

Space-Efficiency for Routing Schemes of Stretch Factor Three¹

Cyril Gavoille²

LaBRI, Université Bordeaux I, 351, cours de la Libération, 33405 Talence Cedex, France

and

Marc Gengler³

LIM, Université de la Méditerranée, Campus de Luminy, 13288 Marseille Cedex 09, France

Received December 30, 1997; revised October 6, 2000; accepted October 10, 2000

We deal with deterministic distributed *routing algorithms* on arbitrary n -node networks. For each router, we want to minimize the amount of routing information that must be stored in order to implement the local routing algorithm, even if the names of the routers can be chosen in advance. We take also into account the length of the routing paths and consider the *stretch factor*, which is the maximum ratio between the length of the paths computed by the routing algorithm and the distance between the source and the destination. We show that there exists an n -node network on which every routing algorithm of stretch factor $s < 3$ requires at least a total of $\Omega(n^2)$ bits of routing information. We show a similar result for networks of diameter 2.

© 2001 Academic Press

Key Words: compact routing in distributed networks; near-shortest path routing; routing tables.

1. INTRODUCTION AND MODEL

When routing messages, we need the shortest possible routing paths (or paths that are almost the shortest), which means that the ratio between the length of the path and the distance between the nodes is bounded by a constant, which is called the stretch factor of the path. This definition is naturally extended to a network G ; the stretch factor of a routing R on a network G is defined as $s(R, G) = \max_{x \neq y} d_R(x, y)/d_G(x, y)$, where $d_R(x, y)$ is the routing path length between x and y , and $d_G(x, y)$ the distance in G .

¹ A preliminary version of this paper has been partially published in the proceedings of SIROCCO '97.

² E-mail: gavoille@labri.fr.

³ E-mail: Marc.Gengler@esil.univ-mrs.fr.

Routing tables are commonly used. Messages for node i are sent to the port number which is read at the i th entry of the table. While such a technique is *universal* and can generate the shortest paths, each router needs to store $O(n \log n)$ bits, leading to a total of $O(n^2 \log n)$ bits for an n -node network. However, one can hope to do better because topologies are not always arbitrary. Routing in a ring, for example, requires a simple algorithm for each router, if the labeling of nodes can be freely chosen [4]. If not, the routers in a ring could require $\Theta(n)$ bits. In [2], many models of memory complexity are described with results for lower and upper bounds for the average case. In this paper, we choose a very strong model, which takes the names of the nodes in the set $\{1, \dots, n\}$ and allows output port relabeling.

It is straightforward to show that $\Omega(n)$ bits are necessary to store any neighbor-optimal routing table if no node-labeling is allowed. However, if name and output port relabeling is allowed, less bits could be required to optimally route to neighbors. Consider, as an example, the complete graph K_n and assume that, for each node x , the output port numbered i reaches node y , which has the i th lowest label among the $n - 1$ neighbors of x . Hence, a message from x to a neighbor y uses output port y , if $y < x$, and $y - 1$, otherwise. Thus, it is sufficient for router x to know its name. Less than $\Omega(n)$ bits are also sufficient for a neighbor-optimal routing scheme on trees. A specific labeling shows that $O(\sqrt{n})$ per router suffices to route along shortest paths in every n -node tree [6], and this is done independently of the degree.

Our result is proved for a model of routing function described in [8]. A *routing function* R is a pair (P, H) consisting of a *port function* P and a *header function* H . For any two distinct nodes u and v , R produces a path $u = u_0, u_1, \dots, u_k = v$ of nodes, a sequence h_0, h_1, \dots, h_k of headers, and a sequence p_0, p_1, \dots, p_k of output port numbers. A message with header h_i , arriving at node u_i through input port p_i , is forwarded to the output port $P(u_i, p_i, h_i) = p_{i+1}$ with a new header $H(u_i, p_i, h_i) = h_{i+1}$. The input and output ports numbered 0 are associated with the links between the router and its host. Hence, we impose that $p_0 = 0$, $P(u_k, p_k, h_k) = 0$ and $h_0 = v$, fixing the initial header provided to the routing function. Routing functions are expressed as collections of local routing functions $R_x = (P_x, H_x)$, indexed by the names x of the nodes. A *routing scheme* is a function that returns a routing function R for any n -node network.

