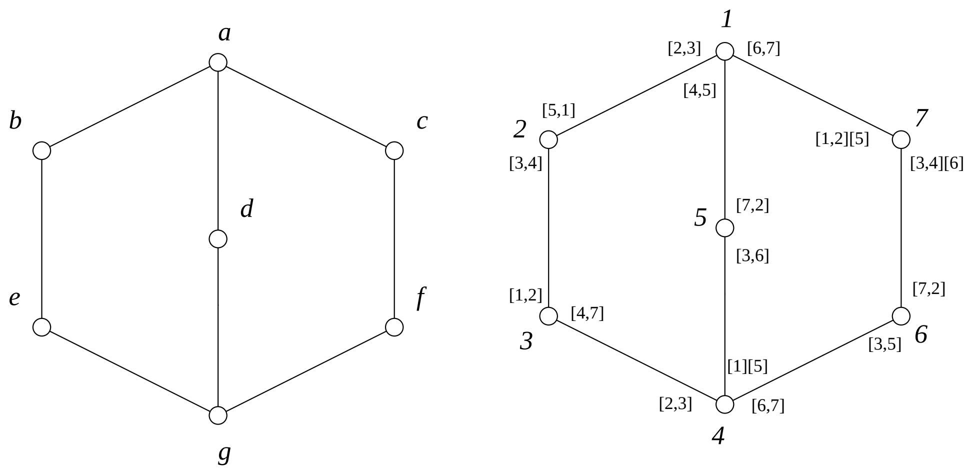


Fig. 1. A graph G and an interval routing scheme for G.

We are interested in the number of intervals that may require an interval routing scheme in the general case.

Definition 2 (Compactness of Graphs).

- Given a graph G of n vertices and an interval routing scheme R on G, we define $\textsc{Irs}(R)$ to be the maximum over all the arcs of G of the number of intervals that is required to encode the destinations associated to that arc.
- The *compactness of a graph* G, denoted by $\textsc{Irs}(G)$, is the $\min_R \textsc{Irs}(R)$ for all shortest path interval routing schemes R on G.

In a sense, $\textsc{Irs}(G)$ is the maximum number of intervals required by the "most compact" interval routing scheme on G. Since interval routing was introduced in the hope of reducing the amount of space required, $\textsc{Irs}(G)$ is an important parameter to consider.

Most of the work in the literature on shortest path interval routing has been concerned with finding the compactness of specific networks: chordal rings in [FGS94], trees, hypercubes, d-meshes, d-tori and r-complete-bipartite graphs in [BvLT91, KKR93, FG94c], unit-interval and unit-circular networks in [FG94c]. In this paper, we exhibit a graph G of maximum degree d with a large compactness, *i.e.* some routers of G need a large local memory to implement any interval routing scheme that uses shortest paths. For every integer n, we define $\textsc{Irs}(n) = \max_G \textsc{Irs}(G)$ such that G is of order n. Similarly, given any two integers n and d, we define $\textsc{Irs}(n, d)$ to be the $\max_G \textsc{Irs}(G)$ such that G has n vertices and a degree bounded by d.

It is well known that, in general, the compactness is not a constant (see [KKR93, FG94b, FvLS94]). It was shown in [KKR93] that there exists a planar graph G of order n such that $\textsc{Irs}(G) = \Omega(\sqrt[3]{n})$ and recently it was proven in [GG95] that $\textsc{Irs}(n) = \Theta(n)$. This last result shows that there is no hope of compressing routing tables using intervals in the general case of graphs of order n. More generally, it was recently proven in [FG95] that for any coding method for shortest path routing, there exists a worst case graph of order n that needs $\Omega(n)$ bits of memory on a router. Therefore, to reduce the routing information on a router we need to consider a smaller class of graphs. We show in this paper, that for every integer n and every integer $d \geqslant 3$, $\textsc{Irs}(n,d) = \Omega(n/\log^2 n)$.

In the next section we introduce a tool that gives a lower bound on the compactness of any graph: the *compactness of a matrix of constraints*. Then in section 3, we show how to build a graph of bounded degree from a given boolean matrix of constraints that possesses a large compactness. Finally in section 4, we give a lower bound on $\textsc{Irs}(n,d)$.

2 Matrices of Constraints

Given an arbitrary graph G, computing $\textsc{Irs}(G)$ is generally difficult and at the present time no polynomial time algorithm is known. In this section, we describe a tool that gives a lower bound on the compactness of a graph G. This tool is based on the notion of a *matrix of constraints*, independently introduced by [FvLS94] and by [GG95].