Our contribution is a lower bound of $\Omega(n)$ bits of local storage needed to implement any local routing function of any stretch factor $s < 3$ in a worst-case n -node network. We show that this situation occurs on $\Theta(n)$ routers. The lower bound is extended to networks of diameter 2. All these results are established in a context where node and port relabeling is allowed, yielding bounds that are valid for whatever choice of the node or port names.

2. PREVIOUS WORK

The best-known upper bound for stretch factor $s = 3$ was established by Awerbuch *et al.* in [1, p. 321]:

THEOREM 1 (Awerbuch, Bar-Noy, Linial, and Peleg). *Every n -node graph has a routing algorithm of stretch factor at most 3 with a total memory requirement $O(n^{3/2} \log^{3/2} n)$ bits.⁴*

We will see that routing with stretch factor $s < 3$ requires $\Omega(n^2)$ bits, significantly more than for $s = 3$. The $\Omega(n^2)$ lower bound recalls the minimum number of bits to code a n -node network as the number of (unlabeled) topologies is $2^{n^2/2 + o(n^2)}$ [12]. However, this counting argument neither gives any way to compute the routing functions, nor can it be applied to obtain a lower bound. Indeed, the number of routing functions generated by a scheme is smaller than the number of topologies. Peleg and Upfal [11] described an example of nonisomorphic graphs supporting a same shortest path routing table. In [11], Peleg and Upfal proved that $2^{\Omega(n^{1+1/(2s+4)})}$ routing functions are required to route with stretch factor at most s on any n -node network. In [5], Fraigniaud and Gavoille improved this lower bound by proving the existence of $2^{\Omega(n^2)}$ nonisomorphic routing functions of stretch factor $s < 2$, including the case of shortest path routing ($s = 1$). Krizanc and Kranakis [9] reformulated the proof of [5] in terms of Kolmogorov Complexity and showed also a lower bound of $\Omega(n^2)$ bits for $s < 2$. For stretch factor 1, Gavoille and Pérennès showed in [8] an optimal lower bound of $\Theta(n^2 \log n)$ bits, while Buhrman *et al.* showed in [2] that $O(n^2)$ bits are sufficient for shortest-path routing on almost all networks. For $s = 3$, Peleg–Upfal’s lower bound gives $\Omega(n^{1.1})$ bits, whereas $O(n^{3/2} \log^{3/2} n)$ bits suffices [1]. We will show that $\Omega(n^2)$ are necessary for any stretch factor $s < 3$.

3. LOWER BOUND OF THE MEMORY REQUIREMENT

We will lower bound the number of bits required to describe a routing function of stretch factor less than 3 on n -node networks. We assume that the names of the nodes are unique integers taken in the set $\{1, \dots, n\}$. For every x , the port numbers are unique integers taken in the set $\{1, \dots, \deg(x)\}$, where $\deg(x)$ is the degree of x . Headers are of unlimited size and can be modified before the message is forwarded. However, the first routing decision, in the router of the source, must be taken knowing only the name of the destination, in agreement with $h_0 = v$ and $p_0 = 0$ of the definition in Section 1. The topology of the network is represented by an undirected, unlabeled, connected graph. Given $G = (V, E)$ and a routing algorithm $R = \{R_x \mid x \in V\}$ on G , we define the *memory requirement of node x* as the length of the smallest program that computes R_x , expressed, e.g., in terms of the Kolmogorov Complexity [10]. The *local memory requirement of R* is the largest memory requirement of a node taken over all nodes of G . The *total memory requirement of R* is the sum of all local memory requirements.

The basic idea is to show that some graphs of girth 4 cannot optimally route messages to their neighbors if all the routers store only $o(n)$ bits.⁵ It follows that certain routers, say x , that must send a message to some neighbors, say y , forward

⁴ The original paper claimed a bound of $O(n^{3/2} \log^2 n)$ bits that can be slightly improved by a remark of [3].

⁵ $f(n) = o(g(n))$ means that functions f and g satisfy $\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = 0$.

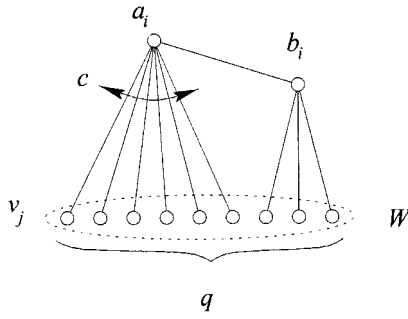


FIG. 1. A router a_i with c connections in the set W .

the message to some other node z , $y \neq z$. Either the message returns from z to x before reaching y , or the message routes from z to y along some other path. In all cases, the length of the path, and hence the stretch factor, is at least 3. So, some routers need at least $\Omega(n)$ bits to route with $s < 3$. This situation may happen independently on $\Theta(n)$ routers, whence a total of $\Omega(n^2)$ bits. We can prove the following result using Stirling's Formula.

LEMMA 1. For every $q \geq 1$, $\binom{q}{4q/5} / \binom{2q/5}{q/5} > \gamma(\frac{5}{4})^q$, with $\gamma \approx 0.619$.

THEOREM 2. For every sufficiently large integer n , there exists an n -node graph on which every routing algorithm of stretch factor $s < 3$ has a total memory requirement of $n^2/25$ bits at least.

Proof. Let $p = \lfloor n/4 \rfloor$, $q = n - 2 \sim^6 n/2$ and c integer, with $q/2 < c < q$. Let \mathcal{M} be the set of $p \times q$ boolean matrices having exactly c 1-entries per row. The cardinality of \mathcal{M} is $|\mathcal{M}| = \binom{q}{c}^p$. For matrix $M = (m_{i,j})$, we build a graph $G = (V, E)$, with $2p + q = n$ nodes. $V = W \cup A \cup B$, where $W = \{v_1, \dots, v_q\}$, $A = \{a_1, \dots, a_p\}$, and $B = \{b_1, \dots, b_p\}$. For every $i \in \{1, \dots, p\}$, a_i is adjacent to b_i , and for every $j \in \{1, \dots, q\}$, v_j is adjacent to a_i if $m_{i,j} = 1$, or to b_i if $m_{i,j} = 0$. In other words, the i th row of M is a bit-vector that indicates which nodes of W are neighbors of a_i . By construction, the distance between nodes of W is at least 2. See Fig. 1 for a component $\{a_i, b_i\}$. Assume that a_i sends a message to one of its neighbors in W , say v_j . If a_i forwards a message to some other node, either to b_j or to some $v_k \in W$ with $k \neq j$, the resulting path will be of length at least 3, which is not admissible with $s < 3$. Thus, any routing function R with $s < 3$ must route messages from any node $a_i \in A$ to one of the neighbors of a_i in W as if the stretch factor was 1.

Let $R = \{R_x \mid x \in V\}$ be any routing algorithm of stretch factor $s < 3$ on G . Let $MR(x)$ be the local memory requirement of x . The nodes and their ports are assumed to be optimally labeled. With the following objects, we can compute exactly M : the integers n , c , p , and q ; the sequence (v_1, \dots, v_q) of the labels of nodes of W ; the set $F = \{R_x \mid x \in A\}$ of routing functions; a permutation π of $\{1, \dots, p\}$; a sequence $T = (t_1, \dots, t_p)$ of integers. We may assume that the routing functions of the set F are ordered lexicographically; i.e., $F = \{F_1, \dots, F_p\}$. The permutation π is defined by $\pi(i) = j$ if and only if $F_j = R_{a_i}$, for every $i \in \{1, \dots, p\}$. Let us now describe how to rebuild M .

⁶ We use notation $f(n) \sim g(n)$ to express that $f(n) = g(n) + o(g(n))$.

(1) For each $v \in W$ and $x \in A$, store in a $p \times q$ integer matrix $M' = (m'_{i,j})$ the values returned by $R_x = (P_x, H_x)$ when the input port is 0; i.e., $m'_{i,j} = Q_{\pi(i)}(0, v_j)$, where $Q_{\pi(i)}$ is the port function of $F_{\pi(i)} \in F$.

If the stretch factor was $s = 1$, each row of M' would use all the output ports of the sender and all the ports, except one, would occur exactly once. From this, we could immediately rebuild M as the *unique occurrences* correspond to the neighbors of the sender in W . However, with a stretch factor $s > 1$, messages (to nonneighbors of the sender) may initially be sent to a wrong node, thereby perturbing the unique occurrences. We now show how to overcome this problem. The following steps are applied to all the rows of M' . Let L be the i th row.

(2) From L , compute U , the bit-vector of *unique occurrences* of L , where the j th bit of U is set if and only if the j th entry of L appears once in L .

(3) Let z be the number of 0-entries of U . Enumerate, using lexicographic order, all the subsets of $\{1, \dots, z\}$ of size $c - (q - z)$. Using sequence T , we take the t_i th subset, say B , which represents the positions of the 0-entries of U to flip. Compute X by flipping the corresponding positions in U .