Consider vertex b in the graph G drawn on the left hand side of Figure 1. Note first that the shortest path from vertex b to vertices a, c, d, e and g is unique. The shortest path from b to a, c and d must use arc (b,a), and the shortest path from b to e and g must use arc (b,e). There is a shortest path from b to f that uses arc (b,a), and another that uses arc (b,e). Either path may be used, and this choice depends on the routing scheme.

Definition 3 (Matrix of Constraints). Let $G=(V,E)$ be a graph, $V' \subset V$ and $E' \subset E$. A *matrix of constraints* of G is a $|V'| \times |E'|$ boolean matrix $M = (m_{v,e})$, $v \in V', e \in E'$, such that the rows of M are labeled by the vertices of V' and the columns of M are labeled by the arcs of E', and such that:

$$\forall v \in V', \forall e = (x,y) \in E',$$

$$\begin{cases} m_{v,e} = 1 \Leftrightarrow \text{ all shortest paths in } G \text{ from } x \text{ to } v \text{ do use the arc } e. \\ m_{v,e} = 0 \Leftrightarrow \text{ no shortest path in } G \text{ between } x \text{ and } v \text{ uses the arc } e. \end{cases}$$

Consider a column (u,v) of a $p \times q$ matrix of constraints and suppose that the vertices of the graph have been labeled with integers by a shortest path interval routing scheme $R = (\mathcal{L}, \mathcal{I})$. If there is a 1 at the intersection of the column (u,v) with the row labeled by vertex w, then the label of w must be on arc (u,v) in R, *i.e.* $\mathcal{L}(w) \in I_{(u,v)}$. Similarly, if there is a 0 then the label of w cannot be on arc (u,v) in R, *i.e.* $\mathcal{L}(w) \notin I_{(u,v)}$. If we permute the rows of the matrix such that the integer labels of the rows are placed in ascending order in the matrix, then clearly the number of blocks of consecutive 1's in column (u,v) is a lower bound on the number of intervals for that arc in R. Figure 2 shows an example of a matrix of constraints M for the Petersen-graph.

$$M = \begin{array}{cc} \begin{matrix} e_1 & e_2 & e_3 \end{matrix} & \\ \left|\begin{matrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{matrix}\right| & \begin{matrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{matrix} \end{array}$$

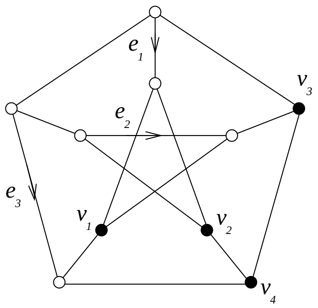


Fig. 2. The matrix M of compactness 2 is a matrix of constraints of the Petersen-graph

Note that a matrix obtained by permuting the rows of a matrix of constraints is also a matrix of constraints for the same graph and corresponds to a relabeling of the vertices. If we can show that under any permutation of the rows of the matrix there must be at least one column with a certain number k of blocks of consecutive 1's, then the graph must require at least k intervals under any shortest path interval routing scheme. Finding a matrix of constraints for a graph G and establishing a bound on the maximum number of blocks of consecutive 1's in a column, minimized over all permutations of the rows of the matrix, therefore yield a lower bound on $\mathrm{IRS}(G)$. We now formalize these ideas.

Definition 4 (Compactness of Matrices). Let M be a boolean matrix. The *compactness* of M, denoted by $\mathrm{I}(M)$, is the maximum, over all columns of M, of the number of blocks of consecutive 1's cyclically, minimized over all row permutations of M.

A block of consecutive 1's is said to be *cyclic* if the first and the last bit in the word formed by a column of M are a 1-entry. Figure 2 gives an example of matrix M of compactness 2. We have the following result due to [Fla94]; it is proven in [FvLS94] and in [GG95].

Lemma 5. *For every matrix of constraints M of a graph G, $\mathrm{IRS}(G) \geqslant \mathrm{I}(M)$.*

Unfortunately we do not know of a better relation between the compactness of a graph and the compactness of the "less compact" matrix of constraints of the graph. Indeed it was shown in [GG95] that the graph G drawn in Figure 1, in section 1, has a compactness 2 though all matrices of constraints of G have a compactness at most 1.

Remark. In [FGS95], the problem of computing the compactness of any boolean matrix has been shown to be NP-hard, just as the problem of computing the compactness of any weighted graph. The problem is unsolved for uniform cost.

3 Graphs of Constraints of Bounded Degree

In this section we show that for every boolean matrix M and every integer $d \geqslant 3$ there exists a graph G of degree bounded by d such that M is a matrix of constraints of G. A minimal class of graphs with the property that for any boolean matrix M there exists a graph of this class that has M as a matrix of constraints, is called a *class of graph of constraints* and all graphs belonging to this class are called a *graphs of constraints*. This notion has been introduced by [GG95]. Note that the problem to compute the compactness of graphs of maximum degree bounded by 2 is straightforward, since such graphs clearly have a compactness of 1 (see [SK85] for Paths and [vLT87] for Rings), *i.e.* $\textsc{Irs}(n, 2) = 1$ for every n. The following Lemma is due to [GG95].

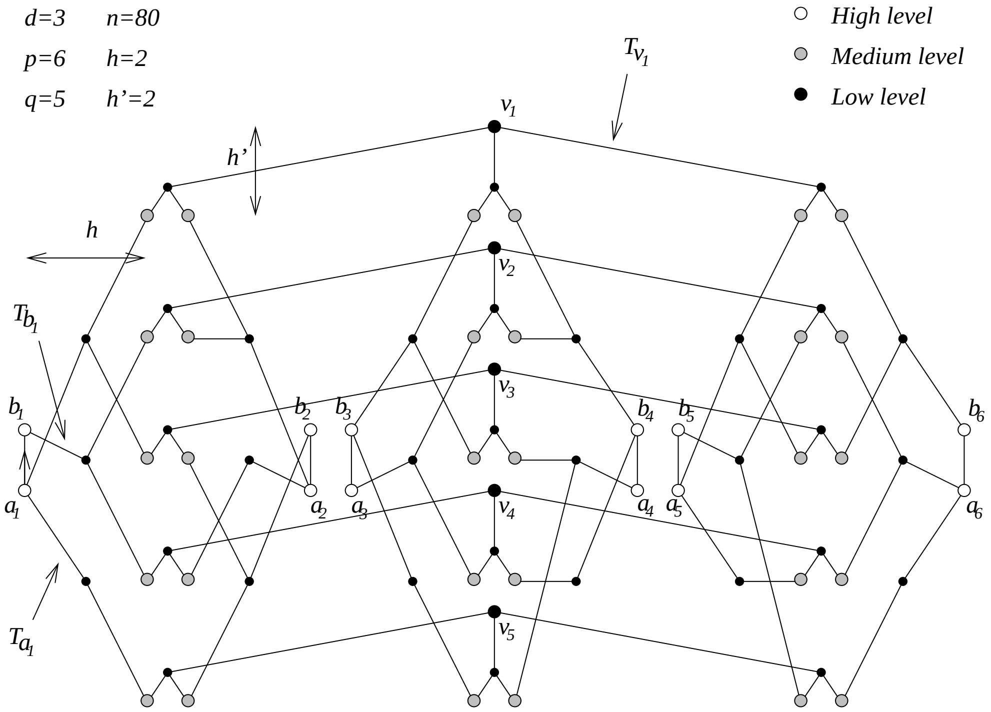
Lemma 6. *For every $p \times q$ boolean matrix M and every integer $d \geqslant 3$, there exists a graph of constraints of matrix M of maximum degree d and with at most $6pq - 4(p + q)$ vertices.*

Refer to Figure 3 for an illustration of the construction. Assume that we are given a $p \times q$ boolean matrix M and an integer $d \geqslant 3$. The symmetric digraph G, that we describe as undirected, is composed of three main levels of vertices denoted Low, Medium and High, and of intermediary vertices (drawn small black dots in Figure 3). The Low level is a set of p independent vertices that we denote v_i, $1 \leqslant i \leqslant p$; the High level consists of q copies of K_2, that we denote $\langle a_j, b_j\rangle$, $1 \leqslant j \leqslant q$; the Medium level is composed of pq independent vertices labeled $w_{i,j}$, $1 \leqslant i \leqslant p$ and $1 \leqslant j \leqslant q$.

These three levels are connected with trees of maximum degree d using intermediary vertices. Let us first describe how the Low and the Medium levels are connected. At every vertex v_i of the Low level, we root a tree T_{v_i}, whose leaves are the $w_{i,j}$'s, for $j \in \{1, \ldots, q\}$. All the trees T_{v_i}'s are isomorphic. They are minimal undirected trees of maximum degree d with exactly q leaves. In each T_{v_i}, $i \in \{1, \ldots, p\}$, the leaves $w_{i,j}$, $j \in \{1, \ldots, q\}$, are all at a distance $h' \in \Theta(\log q)$ from the root v_i.