We claim that X is exactly the i th row of M . Step (1) stores in M' the answers of the routers of A for the nodes of W . Call L the i th row of M' .

Claim 1. The number of unique occurrences of L is at least $2c - q$.

L has at least c distinct entries, otherwise the routing function R would be of stretch factor $s \geq 3$. Thus, at most $q - c$ entries can disturb in L the c distinct entries originally in the i th row of M and there are at least $c - (q - c) = 2c - q > 0$ entries of L with unique occurrences. After Step (2), the bits set to 1 in U are correct. Suppose that $q - z$ bits are correct. Step (3) tries to determine the $c - (q - z)$ entries that are not "correct." Since these are among the z 0-entries of U , it is sufficient to describe the right set of positions to flip, using B , to rebuild the i th row of M entirely. The number of ways to do it is $\binom{c - (q - z)}{z}$. Let us now compute the total number of bits used for this construction. The following claims are easy to establish.

Claim 2. Every finite sequence $S = (s_1, \dots, s_k)$ of integers with $1 \leq s_i \leq m$, can be described, knowing k and m , by at most $k \log m + O(1)$ bits.

Claim 3. $z \leq 2(q - c)$ and $\binom{c - (q - z)}{z} \leq \binom{2(q - c)}{q - c}$.

The integers n , c , p , and q can be described with $O(\log n)$ bits. By Claim 1, the sequence (v_1, \dots, v_q) can be stored on $q \log n + O(1)$ bits. The permutation π can be stored on $p \log p + O(1)$ bits. T can be described by

$$p \log \binom{2(q - c)}{q - c} + O(1)$$

bits using Claim 2 and Claim 3. If K is the number of bits to store the set F of routing functions, we have

$$K \geq \log \left(|\mathcal{M}| \binom{2(q - c)}{q - c}^p \right) - O(n \log n).$$

On the other hand, we know that $K \leq \sum_{x \in \mathcal{A}} \text{MR}(x) + O(\log \text{MR}(x))$. Therefore, using Lemma 1 and choosing $c = 4q/5$, we get $\sum_{x \in \mathcal{A}} \text{MR}(x) > pq \log(\frac{5}{4}) - O(n \log n) \geq \frac{n^2}{25}$. ■

We showed that *at least one* graph has a memory requirement of $\Omega(n^2)$ bits. By a counting argument, one checks that $(1 - o(1)) 2^{n^2/25} \sim 2^{n^2/25}$ graphs need a total of $n^2/25$ bits to route with stretch factor $s < 3$. It is well known that almost all graphs have diameter 2. We strengthen Theorem 2 by showing that $\Omega(n^2)$ bits can be required for some worst-case graphs of diameter 2. The next lemma will be useful for the main theorem.

LEMMA 2. *Let c, p, q be sufficiently large integers with $0 < c < q$. Let \mathcal{M} be the set of $p \times q$ boolean matrices having c 1-entries per row. Let \mathcal{M}_1 be the subset of matrices of \mathcal{M} such that any two columns are noncomplemented. Let \mathcal{M}_2 be the subset of matrices of \mathcal{M} such that any $2 \times q$ matrix composed of a pair of rows of \mathcal{M} contains the submatrix*

$$\begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

up to a column permutation. We have the following two properties. If $q = o(r^{p/2})$, then $|\mathcal{M}_1| \sim |\mathcal{M}|$, where $r = 2(1 - 1/q)$. If $p = o(r^{q/2})$, then $|\mathcal{M}_2| \sim |\mathcal{M}|$, where $r = 2q^{-1/(2q)}$ if $c = q/2$, and $r = \alpha^{-2\alpha}(2\alpha - 1)^{2\alpha - 1}$ with $\alpha = \max\{c/q, 1 - c/q\}$, otherwise.

Proof. The complete proof can be found in [7]. ■

THEOREM 3. *For every sufficiently large integer n , there exists an n -node graph of diameter 2 on which every routing algorithm of stretch factor $s < 3$ has a total memory requirement of $n^2/25$ bits at least.*

Proof. Let us consider the set $\mathcal{M}' = \mathcal{M}_1 \cap \mathcal{M}_2$ of Lemma 2 and show that all the graphs built from \mathcal{M}' are of diameter 2. Let $M \in \mathcal{M}'$ and $G = (V, E)$ its corresponding graph, constructed as in Theorem 2. The distance between any a_i (or b_i) and any v_j is at most 2. $M \in \mathcal{M}_1$ implies that columns of M are pairwise noncomplemented. Thus for every j, j' there exists an i such that $m_{i,j} = m_{i,j'}$. If $m_{i,j} = m_{i,j'} = 1$, then v_j and $v_{j'}$ are both neighbors of a_i . If $m_{i,j} = m_{i,j'} = 0$, both are neighbors of b_i . Hence, $d_G(v_j, v_{j'}) = 2$. Since $M \in \mathcal{M}_2$, we know that, for any two distinct rows i and i' , there exist j_1 such that $m_{i,j_1} = 1 = m_{i',j_1}$, j_2 such that $m_{i,j_2} = 0 = m_{i',j_2}$, j_3 such that $m_{i,j_3} = 1 \neq m_{i',j_3}$ and j_4 such that $m_{i,j_4} = 0 \neq m_{i',j_4}$. Thus, $d_G(a_i, a_{i'}) = 2$ as $m_{i,j_1} = 1 = m_{i',j_1}$ implies that v_{j_1} is a neighbor of both a_i and $a_{i'}$. The other cases are proved in the same way and establish that G is of diameter 2.

Choose $p \sim n/4$, and $q \sim n/2$. Let $c = \alpha q$, and $\alpha = 4/5$. From Lemma 2, $|\mathcal{M}'| \sim |\mathcal{M}|$, if $q = o(r_1^{p/2})$, and $p = o(r_2^{q/2})$, with $r_1 = 2(1 - 1/q) \sim 2(1 - 2/n)$, and with $r_2 = (5/16) \cdot 4^{2/5} \cdot 3^{3/5} \approx 1.05$. We get $q = o(2^{n/8}(1 - 2/n)^{n/8}) = o(2^{n/8})$, and $p = o(1.05^{n/4})$, thus $|\mathcal{M}'| = (1 - o(1)) |\mathcal{M}|$. Let us apply the proof of Theorem 2 for the set \mathcal{M}' instead of \mathcal{M} . We have

$$\sum_{x \in \mathcal{A}} \text{MR}(x) > \log \left(|\mathcal{M}'| \left/ \left(\frac{2(q-c)^p}{q-c} \right) \right) - O(n \log n),$$

with

$$|\mathcal{M}'| = (1 - o(1)) \binom{q}{c}^p.$$

For $\alpha = c/q = 4/5$, Lemma 1 gives

$$|\mathcal{M}'| / \binom{2q/5}{q/5}^p > (1 - o(1)) \gamma^p \left(\frac{5}{4}\right)^{pq}.$$

As $\gamma < 1$, we have

$$\sum_{x \in \mathcal{A}} \text{MR}(x) > pq \log \left(\frac{5}{4}\right) - O(p) - O(n \log n) > \frac{n^2}{25}. \blacksquare$$

4. IMPROVING THE LOWER BOUNDS

THEOREM 4. *For every sufficiently large n , there exists an n -node graph of diameter 2 on which every routing algorithm of stretch factor $s < 3$ has a local memory requirement asymptotically of $n \log(5/4) \approx 0.32n$ bits at least.*

Proof. Consider the same graphs as in Theorem 3, built from the set of matrices \mathcal{M}' , with $p = n/\log n$, $q = n - 2p \sim n$ and $\alpha = c/q = 4/5$. To apply Lemma 2 it suffices to check that $q = o(r^{n/\log n})$, and $p = o(1.05^{n/2 - o(n)})$, with $r = 2(1 - 1/q) \sim 2(1 - 1/n)$. We check that $r^{n/\log n} > 2^{(n-2)/\log n}$ and $q = o(2^{(n-2)/\log n})$. So, $|\mathcal{M}'| \sim |\mathcal{M}|$. By the last inequality of Theorem 3, there exists at least one router of A , x_0 , such that

$$\text{MR}(x_0) > q \log \left(\frac{5}{4}\right) - O\left(\frac{n \log n}{p}\right),$$

which implies that $\text{MR}(x_0) > n \log(5/4) - O(\log^2 n)$. \blacksquare

Actually, the bound of the total memory requirement in Theorem 2 can be improved by a factor two, but the graph is not of diameter two.