To connect the Medium and High levels, we use a second type of tree. At vertex a_j (resp. b_j), $j \in \{1, \ldots, q\}$, we root a $(d-1)$-ary tree T_{a_j} (resp. T_{b_j}). In such trees, the root has a degree of at most $d - 1$ while the other vertices have a degree of at most d. In T_{a_j} (resp. T_{b_j}), $j \in \{1, \ldots, q\}$, the leaves are the vertices $w_{i,j}$, for every i such that $m_{i,j} = 0$ (resp. $m_{i,j} = 1$). These trees have at most p leaves. Furthermore, every leaf of the tree T_{a_j} (resp. T_{b_j}) is at distance $h = \Theta(\log p)$ from a_j (resp. b_j). T_{a_j} and T_{b_j} have the smallest possible number of vertices necessary to satisfy these requirements.

In fact, the vertices $w_{i,j}$ can be seen as a grid, where the $w_{i,j}$'s of row i are connected by tree T_{v_i}, while some of the $w_{i,j}$ of column j are connected by tree T_{a_j} and the others of column j are connected by tree T_{b_j}.



$$M = \begin{array}{c|cccccc|c} & (a_1,b_1) & (a_2,b_2) & (a_3,b_3) & (a_4,b_4) & (a_5,b_5) & (a_6,b_6) & \\ \hline & 0 & 0 & 1 & 1 & 0 & 1 & v_1 \\ & 1 & 0 & 0 & 1 & 1 & 0 & v_2 \\ & 0 & 1 & 1 & 0 & 0 & 1 & v_3 \\ & 1 & 0 & 0 & 1 & 0 & 0 & v_4 \\ & 0 & 1 & 1 & 0 & 1 & 0 & v_5 \\ \hline \end{array}$$

Fig. 3. Graph of constraints of the matrix M with $n = 80$ vertices and with a maximum degree $d = 3$.

4 A Lower Bound on Local Memory Requirements

The following theorem establishes a lower bound of $\Omega(n/\log^2 n)$ on the compactness of every bounded degree graph of order n.

Theorem 7. *For any sufficiently large integer n and for every $d \geqslant 3$,*

$$\textsc{Irs}(n,d) \geqslant \frac{n + 4\log_2 n + 5}{4\log_2 n(3\log_2 n + 1)} = \Omega\left(\frac{n}{\log^2 n}\right)$$

Proof. Let n be a sufficiently large integer, let $p = 2^{q-1}$ and let q such that $6pq - 4(p+q) = 3q2^q - 2^{q+1} - 4q \leqslant n < 3(q+1)2^{q+1} - 2^{q+2} - 4(q+1)$. Applying Lemma 6, for every $p \times q$ boolean matrix M there exists a graph of constraints G of M of maximum degree d and with n' vertices, $n' \leqslant n$. We construct a graph

G' with a maximum degree d from G with exactly n vertices by adding a path of $n-n'$ vertices connected to one of the $w_{i,j}$'s vertices of G, which are all of degree two. Clearly $\textsc{Irs}(G') \geqslant \textsc{Irs}(G)$. Hence, for every $d \geqslant 3$, $\textsc{Irs}(n,d) \geqslant \textsc{Irs}(G)$. Thus by applying Lemma 5, for every $2^{q-1} \times q$ boolean matrix M, $\textsc{Irs}(n,d) \geqslant \mathrm{I}(M)$. Let $P = (v_1, \ldots, v_{2^q})$ be a Hamiltonian path in the Hypercube of dimension q. Let M be the $2^{q-1} \times q$ boolean matrix such that row r_i of M, $i \in \{1, \ldots, 2^{q-1}\}$, is the standard binary representation of the vertex v_{2i-1} of P. By construction, any two rows r_i and r_j of M, $i \neq j$, differ by at least two places, since two adjacent vertices in the Hypercube are at an odd distance in P. As it is shown in [GG95, Lemma 3], for every $k \times l$ boolean matrix A such that the Hamming distance between any two rows of A is at least d, $\mathrm{I}(A) \geqslant kd/(2l)$. Therefore $\mathrm{I}(M) \geqslant 2^q/(2q)$.

$$n < 3(q+1)2^{q+1} - 2^{q+2} - 4(q+1) \quad \Rightarrow \quad n + 4q + 5 \leqslant \frac{2^q}{2q} 4q(3q+1)$$
$$\Rightarrow \quad \frac{n+5}{4q(3q+1)} + \frac{1}{3q+1} \leqslant \mathrm{I}(M)$$

By assumption, $n \geqslant 3q2^q - 2^{q+1} - 4q \geqslant 2^q$, for $q \geqslant 3$ and $n \geqslant 44$. Therefore $\log_2 n \geqslant q$ and finally,

$$\textsc{Irs}(n,d) \geqslant \frac{n+5}{4\log_2 n(3\log_2 n + 1)} + \frac{1}{3\log_2 n + 1} = \Omega\left(\frac{n}{\log^2 n}\right)$$

$\square$

Remark. The exact lower bound given by Theorem 7, can be quite improved since the construction is based on Lemma 6 that gives a general construction from any degree $d \geqslant 3$ and any boolean $p \times q$ matrix. However the order of the lower bound, using a graph of constraints of order $O(pq)$ and maximum degree 3, is $\Omega(n/\log^2 n)$.

5 Conclusion

From a local memory requirement point of view, we have seen that there exists a bounded degree graph that still uses $\Omega(n/\log n)$ bits locally for interval routing, compared to $\Theta(n)$ bits for table routing. It would be interesting to determine whether or not there is a bounded degree graph G of n vertices such that $\textsc{Irs}(G) = \Omega(n/\log n)$. If no such graph exists, then the class of bounded degree graphs is a large class of graphs for which interval routing schemes are better than tables in performing shortest path routing.

References

[BvLT91] Erwin M. Bakker, Jan van Leeuwen, and Richard B. Tan. Linear interval routing. *Algorithms Review*, 2:45–61, 1991.

[FG94a] Pierre Fraigniaud and Cyril Gavoille. A characterisation of networks supporting linear interval routing. In ACM PRESS, editor, 13^{th} *Annual ACM Symposium on Principles of Distributed Computing (PODC)*, pages 216–224, August 1994.

[FG94b] Pierre Fraigniaud and Cyril Gavoille. Interval routing schemes. Research Report 94-04, LIP, École Normale Supérieure de Lyon, 69364 Lyon Cedex 07, France, January 1994.

[FG94c] Pierre Fraigniaud and Cyril Gavoille. Optimal interval routing. In Bruno Buchberger and Jens Volkert, editors, *Parallel Processing: CONPAR '94 - VAPP VI*, volume 854 of Lecture Notes in Computer Science, pages 785–796. Springer-Verlag, September 1994.

[FG95] Pierre Fraigniaud and Cyril Gavoille. Memory requirement for universal routing schemes. Research Report 95-05, LIP, École Normale Supérieure de Lyon, 69364 Lyon Cedex 07, France, March 1995.

[FGS94] Michele Flammini, Giorgio Gambosi, and Sandro Salomone. Interval labeling schemes for chordal rings. In Paola Flocchini, Bernard Mans, and Nicola Santoro, editors, 1^{st} *Colloquium on Structural Information & Communication Complexity (SIROCCO)*, pages 111–124, Ottawa, Canada K1S 5B6, May 1994. Carleton University Press.

[FGS95] Michele Flammini, Giorgio Gambosi, and Sandro Salomone. Interval routing schemes. In Ernst W. Mayr and Claude Puech, editors, 12^{th} *Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 900 of Lecture Notes in Computer Science, pages 279–290. Springer, March 1995.

[FJ88] Greg N. Frederickson and Ravi Janardan. Designing networks with compact routing tables. *Algorithmica*, 3:171–190, 1988.

[FJ89] Greg N. Frederickson and Ravi Janardan. Efficient message routing in planar networks. *SIAM Journal on Computing*, 18(4):843–857, August 1989.

[FJ90] Greg N. Frederickson and Ravi Janardan. Space-efficient message routing in c-decomposable networks. *SIAM Journal on Computing*, 19(1):164–181, February 1990.

[Fla94] Michele Flammini. Private communications in New York, June 1994.

[FvLS94] Michele Flammini, Jan van Leeuwen, and A. Marchetti Spaccamela. Lower bounds on interval routing. Technical Report 69, Università di L'Aquila, Dipartimento di matematica Pura ed Applicata, October 1994.

[GG95] Cyril Gavoille and Eric Guévremont. Worst case bounds for shortest path interval routing. Research Report 95-02, LIP, École Normale Supérieure de Lyon, 69364 Lyon Cedex 07, France, January 1995.

[KKR93] Evangelos Kranakis, Danny Krizanc, and S. S. Ravi. On multi-label linear interval routing schemes. In 19^{th} *International Workshop on Graph - Theoretic Concepts in Computer Science - Distributed Algorithms (WG)*, volume 790 of Lecture Notes in Computer Science, pages 338–349, Utrecht, June 1993. Springer-Verlag.

[SK85] Nicola Santoro and Ramez Khatib. Labelling and implicit routing in networks. *The Computer Journal*, 28(1):5–8, 1985.

[vLT87] Jan van Leeuwen and Richard B. Tan. Interval routing. *The Computer Journal*, 30(4):298–307, 1987.

This article was processed using the LaTeX macro package with SIROCCO style