THEOREM 5. *For every sufficiently large integer n , there exists an n -node graph of diameter 3 on which every routing algorithm of stretch factor $s < 3$ has a total memory requirement $n^2/13$ bits at least.*

Proof. Let G be any graph built like in Theorem 2 from the set of $p \times q$ matrices with c 1-entries per row, \mathcal{M} . Choose $c = 4q/5$, $p = \lfloor n/2 \rfloor$, and $q = \lceil n/2 \rceil - 1$. We transform G into H obtained by removing all the nodes b_i of G , and by adding a new node a_0 connected to all the nodes v_i , for $i \in \{1, \dots, q\}$. H has $p + q + 1 = n$ nodes exactly and diameter 3. Let R be any routing algorithm on H of stretch factor $s < 3$. The proof of Theorem 2 applies for H as well. If a router a_i would not route to all its neighbors optimally, R would be of stretch factor $s \geq 3$. As the rebuilding procedure of Theorem 2 does not depend of the nodes b_i , we can rebuild

the routing functions of the a_i of H using the same procedure as in Theorem 2. The last inequality of Theorem 3 holds and

$$\sum_{x \in \mathcal{A}} \text{MR}(x) > pq \log \left(\frac{5}{4} \right) - O(n \log n) > \frac{n^2}{13}. \quad \blacksquare$$

5. CONCLUSION

We proved a lower bound of $\Omega(n^2)$ bits for the entire network to route with any stretch factor $s < 3$. Let $s(m)$, for any integer function m , be the smallest real such that every network supports a distributed algorithm of stretch factor $s \leq s(m)$ with a total of $o(m)$ bits. We showed that there is a gap for stretch factor 3, and $s(n^2) = 3$. It would be interesting to compute $s(n^2 \log n)$, i.e., the threshold stretch factor for compression of routing tables. At the present time, best estimations are $1 < s(n^2 \log n) \leq 3$. The same evaluation on stretch factor threshold for compression of routing information, $\bar{s}(m)$, could be done for the average of all the networks, rather than for the worst-case. Works of [2] tends to show a quite different behavior between $s(m)$ and $\bar{s}(m)$, for instance $\bar{s}(n^2 \log n) = 1$.

ACKNOWLEDGMENTS

We thank André Raspaud and Guy Mélançon for their help in the proof of Lemma 2.

REFERENCES

1. B. Awerbuch, A. Bar-Noy, N. Linial, and D. Peleg, Improved routing strategies with succinct tables, *J. Algorithms* **11** (1990), 307–341.
2. H. Buhrman, J. H. Hoepman, and P. Vitányi, Space-efficient routing tables for almost all networks and the incompressibility method, *SIAM J. Comput.* **28**(4) (1999), 1414–1432.
3. T. Eilam, C. Gavoille, and D. Peleg, Compact routing schemes with low stretch factor, in “17th Annual ACM Symposium on Principles of Distributed Computing (PODC),” pp. 11–20, 1998.
4. P. Flocchini, B. Mans, and N. Santoro, Sense of direction: Formal definitions and properties, in “1st International Colloquium on Structural Information and Communication complexity (SIROCCO),” Carleton University Press, pp. 9–34, 1994.
5. P. Fraigniaud and C. Gavoille, Memory requirement for universal routing schemes, in “14th Annual ACM Symposium on Principles of Distributed Computing (PODC),” pp. 223–230, 1995.
6. C. Gavoille, A survey on interval routing, *Theoret. Comput. Sci.* **245** (2000), pp. 217–253.
7. C. Gavoille and M. Gengler, “Space-Efficiency of Routing Schemes of Stretch Factor Three,” Research Report RR-1181-97, LaBRI, University of Bordeaux, 33405, Talence Cedex, France, October 1997.
8. C. Gavoille and S. Pérennès, Memory requirement for routing in distributed networks, in “15th Annual ACM symposium on Principles of Distributed Computing (PODC),” pp. 125–133, 1996.

9. E. Kranakis and D. Krizanc, Lower bounds for compact routing, in "13th Annual Symposium on theoretical Aspects of Computer Science (STACS)," Lecture Notes in Computer Science, Vol. 1046, pp. 529–540, Springer-Verlag, Berlin/New York, 1996.
10. M. Li and P. M. B. Vitányi, "An Introduction to Kolmogorov Complexity and Its Applications," Springer-Verlag, Berlin/New York, 1993.
11. D. Peleg and E. Upfal, A trade-off between space and efficiency for routing tables, *J. Assoc. Comput. Math.* **36**(3) (1989), 510–530.
12. E. M. Wright, Graphs on unlabelled nodes with a given number of edges, *Acta Math.* **126** (1971), 1–9.